

특집논문 (Special Paper)

방송공학회논문지 제28권 제6호, 2023년 11월 (JBE Vol.28, No.6, November 2023)

<https://doi.org/10.5909/JBE.2023.28.6.720>

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

뉴럴네트워크의 잔차 파라미터를 활용한 AI-PCC 기반 스트리밍 프레임워크

이 민 석^{a)}, 김 준 식^{b)}, 이 성 배^{a)}, 김 규 현^{a)†}

AI-PCC based Streaming Framework using Residual Parameters of Neural Networks

Minseok Lee^{a)}, Junsik Kim^{b)}, Seongbae Rhee^{a)}, Kyuheon Kim^{a)†}

요 약

최근 3차원 포인트 클라우드의 획득/표현 장비가 발전함에 따라 자율주행 등의 응용 분야에서 포인트 클라우드의 활용에 관한 기술 연구가 활발히 진행 중이다. 그러나 포인트 클라우드 기반의 응용 프로그램 상용화에는 여러 제한 사항이 존재하며, 이 중 하나는 포인트 클라우드 콘텐츠의 막대한 데이터 크기이다. 이러한 제한 사항을 해결하기 위해 여러 포인트 클라우드 압축 기술이 개발되고 있으며, 최근 인공지능 기반의 AI-PCC(AI-based PCC) 기술이 등장하였다. AI-PCC는 압축 성능이 우수하다는 장점이 있으나, 압축된 포인트 클라우드 비트스트림과 함께 뉴럴네트워크를 구성하는 파라미터들이 전송되어야 한다는 한계가 존재한다. 이는 AI-PCC 뉴럴 네트워크 내부의 파라미터가 학습 방식에 따라 다르게 설정되기에 발생하는 문제이다. 큰 크기를 갖는 뉴럴네트워크 파라미터를 수시로 전송한다면 대역폭 이용에 있어 효율이 많이 떨어지기에 AI-PCC를 활용한 포인트 클라우드의 전송 및 소비는 어려움이 있다. 따라서 본 논문은 AI-PCC의 잔차 파라미터를 INNC(Incremental Neural Network Coding) 기반으로 압축 후 전송하여 효율적인 AI-PCC 기반 스트리밍을 가능케 하는 프레임워크를 제안한다.

Abstract

Recent advancements in 3D point cloud capture and expression technologies have led to progressions in point cloud usage for applications such as metaverse or autonomous driving. However, several challenges exist when trying to use point clouds in commercial applications. One of these key challenges is the vast size of point cloud contents. Various technologies for efficient PCC(Point Cloud Compression) are undergoing active development to resolve this limitation. The most recent approach for compressing point clouds is AI-PCC(AI-based PCC), which refers to PCC technologies using neural networks usually applying the concept of the well-known VAE(Variational Auto Encoder). Even though AI-PCC achieves great performance, it faces a problem: the server needs to transmit the parameters of the AI-PCC neural network along with the compressed point cloud bitstream. The reason for transmitting the parameters is that AI-PCC codecs have been trained with different parameters on different datasets according to their predetermined quality rate, and this can cause critical memory and bandwidth issues when implementing a streaming protocol. Therefore, this paper proposes a framework for AI-PCC streaming by compressing the residual parameters of AI-PCC neural networks using INNC(Incremental Neural Network Compression).

Keywords : Point Cloud, AI-PCC, INNC, Neural Network, Residual Parameters

Copyright © 2023 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 3D 데이터의 획득 및 표현 기술이 발전함에 따라 AR/VR 등 사용자가 초실감 미디어 콘텐츠를 소비할 수 있는 서비스가 많이 증가하였다. 이러한 초실감 미디어 콘텐츠를 표현하기 위한 데이터형은 여러 존재하는데, 3차원상의 점의 위치를 정밀하게 표현할 수 있다는 장점으로 인해 점의 집합으로 표현되는 3D 포인트 클라우드 데이터에 관심이 주목되고 있다^[1]. 그러나 3D 포인트 클라우드는 3차원 위치 정보와 색상(색상 등) 정보를 모두 포함하기 때문에, 기존 2D 비디오에 비해 데이터 크기가 막대하여 전송 및 소비에 어려움을 겪고 있다^[2]. 이에 따라 3차원 포인트 클라우드의 효율적 전송 및 활용을 위해 국제 표준화 기구인 ISO/IEC 산하 MPEG(Moving Pictures Expert Group)의 WG(Working Group)7에서 포인트 클라우드의 압축에 관한 연구가 활발히 진행 중이다^[3]. MPEG의 포인트 클라우드 압축 기술은 압축 방식에 따라 G-PCC(Geometry-based Point Cloud Compression), V-PCC(Video-based Point Cloud Compression) 등으로 명명되어 연구가 진행 중이며, 그중 최근 큰 관심을 받는 기술은 뉴럴네트워크를 활용한 AI-PCC(AI-based PCC)이다^[4]. AI-PCC는 뉴럴네트워크가 동반된 모든 포인트 클라우드 압축 기술을 일컫으며, 현재 표준화 준비 단계에 있다. AI-PCC 기술 대부분은 2D 영상 압축 분야에서 큰 성공을 거둔 VAE(Variational Auto Encoder)^[5]의 개념을 3D에 적용한 접근법을 활용하며, 기존 MPEG의 PCC 기술들과 비교하여 월등히 높은 압축 및 복원 성능을 낸다는 특징이 있다.

그러나 AI-PCC 기술은 전통적인 PCC 기술들과 비교

하여 많은 양의 CP(Codecistic Parameter)가 발생한다는 단점이 존재한다. 전통적인 PCC 기술은 양자화 파라미터(Quantization Parameter)라는 CP를 활용하여 압축률과 복원 품질 간의 트레이드오프(Trade-off)를 조절하며^[6], 이러한 양자화 파라미터는 그 크기가 매우 작아 비트스트림의 크기에 미치는 영향이 미미하다. 그에 반해 AI-PCC 기술은 학습된 뉴럴네트워크(코덱)의 내부 파라미터(가중치, 편향 등)를 CP로 활용하여 포인트 클라우드의 압축률 및 복원 품질을 조절한다. 이는 곧 뉴럴네트워크 자체가 포인트 클라우드 복원에 필요한 정보임을 뜻하는데, 압축된 포인트 클라우드 비트스트림의 수백 배 크기를 웃도는 AI-PCC 뉴럴네트워크의 크기를 감안한다면^[7] 상당히 많은 양의 CP가 동반되어야 함을 알 수 있다. 즉, AI-PCC 기술은 기존 방식에 비해 입력 포인트 클라우드를 매우 효율적으로 압축할 수 있지만, 요구되는 CP의 크기가 너무 커서 전송 페이로드에 영향을 끼치기에 스트리밍 환경에서 활용하기에는 부적절하다는 한계가 있다. 이러한 한계를 극복하고자 본 논문에서는 IASF(ICNN-based AI-PCC Streaming Framework)를 제안한다. IASF는 AI-PCC 뉴럴네트워크 간의 잔차 파라미터를 구한 후 이를 INNC(Incremental Neural Network Coding) 기반으로 압축하여 CP의 크기를 줄임으로써, 더욱 효율적인 AI-PCC 기반 스트리밍을 지원하는 프레임워크이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기반이 되는 배경 기술 및 관련 연구에 대한 설명을 제공하며, 3장에서는 본 논문이 제안하는 INNC 기반 AI-PCC 스트리밍 프레임워크의 구조 및 작동 원리에 대해 자세히 설명한다. 4장에서는 제안 기술을 활용한 실험 결과를 통해 기술의 성능을 검증하며, 5장에서는 본 논문에 대한 결론을 맺는다.

a) 경희대학교 전자공학과(Department of Electronic Engineering, Kyung Hee University)

‡ Corresponding Author : 김규현(Kyuheon Kim)

E-mail: kyuheonkim@khu.ac.kr

Tel: +82-31-201-3810

ORCID: <https://orcid.org/0000-0003-1553-936X>

※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (Grant number: IITP-2023-2021-0-02046) and (No. RS-2023-00227431, Development of 3D space digital media standard technology) supervised by the Institute of Information & communications Technology Planning & evaluation (IITP).

· Manuscript September 11, 2023; Revised October 23, 2023; Accepted October 23, 2023.

II. 배경 기술

1. INNC(Incremental Neural Network Coding)

본 절에서는 INNC(Incremental Neural Network Coding) 표준, 혹은 ISO/IEC 15938-17(Compression of Neural Networks for Multimedia Content Description and Analy- sis)

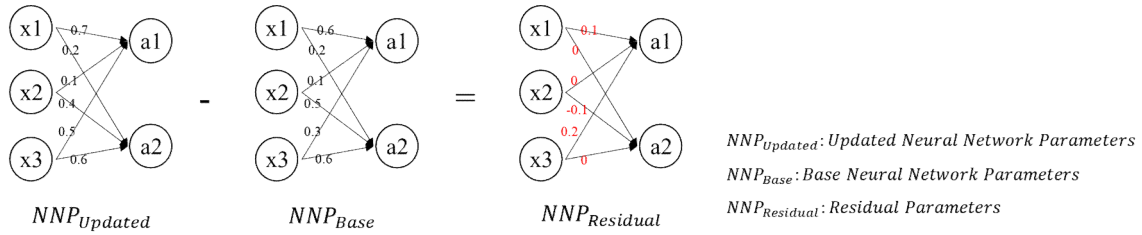


그림 1. 잔차 파라미터 생성에 대한 예시
 Fig. 1. Example for generation of residual parameters

edition 2^[8]에 관해 설명한다. INNC는 국제 표준화 단체인 MPEG(Moving Picture Experts Group)에서 뉴럴네트워크를 구성하는 파라미터들(NNP(Neural Network Parameters))을 효율적으로 압축하기 위해 제작한 표준인 NNC(Neural Network Coding)의 두 번째 에디션(edition)이며, 기본 압축 구조도는 NNC의 첫 에디션과 동일하나 압축하는 대상이 뉴럴네트워크의 잔차 파라미터라는 점에서 차이가 존재한다. 여기서 잔차 파라미터란, [그림 1]의 예시와 같이 동일한 구조를 갖는 서로 다른 두 뉴럴네트워크의 파라미터(가중치, 편향 등) 간 차이값에 해당하며, 잔차 파라미터 또한 두 뉴럴네트워크와 동일한 뉴럴네트워크 구조를 가지게 된다. 이러한 잔차 파라미터는 연합학습^[9]과 같이 뉴럴네트워크의 파라미터 전송이 수시로 일어나는 환경에서 그 필요성이 제안되었다.

[그림 1]에서 확인할 수 있듯, 잔차 계산에 사용되는 두 NNP 간의 유사도가 높으면, 그에 따른 $NNP_{Residual}$ 은 0의 분포가 매우 높은 하이퍼 라플라시안 분포(Hyper-Laplacian Distribution)를 띄게 되는데, 하이퍼 라플라시안 분포를 띄는 데이터는 엔트로피가 매우 낮아 샤논의 정리^[10]에 의해 큰 압축률을 달성할 수 있다. 이와 같은 특성을 기반으로 MPEG은 잔차 파라미터를 활용한 뉴럴네트워크 압축 방

법인 INNC를 고안하였다.

INNC 표준 부호화기의 압축 구조도 및 입력 데이터에 대한 변화 과정은 [그림 2]와 같다. 입력 데이터는 뉴럴네트워크를 구성하는 가중치 및 편향 등의 파라미터 값들의 집합인 NNP이며, 입력 시 다차원 행렬로써 표현된다. 이와 같은 NNP는 파라미터 제거(Parameter Reduction) 모듈을 먼저 거치게 되는데, 해당 모듈에서는 희소화(Sparsification) 등의 기술을 통해 NNP들 중 0의 비율을 강제로 높이는 작업을 수행한다. 이와 같은 파라미터 제거 모듈을 거친 후에 남겨진 파라미터들은 파라미터 양자화(Parameter Quantization) 모듈에 입력되며, 균일 양자화(Uniform Quantization) 등 널리 알려진 양자화 기술을 통해 비트 템스(Bit Depth)를 줄이는 작업을 진행한다. 이렇게 양자화된 파라미터들은 엔트로피 코딩(Entropy Coding) 모듈을 거치며 부호화되어 최종 압축 비트스트림이 완성된다. INNC에서는 DeepCABAC(Context-Adaptive Binary Arithmetic Coding for Deep Neural Network Compression)을 엔트로피 코딩 기술로 채택하였으며, 이는 HEVC(High Efficiency Video Codec) 등 여러 비디오 코덱에서 활용되는 CABAC (Context-Adaptive Binary Arithmetic Coding)을 기반으로 디자인된 뉴럴네트워크 압축을 위한 엔트로피 코딩 기술이다. 또한,

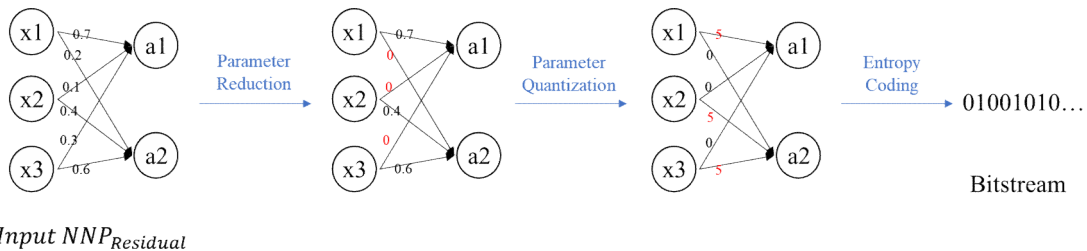


그림 2. INNC 부호화기의 작동 예시
 Fig. 2. Example for INNC encoder workflow

INNC 복호화기의 작동 방식은 부호화기의 역순이다.

INNC 표준 코덱은 이전 버전인 NNC에는 없던 SBTQ (Stochastic Binary-Ternary Quantization)^[11] 혹은 RS(Row Skipping)^[12] 등 많은 양의 0을 최대한 효과적으로 압축하는 기술들을 활용하여 $NNP_{Residual}$ 을 더욱 효율적으로 압축한다는 데에 그 의미가 있으며, 기존 NNC 표준 코덱으로 압축한 데이터의 크기를 약 5%까지 더 줄일 수 있음이 확인되었다.

2. AI-PCC(AI-based Point Cloud Compression)

AI-PCC는 뉴럴네트워크가 동반된 모든 포인트 클라우드 압축 기술로서 2D 영상 압축 분야에서 큰 성공을 거둔 VAE(Variational Auto Encoder)의 개념을 3D에 데이터 특성에 맞게 최적화하여 적용하는 접근법을 활용하고 있으며, 기존 포인트 클라우드 압축 기술인 G-PCC 혹은 V-PCC와 같이 전통적인 영상 신호처리 기법 대비 상당히 높은 압축 성능을 보인다^[4]. 국제 표준 기구인 MPEG WG07에서는 GRASP-Net^[13], SparsePCGC^[14], OctSqueeze^[15] 등의 기술을 기반으로 AI-PCC의 표준화 방안에 대해 논의 중이다.

현재 논의 중인 AI-PCC 기술은 상호 그 유사성이 매우 높기에, 그중 GRASP-Net의 부호화기 구조를 소개함으로써 AI-PCC의 기본 개념을 설명한다. 먼저, 상기 [그림 3]에

서 보이는 바와 같이, 입력 포인트 클라우드(PC_{in})는 양자화(Quantization) 모듈을 거치며 포인트 클라우드의 비트 렵스(Bit Depth)가 줄어든다. 즉, 포인트 클라우드를 표현하는 점의 개수를 줄이며, 여러 개의 점이 하나의 점으로 대표되도록 만든다. 이는 포인트 클라우드의 기초적인 골격(PC_{Rough})만을 남기는 효과를 야기하며, PC_{Rough} 는 손실을 방지하기 위해 G-PCC의 무손실 압축 기술인 옥트리 부호화(Octree Encoding)를 통해 압축되어 러프 비트스트림(B_{Rough})으로 출력된다.

양자화(Quantization) 모듈을 통해 손실된 상세 정보는 뉴럴네트워크 기반의 모듈들을 통해 부호화되는데, 이는 포인트 분석(Point Analysis) 모듈과 피쳐 분석(Feature Analysis) 모듈로 구성되어 있다. PC_{Rough} 와 원본 포인트 클라우드의 기하 정보(Geometry Values)는 포인트 분석 모듈에 입력되어, PC_{Rough} 를 기하 정보에 대응시킨다. 포인트 분석 모듈은 대응된 PC_{in} 의 각 포인트별로 K-최근접 이웃(KNN, K-Nearest Neighbors) 탐색을 수행하고, PC_{in} 와 탐색된 이웃 포인트들을 점 위치 분석 뉴럴네트워크를 통해 하나의 F_{Point} (Point Feature)로 표현한다. 포인트 분석 모듈에서 출력된 PC_{Rough} 와 F_{Point} 는 피쳐 분석 모듈로 입력되며, 피쳐 분석 모듈은 2D 영상 압축에 활용되는 VAE와 유사하게 3D 컨볼루션 레이어(Convolutional Layer)를

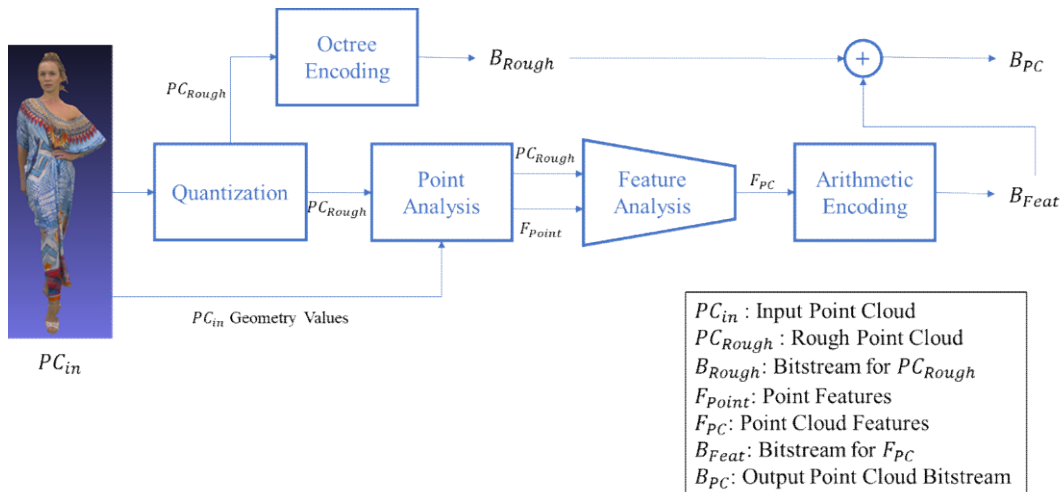


그림 3. GRASP-Net 부호화기의 구조도
 Fig. 3. GRASP-Net encoder architecture

통해 입력 데이터의 차원을 낮추는 작업을 수행한다. 따라서, 고차원의 입력 데이터가 낮은 차원을 갖는 적은 양의 데이터로도 표현이 가능해지며, 최종적으로 포인트 클라우드 피쳐(F_{PC})를 출력한다.

이후 F_{PC} 는 산술 부호화기(Arithmetic Encoder)를 통해 포인트 피쳐 비트스트림(B_{Feat})으로 압축된다. 압축된 B_{Feat} 는 사전에 옥트리 부호화 모듈에서 압축된 B_{Rough} 와 함께 합쳐져 포인트 클라우드 비트스트림(B_{PC})을 구성하며, B_{PC} 는 부호화기의 역순으로 구성된 GRASP-Net 복호화기를 통해 포인트 클라우드로 복원된다. GRASP-Net은 손실 압축 기법이기에 원본의 왜곡 정도와 압축된 B_{PC} 의 크기는 트레이드오프 관계에 있고, 뉴럴네트워크의 학습 시작 전에 설정한 하이퍼 파라미터에 따라 트레이드오프의 정도가 결정된다.

GRASP-Net 등의 AI-PCC 기술이 기존 포인트 클라우드 압축 기술에 비해 높은 압축 효율을 지녔음에도 불구하고, 전송 및 소비에 활용하기에 기존 기술에 비해 추가적인 비용이 요구되며 이는 [그림 4]를 통해 확인할 수 있다. [그림 4]에서 보이는 바와 같이, 전통적인 코덱에서는 품질을 결정하는 CP가 양자화 파라미터만이 존재하며, 동일한 코덱 구조 상에서 양자화 파라미터를 변경하여 품질을 결정할 수 있다. 하지만, AI-PCC 기술은 NNP 전체가 CP로 사용되므로 전통적인 코덱에 비해 굉장히 비대한 양의 CP가 요구된다. 이는, AI-PCC 기술의 특성상 학습을 통해 뉴럴네트워크가 완성되며, 이를 구성하는 NNP는 학습 전에 설정된 하이퍼 파라미터에 따라 서로 다른 성능을 내도록 학습되었기 때문이다. 그러나 [표 1]에서 보이는 바와 같이, AI-PCC에서 활용되는 NNP의 크기가 압축된 포인트 클라우드 비트스트림의 크기보다 훨씬 크기에, 전달해야하는

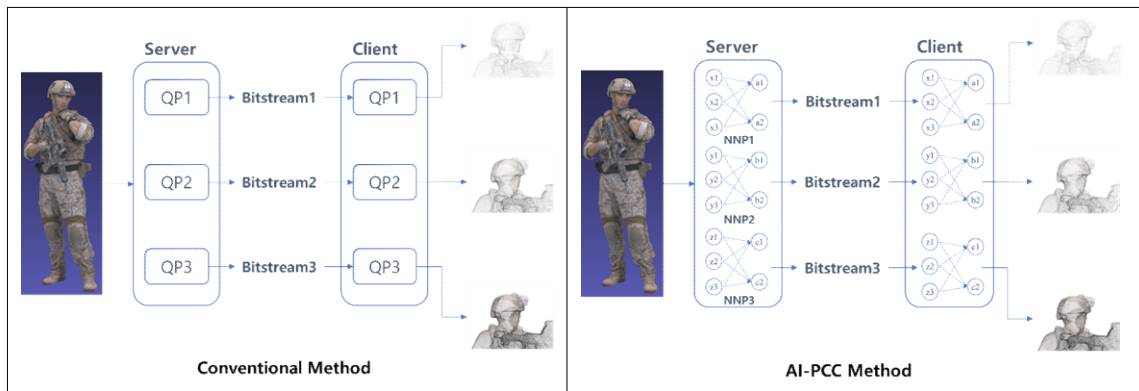


그림 4. 전통적인 기술과 AI-PCC 기술의 CP 비교
 Fig. 4. CP Comparison between the conventional method and AI-PCC method

표 1. 콘텐츠에 따른 크기 비교
 Table 1. Size comparison of different content

Content	Size (KB)
GRASP-Net NNP	1960
PCGCv2 NNP	3119
Longdress_vox10_1051 Point Cloud Frame	16813
Compressed Longdress_vox10_1051 using GRASP-Net (R01)	2.82
Compressed Longdress_vox10_1051 using GRASP-Net (R02)	10.87
Compressed Longdress_vox10_1051 using GRASP-Net (R03)	19.41
Compressed Longdress_vox10_1051 using GRASP-Net (R04)	32.43
Compressed Longdress_vox10_1051 using GRASP-Net (R05)	58.86

정보의 양은 증가하여 스트리밍 환경에서 활용하기에는 부적절하다는 한계가 있다.

III. 잔차 파라미터의 압축을 활용한 AI-PCC 기반 스트리밍 프레임워크

2장에서 기술된 배경 기술을 기반으로, 본 장에서는 INNC 기반 $NNP_{Residual}$ 압축을 활용한 AI-PCC 스트리밍 프레임워크를 설명한다. 서론과 2장에서 언급했듯 AI-PCC 기술의 활용에는 많은 양의 CP 생성이 뒤따르며, 이러한 CP는 압축된 포인트 클라우드 비트스트림보다 크기가 훨씬 크다는 것을 확인하였다. 따라서 포인트 클라우드 콘텐츠의 전송 및 소비를 지원하는 스트리밍 환경에서 AI-PCC 기술을 사용하기 위해 본 논문은 서버가 단말에게 NNP 가 아닌 $NNP_{Residual}$ 을 전송하여 CP의 크기를 줄이는 스트리밍 프레임워크인 IASF(INNC-based AI-PCC Streaming Framework)를 제안하며, IASF를 통해 더욱 효율적인 AI-PCC 기반 스트리밍을 지원하고자 한다.

본 논문에서 제안하는 IASF는 [그림 5] 및 [그림 6]과 같은 서버 및 단말로 구성되어 있다. 서버단의 기능을 NNP_{Base} 전송 과정과 $NNP_{Residual}$ 전송 과정으로 나누어 각각 3.1.1.절 및 3.1.2.절에서 설명하며, 단말의 NNP_{Base}

수신 과정과 $NNP_{Residual}$ 수신 과정은 각각 3.2.1절과 3.2.2.절에서 서술한다.

1. Server

IASF의 서버는 [그림 5]와 같이, 입력 NNP_n 의 n 값에 따라 전송 방법에 차이를 두며, 여기서의 n 은 NNP 가 업데이트된 횟수이다. 서버가 단말에 최초로 전송하는 NNP 인 $NNP_{Base}(NNP_0)$ 는 주황색의 NNC Encoder로 입력되며, 자세한 전송 과정은 3.1.1절에서 설명한다. 또한 업데이트가 존재하는 $NNP_{Updated}$ 은 녹색의 INNC Encoder로 입력되며, 3.1.2절에서 자세히 설명한다.

1.1 NNP_{Base} transmission

NNP_{Base} 를 전송할 시 IASF의 동작 흐름은 다음과 같다. IASF의 입력 데이터는 NNP 이며, 최초 NNP 는 NNP_{Base} 라고 칭한다. 서버가 단말에게 최초로 NNP 를 전송하는 경우에는 단말이 사전에 소지한 NNP 가 없기에, NNP_{Base} 는 원본 NNP 를 압축하는 NNC(edition 1) 부호화 모듈(NNC Encoder)을 통해 압축되어야 한다. 해당 모듈에서 NNP_{Base} 는 B_{NNC} (NNC Bitstream)로 압축된 후 두개의 모듈에 입력되는데, NNC 복호화 모듈(NNC Decoder)과 멀티플렉싱 모

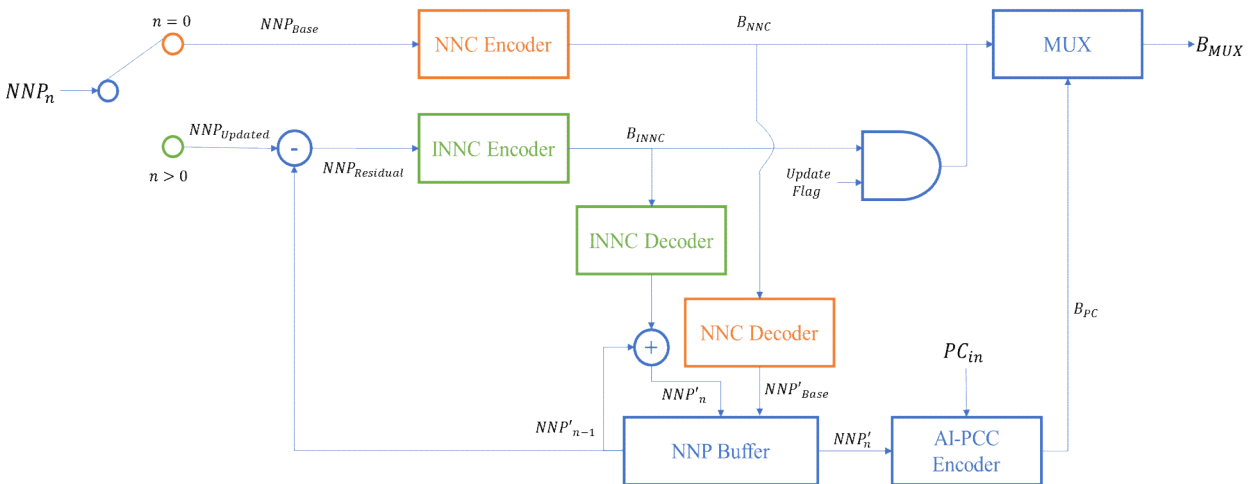


그림 5. IASF의 서버 단 구조도
 Fig. 5. Architecture for server side of IASF

들(MUX)이 그에 해당한다.

먼저, NNC 복호화 모듈에 입력되는 B_{NNC} 는 해당 모듈에서 복호화 되어 NNP'_{Base} 로 복원된다. 여기서 NNP'_{Base} 는 손실 압축기인 NNC 부-복호화기를 거쳤기 때문에 NNP_{Base} 와 비교하여 왜곡이 존재하는 상태이다. 이러한 NNP'_{Base} 는 NNP의 저장공간인 NNP 버퍼 모듈(NNP Buffer)에 저장된다. NNP'_{Base} 가 저장되는 이유는 이후에 서버단에서 NNP의 업데이트가 발생할 경우, 그에 대한 $NNP_{Residual}$ 의 계산에 이용되기 때문이다. 이러한 NNP'_{Base} 는 NNP 버퍼 모듈에 입력됨과 동시에 포인트 클라우드 압축 모듈인 AI-PCC 모듈에 입력된다. AI-PCC 모듈에서는 NNP'_{Base} 를 기반으로 AI-PCC 뉴럴네트워크가 완성되며, 이를 통해 입력 포인트 클라우드인 PC_{in} 을 압축하는 작업이 진행된다. 이에 따라 PC_{in} 은 AI-PCC 모듈에서 포인트 클라우드 비트스트림인 B_{PC} 로 압축되며, 이러한 B_{PC} 는 MUX 모듈로 입력된다. MUX 모듈에는 NNC 부호화 모듈을 통해 압축된 B_{NNC} 와 B_{PC} 가 함께 입력되며, 두개의 비트스트림 모두 단말에게 전송되어야 하기에 MUX 모듈은 B_{NNC} 와 B_{PC} 를 멀티플렉싱하여 B_{MUX} 를 생성한다. 결과적으로 B_{MUX} 는 NNP'_{Base} 의 정보와 PC_{in} 의 정보를 모두 담고 있으며, 이는 서버에서 단말로 전송되는 최종 비트스트림이다.

1.2 $NNP_{Residual}$ transmission

단일 프레임이 아닌 다중 프레임의 포인트 클라우드 시퀀스를 전송하는 상황에서는, 서버가 PC_{in} 프레임을 압축할 때에 이전 프레임 압축에 활용한 NNP를 동일하게 사용하지 않을 수 있다. 이는 PC_{in} 의 특성이 크게 바뀌거나 단말이 네트워크 상황에 따라 다른 크기의 비트스트림을 요구할 때에, 2장에서 언급한 AI-PCC 기술의 특성상 다른 방식으로 학습된 새로운 NNP를 통해 기존 NNP를 대체하기 때문이다. IASF는 이러한 경우에 잔차 파라미터인 $NNP_{Residual}$ 을 사용하며, 본 절에서는 $NNP_{Residual}$ 을 전송할 시 IASF의 동작 흐름을 기술한다.

[그림 5]의 구조도에서 입력되는 NNP_n 은 NNP 버퍼 모

듈에 저장되어있던 NNP'_{n-1} 과 뺄셈 연산이 수행되어 $NNP_{Residual}$ 이 생성되는데, 여기서 NNP_{n-1} 이 아닌 왜곡이 존재하는 NNP'_{n-1} 를 활용하는 이유는 단말 또한 왜곡된 NNP'_{n-1} 을 소지하고 있기 때문이다. 뺄셈 연산을 통해 획득된 $NNP_{Residual}$ 은 0의 분포 비율이 높은 잔차 파라미터 이기에 잔차 파라미터 압축에 최적화된 INNC 부호화 모듈(INNC Encoder)에 입력되어 B_{INNC} 로 압축된다.

B_{INNC} 는 3.1.1절에서 설명한 B_{NNC} 와 유사하게 두가지의 모듈로 동시에 입력되는데, 첫번째는 INNC 복호화 모듈(INNC Decoder)이며, 다른 하나의 모듈은 MUX 모듈이다. MUX 모듈 입력 시 B_{NNC} 와 달리 B_{INNC} 는 논리 회로 중 AND 게이트를 통과하게 된다. 해당 AND 게이트는 업데이트 플래그(Update Flag)와 B_{INNC} 를 입력 값으로 받는데, 업데이트 플래그는 NNP의 업데이트 존재 유무를 나타내는 플래그이다. 업데이트 플래그가 0이라면, AND 게이트를 통해 B_{INNC} 는 MUX 모듈로 전달되지 않게 되며, 플래그가 1일 경우($n>0$)에는 MUX 모듈로 전달된다.

INNC 복호화 모듈로 입력된 B_{INNC} 는 $NNP'_{Residual}$ 과 같이 왜곡이 있는 형태로 출력되며, 이는 NNP 버퍼에 이미 존재했던 NNP'_{n-1} 과 덧셈 연산을 거친다. 덧셈 연산을 통해 NNP'_n 이 생성되며, 이러한 NNP'_n 은 3.1.1절과 동일하게 PCC 모듈과 NNP 버퍼 모듈에 각각 입력된다. NNP 버퍼 모듈에 입력된 NNP'_n 은 추후 업데이트될 NNP_{n+1} 과의 뺄셈 연산을 위해 NNP 버퍼에 저장되고, PCC 모듈에 입력된 NNP'_n 은 PC_{in} 에 대한 포인트 클라우드 압축에 활용된다. PCC 모듈에서 출력된 B_{PC} 는 사전에 생성된 B_{INNC} 와 함께 MUX 모듈에 입력되며, MUX 모듈은 최종적으로 B_{MUX} 를 생성하여 이를 단말에게 전송한다. 단, 업데이트 플래그가 0인 경우에는 B_{INNC} 가 MUX 모듈에 입력되지 않기에 이때의 B_{MUX} 는 B_{PC} 와 NNP의 업데이트가 없음을 알리는 업데이트 플래그의 정보만을 담게 된다.

2. Client

[그림 6]에서 확인할 수 있듯이 단말 단에서의 B_{MUX} 수신 과정은 서버단의 전송 과정의 역순과 유사하다. 입력된

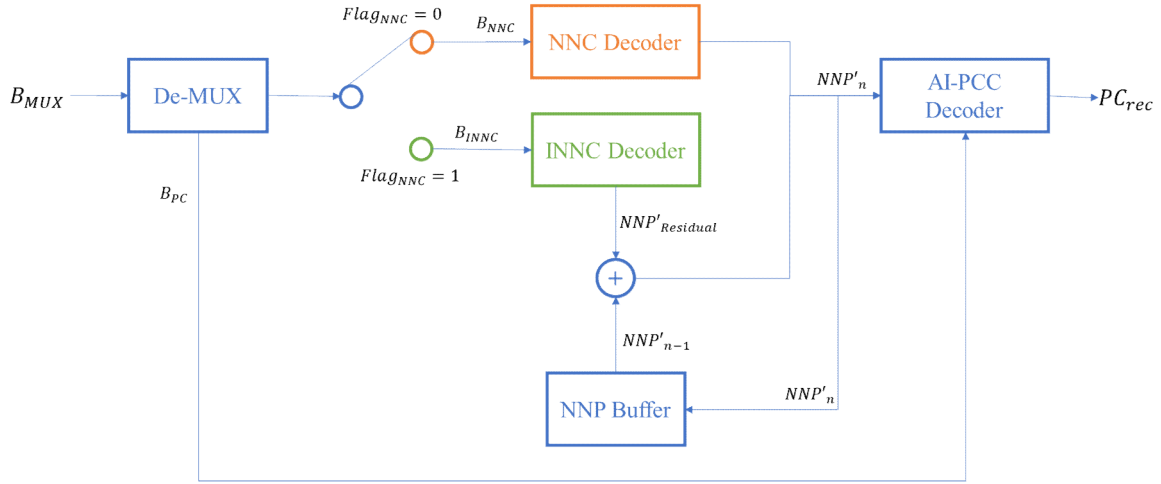


그림 6. IASF의 단말 단 구조도
 Fig. 6. Architecture for client side of IASF

B_{MUX} 는 디멀티플렉싱 모듈(De-MUX)을 통해 B_{PC} 와 NNP 정보를 담은 비트스트림으로 분리되며, B_{PC} 는 AI-PCC 복호화 모듈(AI-PCC Decoder)에 입력된다. NNP 정보를 담은 비트스트림은 B_{NNC} 혹은 B_{INNC} 에 해당하는데, 두 비트스트림 모두 헤더(Header)에 $Flag_{NNC}$ 정보가 존재한다. $Flag_{NNC}$ 가 0일 경우에는 비트스트림이 NNC 부호화 모듈을 통해 압축된 B_{NNC} 이며, 반대로 1일 경우에는 INNC 부호화 모듈을 통해 압축된 B_{INNC} 임을 의미한다. 이러한 $Flag_{NNC}$ 정보는 디멀티플렉싱 모듈에서 미리 파악하여 $Flag_{NNC}$ 가 0일 경우 주황색 스위치로 B_{NNC} 를 입력하며, $Flag_{NNC}$ 가 1일 경우에 녹색 스위치로 입력한다. 주황색 스위치로 입력된 B_{NNC} 의 처리 과정은 3.2.1절에서, 녹색 스위치로 입력된 B_{INNC} 의 처리 과정은 3.2.2절에서 설명한다.

2.1 B_{NNC} reception

B_{NNC} 를 수신한 경우에 단말은 먼저 NNC 복호화 모듈(NNC Decoder)을 통해 B_{NNC} 를 복호화한다. 서버단에서 NNC 부호화 모듈을 통해 압축된 NNP는 NNP_{Base} 가 유일하므로, NNC 복호화 모듈의 출력 데이터는 NNP_{Base} 이다 ([그림 6]에는 NNC 복호화 모듈의 출력 데이터를 NNP'_n

으로 나타냈으며, 이때의 NNP'_n 은 NNP'_0 이다). 3.1.1절에서도 언급을 했듯, NNC 코덱은 데이터 손실을 동반하기에 단말이 획득하는 NNP'_{Base} 는 서버단에 처음 입력된 NNP_{Base} 와 비교하여 데이터의 왜곡이 존재한다. 이러한 NNP'_{Base} 는 NNP 버퍼 모듈(NNP Buffer)과 AI-PCC 복호화 모듈(AI-PCC Decoder)에 입력된다. NNP 버퍼에 저장되는 이유는 추후 수신하게 될 $NNP'_{Residual}$ 과의 덧셈 연산을 통해 NNP'_1 을 생성하기 위함이며, AI-PCC 복호화 모듈에 입력되는 이유는 AI-PCC 뉴럴네트워크의 NNP로써 활용하기 위함이다. NNP'_{Base} 로 채워진 AI-PCC 뉴럴네트워크는 AI-PCC 복호화 모듈에 입력된 B_{PC} 를 복호화하는 데에 활용되며, 최종적으로 AI-PCC 복호화 모듈로부터 복원 포인트 클라우드인 PC_{rec} 이 출력된다.

2.2 B_{INNC} reception

단말이 B_{INNC} 를 수신한 경우에는 이를 INNC 복호화 모듈(INNC Decoder)로 입력하며, INNC 복호화 모듈은 입력 받은 B_{INNC} 를 복호화하여 $NNP'_{Residual}$ 을 출력한다. 이와 같은 $NNP'_{Residual}$ 은 NNP 버퍼 모듈에 이미 저장된 NNP'_{n-1} 과의 덧셈 연산을 통해 NNP'_n 의 복원에 기여한다. 복원된 NNP'_n 은 두가지 모듈에 입력되는데, NNP 버퍼 모

들과 AI-PCC 복호화 모듈이 이들에 해당한다. 3.2.1절의 NNP'_{Base} 와 마찬가지로 NNP'_n 또한 NNP 버퍼에 저장되며, 이는 추후 입력될 $NNP'_{Residual}$ 과의 덧셈을 통해 NNP'_{n+1} 을 생성하기 위함이다. AI-PCC 복호화 모듈에 입력된 NNP'_n 은 모듈 내 AI-PCC 뉴럴네트워크의 기존 $NNP(NNP'_{n-1})$ 을 대체(업데이트)하며, 이를 통해 업데이트된 AI-PCC 뉴럴네트워크는 De-MUX 모듈로부터 출력된 B_{PC} 를 복호화하는 작업을 진행하며, 최종적인 AI-PCC 복호화 모듈의 출력 데이터는 PC_{rec} 이다.

IV. 실험 결과

3장에서는 2장에서 설명한 INNC 부호화기를 기반으로 잔차 파라미터를 압축하여 더욱 효율적인 AI-PCC 스트리밍을 가능케 하는 ISAF라는 프레임워크를 제안하였다. 본 장에서는 3장에서 설명한 본 논문의 ISAF를 구현하여 이에 대한 결과를 확인 및 검증한다. 설계한 프레임워크는 Ubuntu 20.04 환경에서 Python 언어를 사용하여 구현하였다.

실험은 [그림 5]와 [그림 6]의 프레임워크를 따랐으며, AI-PCC 기술로는 GRASP-Net을 활용하였다. NN_{Base} 를 GRASP-Net R04(Rate 4)로 설정하고 $NN_{Updated}$ 은 GRASP-Net R05(압축된 포인트 클라우드의 비트스트림이 가장 크지만, 가장 고품질의 포인트 클라우드 복원을 지원하는 뉴럴네트워크)로 설정하였다. 여기서 GRASP-Net R04와 R05는 동일 뉴럴네트워크 형태를 가지고 있으며, NNP만이 다르기에 서로 간의 $NNP_{Residual}$ 획득이 가능하다. ISAF의 성능 평가를 위해 앵커(Anchor)는 GRASP-Net R05의 NNP를 NNC 부호화기를 통해 압축한 결과로 설정하였다. 추가적으로, 성능 평가의 참고를 위해 NNP 압축을 거치지 않은 원본 GRASP-Net R01부터 R05까지의 결과 또한 그래프에 삽입하였다. 압축 대상으로 활용한 포인트 클라우드는 MPEG CTC^[16]에서 활용되는 8i의 “Longdress”, “Soldier”, “Loot”, 그리고 “Red and Black” 시퀀스의 각 첫 번째 프레임이다.

현재 GRASP-Net 기술은 포인트 클라우드의 위치 정보

의 압축만을 지원하기 때문에, 포인트 클라우드의 복원 품질 평가 또한 MPEG WG7에서 활용하는 위치 정보의 평가 지표인 D1 PSNR(Point-to-Point PSNR)과 D2 PSNR(Point-to-Plane PSNR)을 활용하였으며, 그 결과는 [그림 7]과 같다. PSNR 값들은 그래프의 Y축을 나타낸다. X축에 해당하는 값은 CP와 B_{PC} 의 합에 해당하는데, 여기서 CP는 기술 별로 다르게 표현된다. ISAF의 경우 CP는 B_{INNC} 이며 이에 따라 X축은 B_{MUX} 의 크기와 동일하다. 앵커(Anchor)의 CP 크기는 GRASP-Net R05의 NNP를 NNC 부호화기를 통해 압축한 B_{NNC} 의 크기이며, 이는 3.1.1절에서 NNP_{Base} 를 압축하는 과정과 동일하게 구현하였다. 각각의 압축에 활용한 양자화 파라미터(QP)는 [표 2]에서 확인할 수 있다. NNC 및 INNC에서 활용되는 QP는 전부 음수이며, 절대값이 클수록 데이터 손실이 적다. 추가적으로, 참고용으로 삽입한 원본 GRASP-Net R01~R05의 경우에는 CP를 아무런 압축 과정을 거치지 않은 NNP의 크기(뉴럴네트워크 파이토치 파일(.pth)의 원본 크기)로 상정하였다.

[그림 7]을 통해 제안 기술을 활용할 시 압축을 거치지 않는 CP 전송 방법과 비교하면 전송 시 요구되는 CP의 데이터 크기가 현저히 줄어드는 것을 확인할 수 있으며, NNC 표준 부호화기를 기반으로 압축된 CP 전송과 비교하여도 필요 데이터 크기가 적다는 것을 알 수 있다. 그러나 높은 압축률을 달성하지만 복원 포인트 클라우드의 PSNR이 낮다는 한계 또한 [그림 7]을 통해 확인할 수 있다.

[그림 8]은 [그림 7]에서 각 그래프의 PSNR이 가장 높은 지점을 기준으로 그에 따른 복원 포인트 클라우드의 렌더링 결과를 나타낸 것이며, 이와 함께 CP의 크기와 D1 PSNR 결과를 확인할 수 있다. [그림 8]를 통해 객관적인 지표인 D1 PSNR 외에도 주관적인 평가를 할 수 있으며, 주관적 평가에서도 INNC 기술을 활용한 경우에 왜곡이 더욱 심하게 발생함을 확인할 수 있다. 이는 INNC 부호화기가 SBTQ 등 압축 효율을 극대화하기 위한 기술을 사용하기 때문이다. 그러나 IASF를 통해 $NNP_{Residual}$ 의 압축을 활용할 경우에 전송하는 CP의 크기(B_{INNC})는 무손실 NNP와 비교하여 약 0.3%에 달하며, B_{NNC} 와 비교하여도 1.3%에 달하기에 왜곡이 발생하는 정도에 비해 얻는 압축 효율

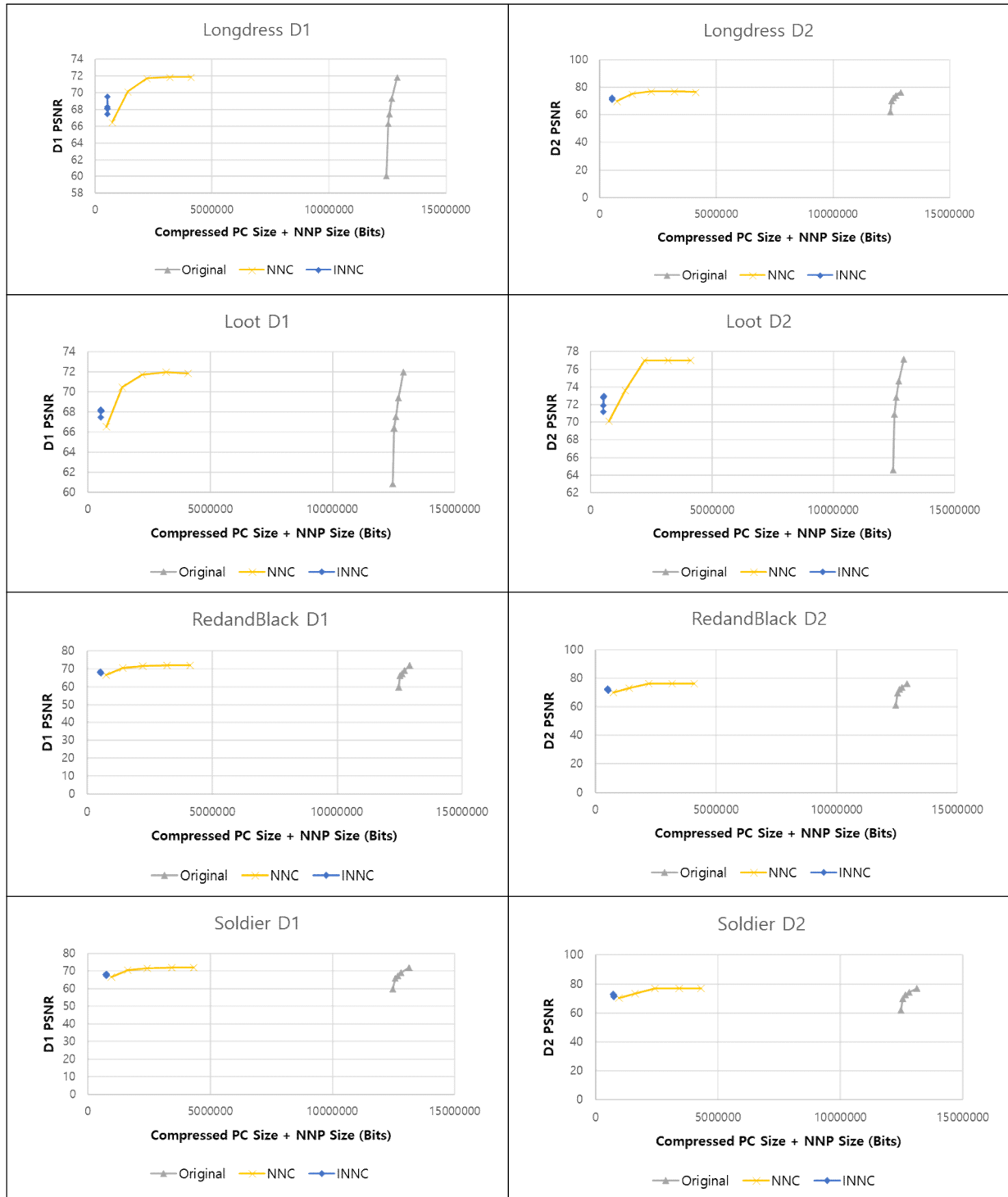


그림 7. IASF 구현에 대한 실험 결과
 Fig. 7. Experimental results for IASF implementation

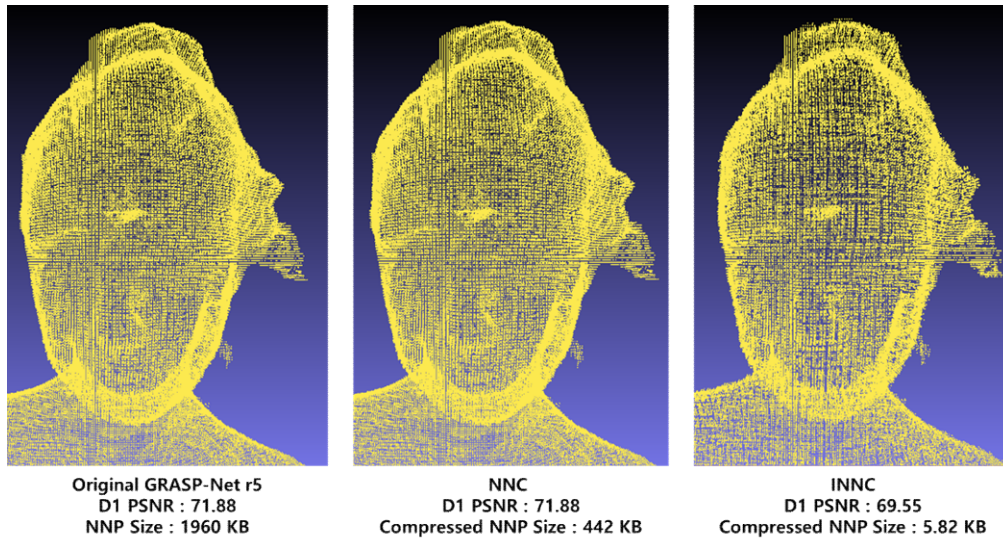


그림 8. 주관적 평가를 위한 복원 포인트 클라우드 렌더링 결과
 Fig. 8. Reconstructed point cloud rendering results for subjective evaluation

표 2. 압축 대상에 따른 부호화기 및 QP
 Table 2. Encoder type & QP based on compression target

Compression Target	NNC Encoder QP	Compression Target	INNC Encoder QP
Grasp-Net R05 NNP	-10	Grasp-Net R05 - R04 ($NNP_{Residual}$)	-2
	-20		-4
	-30		-6
	-40		-8
	-50		-10

이 매우 크다. 또한, [그림 8]에서 확인할 수 있듯 INNC 기반 압축을 통해 발생한 왜곡은 전체적인 형태가 일그러지는 것이 아닌, 점 간의 거리가 멀어져 밀도가 낮아지는 방향으로 발생한다. 이러한 형태의 왜곡은 단말 단에서 인터폴레이션(Interpolation) 등의 후처리 기술을 활용하여 점 사이의 빈 공간을 메워 줌으로써 해결할 수 있을 것으로 사료된다.

V. 결론

본 논문에서는 AI-PCC 기술을 스트리밍 환경에서 사용할 경우에 한계가 되는 큰 크기의 CP를 극복하기 위해, NNP가 아닌 $NNP_{Residual}$ 을 압축하여 전송할 수 있는 프레임워크인

IASF(INNC-based AI-PCC Streaming Framework)를 제안하였다. 이와 같은 IASF는 NNP 간의 잔차 파라미터인 $NNP_{Residual}$ 을 최초로 활용했다는 점에서 의의가 있으며, $NNP_{Residual}$ 에 대한 압축 효율을 최대화하기 위해 최첨단 기술인 INNC 부호화기를 최초로 스트리밍 환경에 적용했다는 점에서도 그 의의를 찾을 수 있다.

IASF는 NNP_{Base} 를 전송하는 단계와 $NNP_{Residual}$ 를 전송하는 두 단계로 나뉜다. 최초로 활용되는 NNP인 NNP_{Base} 는 NNC 부호화기를 기반으로 압축되어 B_{PC} 와 함께 B_{MUX} 로써 단말에게 전송되며, 단말은 B_{MUX} 를 수신하여 NNP'_{Base} 를 복원한 뒤 이를 통해 PC_{rec} 을 복원한다. 이후 NNP의 업데이트가 발생하는 경우 서버는 NNP_n 와 NNP'_{n-1} 간의 $NNP_{Residual}$ 을 생성하며, $NNP_{Residual}$ 은

INNC 부호화기를 통해 B_{INNC} 로 압축된다. B_{INNC} 는 B_{NNC} 와 동일하게 B_{PC} 와 멀티플렉싱되어 단말에게 전송되며, 단말은 $NNP'_{Residual}$ 을 복원한 뒤 이를 NNP'_{n-1} 에 더하여 NNP'_n 을 완성시키며, NNP'_n 을 통해 PC_{rec} 을 복원한다. 이러한 제안기술을 검증하기 위해 본 논문은 GRASP-Net을 활용하여 IASF를 구현하였고 4장에서 실험 결과를 검증하였다.

4장에서 검증된 바와 같이, IASF를 통해 $NNP_{Residual}$ 을 활용할 경우 기존 방식 대비 약 1.3%에 달하는 적은 양의 CP가 발생하였다. 따라서 IASF는 기존 방식 대비 효율적인 AI-PCC 기반 스트리밍을 지원한다고 할 수 있다. 특히, IASF는 한번의 NNP 업데이트만을 가정한 4장에서의 실험과 달리 여러 번의 NNP 업데이트가 존재하는 경우에 더욱 큰 효율을 낼 수 있다는 점을 감안하면 AI-PCC 기반 스트리밍 프레임워크로서 매우 적합하다고 할 수 있다. 그러나 제안 기술은 CP의 압축에 대해 높은 압축률을 달성하였으나, 포인트 클라우드의 복원 품질은 개선될 필요가 있음이 확인되었다. 이러한 제한 사항은 인터플레이션 (Interpolation) 등의 후처리 기능을 추후 연구에 추가함으로써 극복할 수 있을 것으로 사료된다.

참 고 문 헌 (References)

- [1] J. Kim, S. Rhee, H. Kwon and K. Kim, "LiDAR Point Cloud Compression by Vertically Placed Objects Based on Global Motion Prediction," in IEEE Access, vol. 10, pp. 15298-15310, 2022. doi: <https://doi.org/10.1109/ACCESS.2022.3148252>.
- [2] M. Lee, et al., "Point Cloud Video Codec using 3D DCT based Motion Estimation and Motion Compensation," 2021 Journal of Broadcast Engineering, Vol 26, No. 6, pp.680-690, November 2021. doi: <https://doi.org/10.5909/JBE.2021.26.6.680>.
- [3] R. Mekuria and L. Bivolarsky, "Overview of the MPEG Activity on Point Cloud Compression," 2016 Data Compression Conference (DCC), Snowbird, UT, USA, 2016, pp. 620-620. doi: <https://doi.org/10.1109/DCC.2016.91>.
- [4] "Performance analysis of currently AI-based available solutions for PCC," ISO/IEC JTC1/SC29/WG7, N582, Apr. 2023.
- [5] X. Hou, L. Shen, K. Sun and G. Qiu, "Deep Feature Consistent Variational Autoencoder," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 2017, pp. 1133-1141, doi: <https://doi.org/10.1109/WACV.2017.131>.
- [6] H. Liu, H. Yuan, Q. Liu, J. Hou and J. Liu, "A Comprehensive Study and Comparison of Core Technologies for MPEG 3-D Point Cloud Compression," in IEEE Transactions on Broadcasting, vol. 66, no. 3, pp. 701-717, Sept. 2020. doi: <https://doi.org/10.1109/TBC.2019.2957652>.
- [7] M. Lee, et al., "[NNC] Compression performance of NNCodec on AI-PCC," ISO/IEC JTC1/SC29/WG4, M64314, Jul. 2023.
- [8] "Draft White Paper on Neural Network Coding (Ed2.)," ISO/IEC JTC1/SC29/AG3, N122, Jul. 2023.
- [9] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, May 2020. doi: <https://doi.org/10.1109/MSP.2020.2975749>.
- [10] S. Verdu, "Fifty years of Shannon theory," in IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2057-2078, Oct. 1998. doi: <https://doi.org/10.1109/18.720531>.
- [11] M. Lee, S. Rhee and K. Kim, "Variance based Averaging and Standard Normal Distribution for Improvement of SBT Quantization in Neural Network Coding," 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS), Ise, Japan, 2022, pp. 1-6. doi: <https://doi.org/10.1109/SCISISIS55246.2022.10001990>.
- [12] P. Haase et al., "[NNR] CE1-CE3: Report on HHI Results for Incremental Neural Network Compression," ISO/IEC JTC1/SC29/WG4, M57416, Jul. 2021.
- [13] J. Pang et al., "GRASP-Net: Geometric Residual Analysis and Synthesis for Point Cloud Compression," in APCCPA '22: Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis, Oct. 2022, pp. 11-19. doi: <https://doi.org/10.1145/3552457.3555727>.
- [14] J. Wang, D. Ding, Z. Li and Z. Ma, "Multiscale Point Cloud Geometry Compression," 2021 Data Compression Conference (DCC), Snowbird, UT, USA, 2021, pp. 73-82. doi: <https://doi.org/10.1109/DCC50243.2021.00015>.
- [15] L. Huang, S. Wang, K. Wong, J. Liu and R. Urtasun, "OctSqueeze: Octree-Structured Entropy Model for LiDAR Compression," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 1310-1320. doi: <https://doi.org/10.1109/CVPR42600.2020.00139>.
- [16] "Common Test Conditions for G-PCC," ISO/IEC JTC1/SC29/WG7, N578, Apr. 2023.

저 자 소 개



이 민 석

- 2021년 8월 : 경희대학교 전자공학과 공학사, 컴퓨터공학과 공학사
- 2021년 9월 ~ 현재 : 경희대학교 전자정보융합공학과 석사과정
- ORCID : <http://orcid.org/0000-0001-6500-7156>
- 주관심분야 : 영상처리, 멀티미디어 시스템, 딥러닝모델 경량화



김 준 식

- 2017년 2월 : 경희대학교 전자·전파공학과 공학사
- 2019년 2월 : 경희대학교 전자공학과 공학석사
- 2022년 8월 : 경희대학교 전자정보융합공학과 공학박사
- 2022년 5월 ~ 현재 : 한국정보통신기술협회 소프트웨어시험인증연구소 선임연구원
- ORCID : <https://orcid.org/0000-0002-0287-9640>
- 주관심분야 : 멀티미디어 시스템, 포인트 클라우드 압축, 디지털 대화형 방송



이 성 배

- 2019년 2월 : 경희대학교 전자공학과 공학사
- 2021년 2월 : 경희대학교 전자정보융합공학과 공학석사
- 2021년 3월 ~ 현재 : 경희대학교 전자정보융합공학과 박사과정
- ORCID : <https://orcid.org/0000-0002-7392-6870>
- 주관심분야 : 딥러닝, 영상처리, 멀티미디어 시스템



김 규 현

- 1989년 2월 : 한양대학교 전자공학과 공학사
- 1992년 9월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학석사
- 1996년 7월 : 영국 University of Newcastle upon Tyne 전기전자공학과 공학박사
- 1996년 ~ 1997년 : 영국 University of Sheffield, Research Fellow
- 1997년 ~ 2006년 : 한국전자통신연구원 대화형미디어연구팀장
- 2006년 ~ 현재 : 경희대학교 전자정보대학 교수
- ORCID : <http://orcid.org/0000-0003-1553-936X>
- 주관심분야 : 영상처리, 멀티미디어통신, 디지털 대화형방송