

Development of Real-time Point Cloud Compression and Transmission System

Kyohei Unno†, Yoshitaka Kidani† and Tomohiro Tsuji††

Abstract Point clouds are attracting increasing attention as a form of photorealistic 3D data expression. Point clouds can represent a wide variety of objects, from people and other objects to large-scale scenes such as buildings and cities, but the amount of data they produce is enormous, making compression technology essential for their utilization. In response to this, MPEG was the first in the world to internationally standardize point cloud compression technology. The standardized V-PCC and G-PCC have applications set according to the density and scale of point clouds, respectively. While real-time encoders and real-time decoders compliant with these standards existed, there was no system that could process both, i.e., compressed transmission, in real time. Therefore, the authors developed a system that complies with V-PCC and G-PCC. We have developed the world's first end-to-end real-time compression transmission system.

Using this system, we demonstrated the world's first real-time point cloud compression and transmission over a commercial network.

Keywords: point cloud, MPEG, V-PCC, G-PCC, real-time transmission, demonstration experiment

1. Introduction

In recent years, advancements and cost reductions in acquisition technologies such as sensors have driven increased utilization of 3D data. While non-photorealistic formats like human avatars and building models were the mainstream for 3D data, there is now growing demand for photorealistic formats capable of representing high-density point clouds and meshes. Among these, point clouds can represent diverse objects and are attracting attention across various fields. For example, dynamic point clouds of people are being explored for entertainment applications like the metaverse and social networking services (SNS). Furthermore, point clouds are highly compatible with digitizing large-scale 3D objects like buildings and cities, as well as entire 3D scenes. Consequently, their use is anticipated across a broad spectrum of fields, including construction, civil engineering, autonomous driving, disaster response, and building digital twin infrastructure for these applications.

A point cloud is a format that represents 3D objects as a collection of points, as shown in Figure 1. Each point possesses geometric information (Geometry), indicating its position coordinates in 3D space. Furthermore, depending on the application and the method of point cloud acquisition, each point may also carry attribute information (Attribute), such as color or reflectance coefficient.

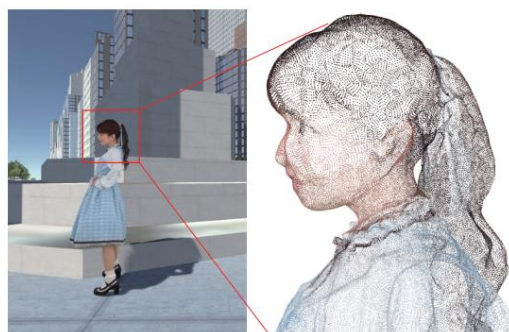


Figure 1. Example of a 3D point cloud

The number of points in a point cloud varies significantly depending on the application. For example, a dynamic point cloud of a person currently contains tens of millions of points per second. In contrast, a static point cloud of a large urban area can contain hundreds of millions to tens of billions of points per scene. Since each point carries the aforementioned Geometry and Attribute information, the volume of point cloud data becomes enormous. Therefore, point cloud encoding technology (point cloud compression technology), which efficiently reduces data volume during storage and transmission, is indispensable for the distribution of point cloud data. In response to the growing societal demand for point clouds and the challenges posed by their data volume during distribution, MPEG (Moving Picture Experts Group)—known for developing international standards for encoding audio, acoustic, and video data—has published the world's first international standard encoding method (compression method) specifically for point cloud data: V-PCC (Video-based Point Cloud Compression) 1) and G-PCC (Geometry-based Point Cloud Compression) 2). V-PCC converts 3D point clouds into image data by projecting them onto a 2D plane,

2024年10月22日受付, 2024年12月26日再受付, 2025年2月10日採録

† 株式会社 KDDI 総合研究所 先端技術研究所 XR 部門

(〒 356-8502 埼玉県ふじみ野市大原 2-1-15, TEL 049-278-7441)

†† KDDI 株式会社 ビジネス事業本部 プロダクト本部 DX・IoTサービス企画部

(〒 105-6490 東京都港区虎ノ門 1-17-1 虎ノ門ヒルズビジネスタワー, TEL 03-3347-0077)

[Copyrights to Machine-Translated Content]

The copyright of the original papers published on this website belongs to the Institute of Image Information and Television Engineers. Unauthorized use of original papers or translated papers is prohibited. Please be sure to cite the original publication when referencing.

For details, please refer to the copyright regulations of the Institute of Image Information and Television Engineers.

リアルタイム点群圧縮伝送システムの開発

Development of Real-time Point Cloud Compression and Transmission System

正会員 海野 恭平[†], 正会員 木谷 佳隆[†], 正会員 辻 智弘^{††}

Kyohei Unno[†], Yoshitaka Kidani[†] and Tomohiro Tsuji^{††}

あらまし フォトリアルな三次元データ表現形式の一つとして、点群への注目が高まっている。点群は人物などの物体から建造物や都市などの大規模シーンまで多様な対象物を表現できる一方そのデータ量は膨大であり、利活用には圧縮技術が不可欠である。これに対して、MPEG は世界で初めて点群圧縮技術を国際標準化した。標準化された V-PCC と G-PCC には、それぞれ点群の密度とスケールに応じた用途が設定されている。これらの標準に準拠したリアルタイムエンコーダあるいはリアルタイムデコーダは存在していたが、これら両方すなわち圧縮伝送をリアルタイム処理できるシステムは存在しなかった。そこで、筆者らは V-PCC および G-PCC に準拠した End-to-End のリアルタイム圧縮伝送システムを世界で初めて開発した。さらに、V-PCC と G-PCC の両方において、同システムを用いて世界初の商用網上でリアルタイム点群圧縮伝送を実証した。

キーワード：点群, MPEG, V-PCC, G-PCC, リアルタイム伝送, 実証実験

1. ま え が き

近年、センサ等の取得技術の進歩や低廉化に伴い、三次元データの利活用が進んでいる。従来の三次元データは人物の-avatarや、建造物の三次元モデルなど非フォトリアルなフォーマットが主流であったが、近年では高密度な点群 (Point Cloud) やメッシュなどによるフォトリアルな三次元データを表現可能なフォーマットへの期待が高まっている。その中で点群は多様な対象物が表現でき、様々な分野で注目されている。例えば、メタバースや SNS 等のエンターテインメント向けには人物の動的な点群の活用が検討されている。また、点群は建造物や都市などの大規模な三次元物体、三次元シーンのデータ化とも親和性が高いため、建設、土木、自動運転、災害対応やそれらの応用に向けたデジタルツイン基盤の構築等の幅広い分野での活用が期待されている。

点群とは、図 1 に示すように点の集合で三次元物体を表現するフォーマットであり、点ごとに三次元空間上の位置座標を示す幾何情報 (Geometry) を持つ。さらに、アプリケーションや点群の取得方法に応じて、点ごとに色や反射係数といった属性情報 (Attribute) を持つ。点

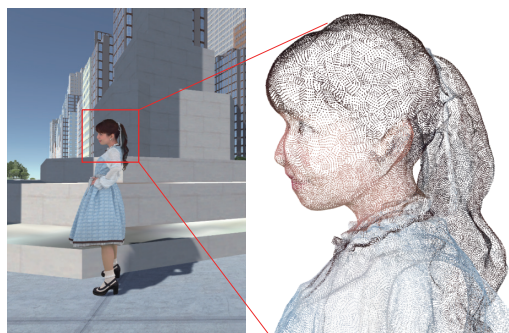


図 1 三次元点群の例

群の点数は、例えば人物の動的な点群であれば現状は秒間で数千万点程度、広域な都市の静的な点群であれば一つのシーンで数億点から数千億点程度となる。これらの 1 点ずつに上記の Geometry と Attribute の情報を持つため、点群のデータ量は膨大となる。よって、点群データの流通には保存、および伝送時のデータ量を効率的に削減する点群符号化技術 (点群圧縮技術) が不可欠である。

このような社会における点群のニーズの高まり、および流通時のデータ量に対する課題に基づいて、音声・音響データや映像データの符号化方式の国際標準の開発団体で知られる MPEG (Moving Picture Experts Group) は、点群データに特化した世界初の符号化方式 (圧縮方式) の国際標準である V-PCC (Video-based Point Cloud Compression)¹⁾ と G-PCC (Geometry-based Point Cloud Compression)²⁾ を発行した。V-PCC は三次元点群を二次元平面に投影することで画像データに変換し、変換したデー

2024 年 10 月 22 日受付, 2024 年 12 月 26 日再受付, 2025 年 2 月 10 日採録

[†] 株式会社 KDDI 総合研究所 先端技術研究所 XR 部門

(〒 356-8502 埼玉県ふじみ野市大原 2-1-15, TEL 049-278-7441)

^{††} KDDI 株式会社 ビジネス事業本部 プロダクト本部 DX・IoT サービス企画部

(〒 105-6490 東京都港区虎ノ門 1-17-1 虎ノ門ヒルズビジネスタワー, TEL 03-3347-0077)

V-PCC is a method of encoding 3D point clouds using video coding technologies such as H.265 | HEVC3) and H.266 | VVC4). Due to the characteristics of this projection process, V-PCC is suitable for encoding high-density point clouds, such as people, for which subjective image quality is important. On the other hand, G-PCC is a method of encoding 3D point clouds directly as 3D data, so it is suitable for encoding point clouds of large-scale 3D scenes, such as buildings and cities, for which highly accurate position coordinates are required. The authors have contributed to the establishment of standard technologies through numerous technical proposals in the international standardization of V-PCC and G-PCC. In particular, for G-PCC, we have led the standard development as the editor of the standard technology specifications to this day.

Practical use of the MPEG standard requires the development of standard-compliant encoders and decoders (these are collectively called codecs). Decoders must be capable of real-time decoding, especially when handling point cloud data of dynamic objects. For example, when decoding and viewing 30-frame-per-second (fps) content, if the decoder's throughput is not 30 fps or higher, playback will stop and the viewing experience will be unsmooth. Furthermore, the MPEG standard only specifies the decoding process, leaving each codec developer flexibility in the design of the encoder's coding parameter determination method. Therefore, for practical use of point clouds, it is important to develop an encoder that achieves a high compression rate and sufficient processing speed while maintaining point cloud playback quality. Previously, there were real-time systems that could only encode (5) or only decode (6) V-PCC. However, there was no end-to-end real-time codec system that supported both real-time encoding and decoding, and had the functionality for real-time input/output of point cloud data and the functionality for sending and receiving compressed data over a network. Furthermore, Google developed Draco7), a technology that can encode and decode point cloud data at high speed, but V-PCC and G-PCC achieve higher compression performance for point cloud data than Draco8).

In response to this, the authors have been working on the following project to quickly verify the applicability and usefulness of the international standard method for point cloud coding in practical use. We have developed the world's first end-to-end real-time codec system compatible with V-PCC and G-PCC9)-12). To enable early verification, both were developed on a software basis. Furthermore, through the world's first real-time point cloud compression and transmission experiment on a commercial network using these codecs, We have demonstrated the potential for development of both V-PCC and G-PCC. These efforts were recognized and received the ITE Technology Promotion Award (Advancement and Development Award). This paper introduces the V-PCC and G-PCC real-time codecs we developed, as well as the demonstration experiments using them.

2. V-PCC real-time codec

2.1 Overview of V-PCC

As its name suggests, V-PCC is a point cloud encoding method that utilizes video encoding techniques. Figure 2 shows an overview of the V-PCC encoding process. When encoding a point cloud, a polyhedron is first assumed to contain the point cloud to be encoded for each frame. Next, the point cloud is divided into several small regions called patches, and each patch is determined to be projected onto a surface of the polyhedron. Conventional methods (reference software) perform segmentation using normal vector information calculated for each point in the input point cloud, and then generate the final patch by correcting adjacent points so that they belong to the same patch as much as possible. Four types of data are then generated, as shown in Figure 2. The generated patches are then tiled over a two-dimensional image to create an image. The reference software is software released by MPEG as an example of an encoder and decoder compatible with each standard. The encoder incorporates parameter search methods proposed by participating companies during the standardization process. Patch information is metadata that indicates where each patch is located within the image. The patch information is not converted into image data, but is encoded as metadata as is. The mask image is mask information that indicates at which pixel in the image the data for the projected patch exists when each patch is projected onto a 2D

image. The depth image is image information converted from the depth information of each point corresponding to each pixel when viewed from the projection surface of each patch. The texture image is the color information of each point. The data converted into images for each frame in this way is then encoded as video data using video coding technologies such as H.265 | HEVC or H.266 | VVC.

On the decoder side, the 3D shape of the point cloud is first reconstructed using the patch information, mask information decoded as an image, and depth information, as shown in Figure 2. Then, the texture is projected to obtain the final decoded point cloud13)-16).

2.2 Development of V-PCC real-time codec The authors have developed the world's first real-time V-PCC codec9). Here, real-time is defined as the throughput (steady output rate excluding initial delay) of the encoder and decoder being equal to or greater than the input rate of the input data. In order to make V-PCC encoding and decoding processes real-time,

Based on the V-PCC reference software, the following two main improvements were implemented9): In addition, by implementing various innovations including the following two, the throughput was increased from approximately 0.006 fps to approximately 30 fps (approximately 5000 times faster) for a point cloud of people with approximately 800,000 points per frame (30 fps).

[Copyrights to Machine-Translated Content]

The copyright of the original papers published on this website belongs to the Institute of Image Information and Television Engineers. Unauthorized use of original papers or translated papers is prohibited. Please be sure to cite the original publication when referencing. For details, please refer to the copyright regulations of the Institute of Image Information and Television Engineers.

タを H.265 | HEVC³⁾ や H.266 | VVC⁴⁾ などの映像符号化技術を用いて符号化する手法である。V-PCC はこの投影処理の特性上、主観画質が重視される人物などの高密度な点群の符号化に適している。一方、G-PCC は三次元点群をそのまま三次元データとして符号化する手法であるため、高精度な位置座標が要請される建造物や都市などの大規模な三次元シーンの点群の符号化に適している。筆者らは V-PCC および G-PCC の国際標準化において、多数の技術提案を通じて標準技術の確立に貢献してきた。特に G-PCC においては、現在に至るまで標準技術の規格書エディタとして規格開発を牽引している。

MPEG の標準方式の実利用には、方式準拠のエンコーダ、およびデコーダ（エンコーダとデコーダをまとめてコーデックとも呼ぶ）を開発する必要がある。デコーダは、特に動的な物体の点群データを扱う場合、リアルタイムにデコードできることが必須である。例えば、30fps (frame per second) のコンテンツをデコードして視聴する場合、デコーダのスループットが 30fps 以上になっていないと、再生が止まってしまうなど、スムーズに視聴することができない。また、MPEG の標準規格で規定されているのはデコード処理のみであり、エンコーダ側での符号化パラメータの決定手法等は各コーデック開発ベンダに設計の余地がある。よって、点群の実利用には、点群の再生品質を保ちつつ、高い圧縮率を実現し、かつ、充分な処理速度を実現するエンコーダの開発が重要である。従来は、例えば V-PCC においてエンコードのみ⁵⁾、あるいはデコードのみ⁶⁾ に対応したリアルタイムシステムは存在していた。しかし、リアルタイムエンコードおよびデコードの両方に対応し、さらに点群データのリアルタイム入出力機能と圧縮したデータのネットワーク送受信機能を備えた、いわゆる End-to-End のリアルタイムコーデックシステムは存在していなかった。また、点群データを高速にエンコードおよびデコード可能な技術として Google が開発した Draco⁷⁾ が存在するが、V-PCC や G-PCC は点群データにおいて Draco より高い圧縮性能を実現している⁸⁾。

これに対し筆者らは、点群符号化の国際標準方式の実利用への適用可能性および有用性を早期に検証するため、V-PCC および G-PCC に対応した End-to-End のリアルタイムコーデックシステムを世界で初めて開発した^{9)~12)}。早期検証を可能とするため、いずれもソフトウェアベースで開発した。また、これらのコーデックを用いた世界初の商用網路上での点群リアルタイム圧縮伝送実験により、V-PCC と G-PCC の両方式の発展性を実証した。これらの取り組みが評価され、ITE 技術振興賞（進歩開発賞）を受賞した。本稿では、開発した V-PCC、G-PCC リアルタイムコーデックと、それらを用いた実証実験についてそれぞれ紹介する。

2. V-PCC リアルタイムコーデック

2.1 V-PCC の概要

V-PCC は、その名の通り Video（映像）符号化手法を活用した点群の符号化方式である。図 2 に V-PCC のエンコード処理の概要を示す。点群をエンコードする際には、まずフレームごとに符号化対象の点群を包含する多面体を仮定する。次に、点群をいくつかのパッチと呼ばれる小領域に分割し、パッチごとに前記の多面体のどの面に投影するかを決定する。ここで、従来手法（参照ソフトウェア）では、入力点群の点ごとに算出した法線ベクトルの情報を用いてセグメンテーションを行い、その後、隣接する点同士がなるべく同じパッチに属するように補正することで最終的なパッチを生成している。その後、図 2 に示すような 4 種類のデータを生成する。この時、生成したパッチを二次元画像上に敷き詰めていくことで画像化する。なお、参照ソフトウェアとは、各規格に対応するエンコーダおよびデコーダの実装例として MPEG が公開しているソフトウェアであり、エンコーダには標準化を通じて参加社から提案されたパラメータ探索手法等が実装されている。

パッチ情報は、各パッチが画像内のどこに配置されているか、などを示すメタデータである。パッチ情報は画像データに変換せず、そのままメタデータとして符号化する。マスク画像は、各パッチを二次元画像上に投影した際に、画像内のどの画素に投影したパッチのデータが存在するかを示すマスク情報である。深度画像は、各パッチの投影面から見た時の、各画素に対応する各点の深度情報を画像情報に変換したものである。テクスチャ画像は各点の色情報である。このようにフレームごとに画像に変換したデータを、映像データとして H.265 | HEVC や H.266 | VVC といった映像符号化技術を用いて符号化する。

デコーダ側では、図 2 に示すようにパッチ情報と画像として復号されたマスク情報、深度情報を用いてまず点群の三次元形状を復元する。そのあと、テクスチャを投影することで最終的な復号点群を得る^{13)~16)}。

2.2 V-PCC リアルタイムコーデックの開発

筆者らは、世界で初めて V-PCC のリアルタイムコーデックを開発した⁹⁾。なお、ここでリアルタイムとはエンコーダ、デコーダのスループット（初期遅延を除いた定常的な出力レート）がそれぞれ入力データの入力レート以上であることと定義する。V-PCC の符号化、復号処理をリアルタイム化するために、MPEG で開発された V-PCC 参照ソフトウェアをベースとして、主に以下二点の工夫を実施した⁹⁾。また、以下 2 点を含む様々な工夫によって、約 80 万点/フレームの人物の点群 (30fps) に対して、スループットを約 0.006fps から約 30fps（約 5000 倍）に高速化した。

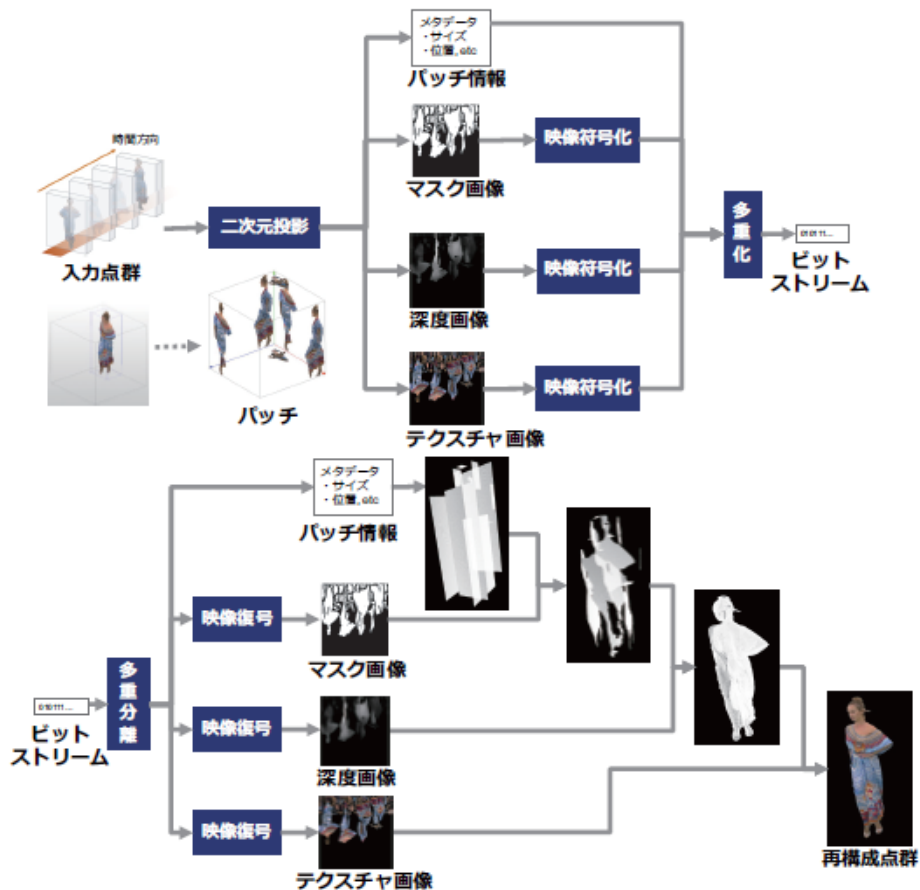


図 2 V-PCC の符号化・復号処理フローの概要
(V-PCC Codec description に基づき作成)

(1) Accelerating patch generation

Our proposed method achieves approximately 20 times faster processing speed compared to conventional methods by improving the patch generation process. A preliminary investigation of the processing time breakdown in the V-PCC reference software revealed that patch generation accounted for 80–90 % of the total processing time. Therefore, the proposed method aims to speed up overall processing by accelerating the patch generation process. As explained in Section 2.1, conventional methods first calculate normal vectors for all points, and then generate patches by segmenting using the normal vectors. In contrast, our proposed method pre-divides the 3D space to be encoded into small spaces smaller than the final patch. We then introduce a method to determine planes for each small space containing a large number of points. Furthermore, for texture and depth images, the encoder has some flexibility in determining what to do with pixel values in regions other than patches in each image (regions where the corresponding mask image shows no patch data). Compression performance and processing time depend on how pixel values in these regions are generated. In the proposed method, leveraging our expertise in video coding, we reduced processing time without compromising compression performance by maintaining temporal correlation for texture and depth images using simpler processing than conventional approaches¹⁷⁾.

[Copyrights to Machine-Translated Content]

The copyright of the original papers published on this website belongs to the Institute of Image Information and Television Engineers. Unauthorized use of original papers or translated papers is prohibited. Please be sure to cite the original publication when referencing. For details, please refer to the copyright regulations of the Institute of Image Information and Television Engineers.

(2) Improved parallel

processing efficiency Throughput has been improved by further evolving the development know-how gained from past real-time codecs for video encoding to achieve efficient parallel processing. Parallel processing that maximizes CPU performance is essential for accelerating processing using software. We previously developed real-time codecs for H.265 | HEVC and H.266 | VVC . 18) In these codecs, image frames were divided into blocks (CTBs), and processing was parallelized across multiple frames to maximize CPU performance. However, as explained in Section 2.1, there are four types of data in V-PCC, and the processing load for each type of data corresponding to a single block differs. Therefore, the efficiency of conventional parallel processing on a block-by-block basis was insufficient. Furthermore, on the V-PCC reference frame on which the development was based, the encoder locally decoded and smoothed depth images were used when generating texture images, so there was a dependency between the encoding process order of the depth image and the texture image, and the configuration was such that parallel processing across frame types could not be realized. Therefore, as shown in Figure 3, we input the unencoded depth image rather than the locally decoded depth image for texture image generation

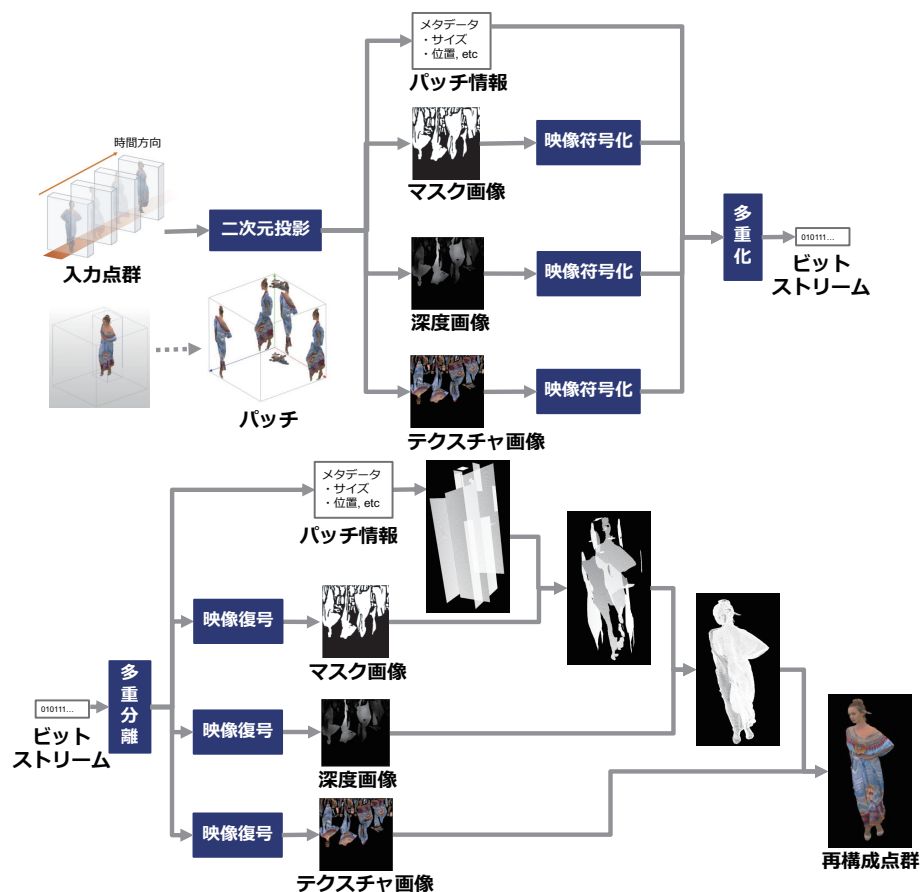


図 2 V-PCC の符号化・復号処理フローの概要
(V-PCC Codec description に基づき作成)

(1) パッチ生成処理の高速化

提案手法では、パッチの生成処理を工夫することで従来方式と比較して約 20 倍の高速化を実現した。事前に V-PCC 参照ソフトウェアにおける処理時間の内訳を調査した結果、パッチ生成処理が全体の 80~90 % を占めていることが分かった。よって、提案手法ではパッチ生成処理を高速化することで全体の処理の高速化を図った。従来手法では、2.1 で説明した通りまずすべての点に対して法線ベクトルを算出し、次に法線ベクトルを用いてセグメンテーションを行うことでパッチを生成していた。これに対して、提案手法ではあらかじめ符号化対象の三次元空間を最終的に生成されるパッチより小さな小空間に分割しておき、多数の点が含まれる小空間のみ、小空間ごとに平面を判定する手法を導入した。さらに、テクスチャ画像と深度画像については各画像内のパッチ以外の領域（対応するマスク画像においてパッチデータが無いことが示されている領域）の画素値をどうするかについてはエンコーダ側に自由度があり、この領域の画素値をどのように生成するかによって圧縮性能と処理時間が変化する。提案手法では、筆者らの映像符号化での知見を活かして、テクスチャ画像と深度画像についてそれぞれ従来より簡易な処理で時間方向の相関を維持することで、圧縮性能を損なうことなく処理にかかる時間を削減

した¹⁷⁾。

(2) 並列処理の効率化

過去の映像符号化に関するリアルタイムコーデックでの開発ノウハウをさらに進化させて効率的な並列処理を実現することでスループットを向上させた。ソフトウェアによる処理高速化においては、CPU の性能を最大限に活用する並列処理が不可欠である。筆者らは、過去に H.265 | HEVC や H.266 | VVC のリアルタイムコーデックを開発しており¹⁸⁾、その際は画像フレームをブロック分割したブロック (CTB) 単位での処理を、複数のフレームにまたがって並列化することで CPU の演算性能を最大限活用していた。しかし、V-PCC において 2.1 で説明した通り 4 種類のデータが存在し、さらに一つのブロックに対応するデータの処理負荷が種類ごとに異なるため、従来のブロック単位での並列処理だけでは効率化が不十分であった。また、開発のベースとした V-PCC 参照フレーム上ではテクスチャ画像生成時にエンコーダ内で局所復号し、かつスムージングした深度画像を用いているため、深度画像とテクスチャ画像の符号化処理順序には依存関係があり、フレーム種別間をまたいだ並列処理が実現できない構成であった。そこで、筆者らは図 3 に示すように、テクスチャ画像生成のために局所復号された深度画像ではなく符号化前の深度画像を入力することで

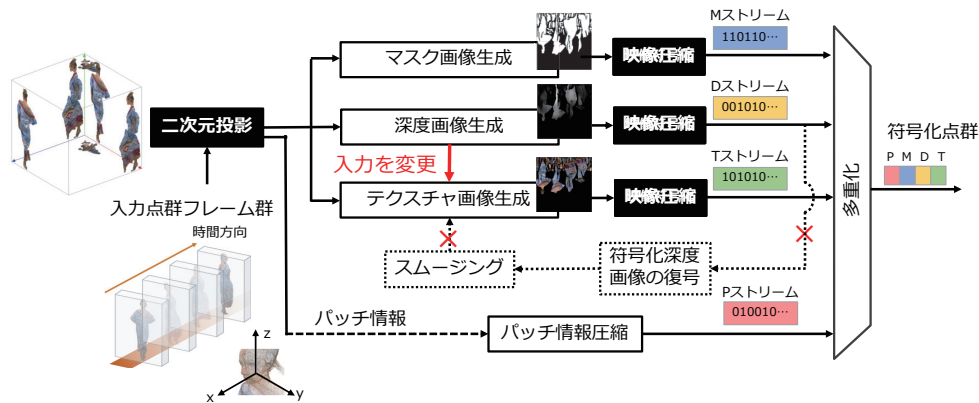


Figure 3: Conceptual diagram of the processing flow of the developed V-PCC real-time encoder

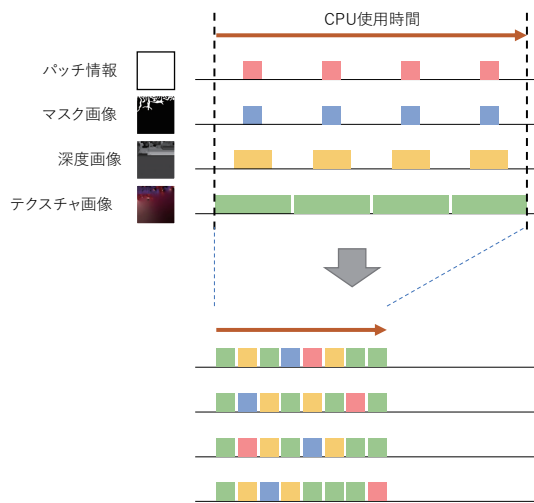


Figure 4: Conceptual diagram of task scheduling across different image frames in the developed V-PCC real-time encoder

We eliminated dependencies and introduced parallel processing across different frame types. In other words, we made it possible to encode texture images without waiting for the encoding and local decoding of depth images. Here, the generation of the texture image is a process that is handled solely by the encoder. On the decoder side, the texture image can simply be decoded from the encoded bitstream, regardless of how it was generated. Therefore, this innovation only affects the encoder side. Furthermore, to compensate for the smoothing effect that suppresses quantization distortion of geometric information and, in turn, contour distortion of human point clouds, we increased the bit depth of the geometric information of the input point cloud. We also introduced task scheduling that takes into account the processing volume of coded frames and blocks across different frame types (Figure 4). These innovations achieved an additional 20 times speedup.

Demonstration

Experiment We

introduce the demonstration experiment conducted in January 2023 using the V-PCC real-time codec.¹⁰ Figure 5 shows the configuration diagram of the demonstration experiment. In the demonstration experiment, as shown in Figure 5

[Copyrights to Machine-Translated Content]

The copyright of the original papers published on this website belongs to the Institute of Image Information and Television Engineers. Unauthorized use of original papers or translated papers is prohibited. Please be sure to cite the original publication when referencing.

For details, please refer to the copyright regulations of the Institute of Image Information and Television Engineers.

In a volumetric studio capable of capturing high-density 3D point clouds of people (approximately 20 million points/second, approximately 1Gbps), captured in advance or in real time, the data was compressed to 25Mbps using the developed V-PCC real-time encoder and transmitted to a remote location via the Internet and optical fiber or 5G networks. At the remote location, the encoded point cloud was received and played back using a V-PCC decoder/viewer implemented on a PC or smartphone, and we confirmed that it could be stably displayed on a holographic stage or smartphone screen. In this way, by enabling the end-to-end acquisition, transmission, and playback of point clouds of people in real time, it is expected that new event experiences will be created, such as for music or fashion show events, where footage shot in a volumetric studio can be directly incorporated into the Metaverse.

3. G-PCC real-time codec

3.1 Overview of G-PCC

G-PCC is a method for processing and encoding 3D point cloud data as it is, without converting it into video or other formats. Figure 6 shows the configuration of G-PCC. G-PCC has multiple tools in the Geometry encoder/decoder and Attribute encoder/decoder, and it is possible to select the appropriate tool on the encoder side depending on the characteristics of the target point cloud¹³⁾¹⁶⁾.

On the encoder side, the geometry of the point cloud is first encoded. The data encoded by Octree or Predictive Geometry is transmitted to the decoder side as a Geometry bitstream.

The Attribute encoder first performs recoloring using the Attribute of the input point cloud and the reconstructed Geometry as input. The recoloring process is performed when the Geometry encoding is lossy, i.e., when the number of points in the reconstructed Geometry is reduced or the positions of the points are shifted compared to the input point cloud.

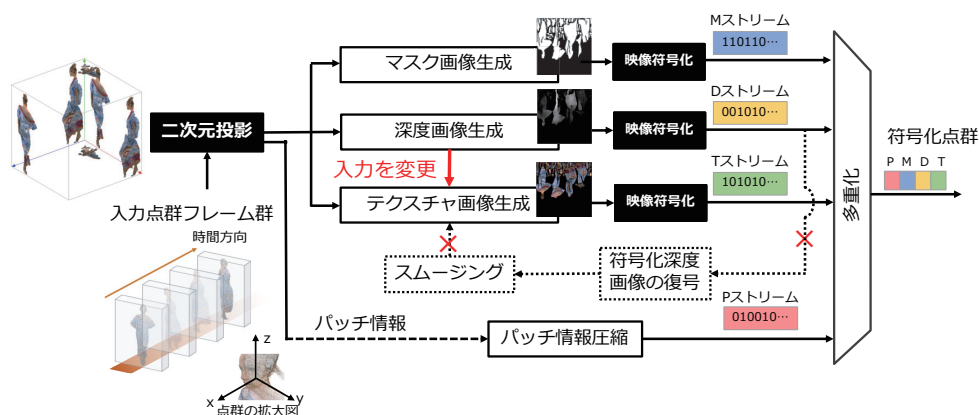


図 3 開発した V-PCC リアルタイムエンコーダの処理フローの概念図

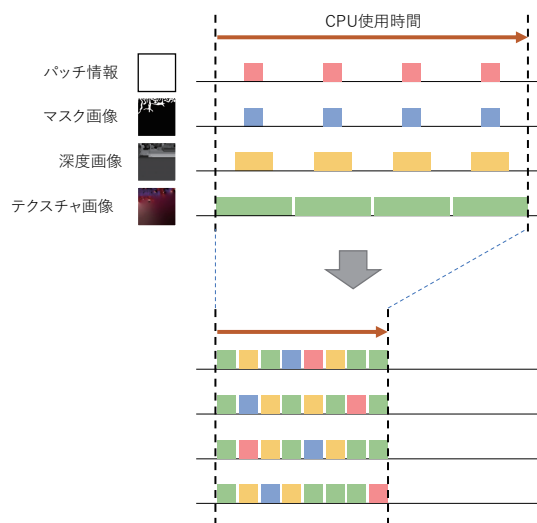


図 4 開発した V-PCC リアルタイムエンコーダの異なる画像フレーム間を跨いだタスクスケジューリングの概念図

依存関係を排除し、異なるフレーム種別間にまたがる並列処理を導入した。言い換えると、深度画像のエンコードおよび局所復号処理を待つことなく、テクスチャ画像のエンコード処理を実行可能とした。ここで、テクスチャ画像をどのように生成するかはエンコーダ側のみの処理であり、デコード側ではどのようにテクスチャ画像が生成されたかに関わらず、符号化伝送されてきたビットストリームからテクスチャ画像をデコードすればよい。よって、この工夫はエンコーダ側のみに影響するものである。さらに、幾何情報の量子化歪みでは人物点群の輪郭歪を抑制するスムージングの効果を補填するため、入力点群の幾何情報を高ビット深度化した。そして、異なるフレーム種別間にまたがる形で符号化フレーム・ブロックの処理量を考慮したタスクスケジューリングを導入した(図 4)。これらの工夫によって、さらに約 20 倍の高速化を実現した。

2.3 実証実験

V-PCC リアルタイムコーデックを用いて 2023 年 1 月に実施した実証実験を紹介する¹⁰⁾。図 5 は実証実験の構成図である。実証実験では、図 5 に示すように、点群

が取得可能なボリュメトリックスタジオで、事前あるいはリアルタイムに取得した人物の高密度三次元点群(約 2000 万点/秒、約 1Gbps)を、開発した V-PCC リアルタイムエンコーダで 25Mbps に圧縮し、Internet および光または 5G 網を経由して遠隔の別拠点へ伝送した。別拠点では、PC またはスマートフォン上に実装した V-PCC デコーダ・ビューワを用いて符号化点群を受信再生してホログラフィックステージまたはスマートフォン画面で安定的に表示できることを確認した。このように、人物の点群を End-to-End でリアルタイムに取得、伝送、および再生表示を可能にすることによって、例えば音楽やファッションなどのショーイベントを対象に、ボリュメトリックスタジオで撮影した映像をそのままメタバースに参加させるといった新しいイベント体験の創出が期待される。

3. G-PCC リアルタイムコーデック

3.1 G-PCC の概要

G-PCC は、三次元点群データを映像などに変換することなく三次元点群のまま処理して符号化する手法である。図 6 に G-PCC の構成を示す。G-PCC はそれぞれ Geometry エンコーダ/デコーダと Attribute エンコーダ/デコーダに複数のツールを備えており、対象の点群の特性に応じてエンコーダ側で適切なツールを選択することが可能である¹³⁾¹⁶⁾。

エンコーダ側では、まず点群の Geometry の符号化を行う。Octree または Predictive Geometry によって符号化したデータは Geometry ビットストリームとしてデコーダ側に伝送される。また、エンコード側で局所復号した Geometry を、再構成 Geometry として Attribute エンコーダへ出力する。Attribute エンコーダでは、まず入力点群の Attribute と再構成 Geometry を入力として Recolor 処理を行う。Recolor 処理は、Geometry 符号化がロッキー符号化であった場合、すなわち、入力点群に対して再構成 Geometry における点の数が減少していたり、点の位置がずれたりしている場合に、再構成 Geometry

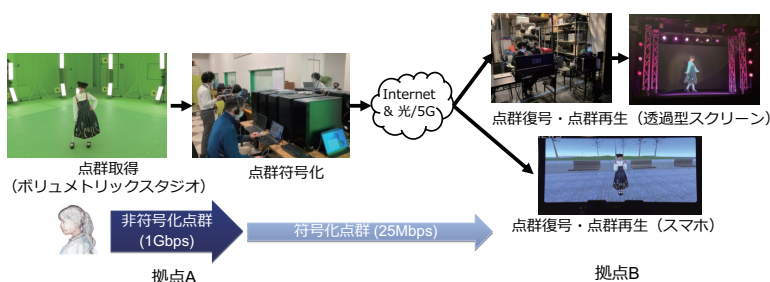


Figure 5. V-PCC transmission experiment configuration.

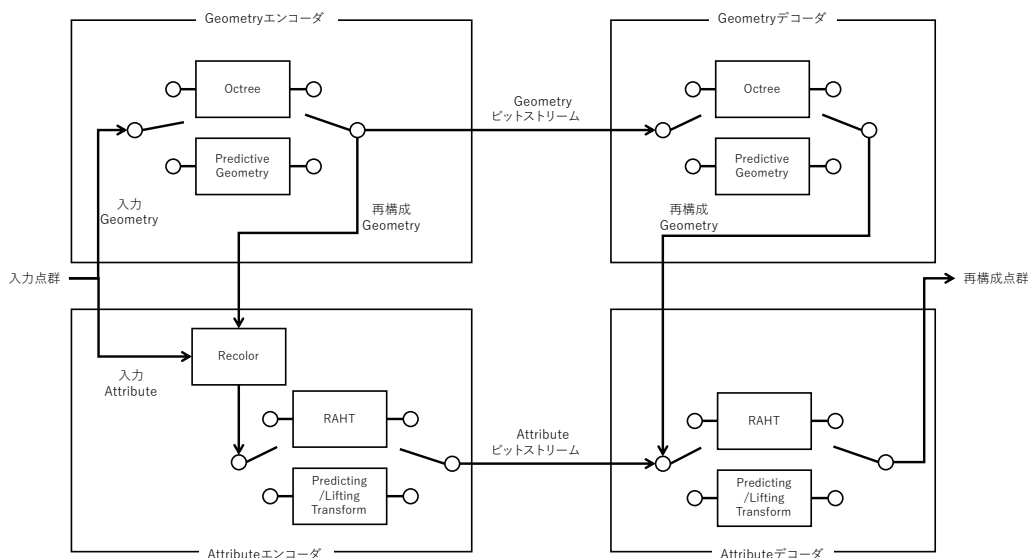


Figure 6. G-PCC encoding and decoding process configuration

The recolor process recalculates attributes according to the coordinates of each point in the reconstructed Geometry. Specifically, the attribute values of each point in the reconstructed Geometry are calculated by filtering using the attribute values of the input point cloud that exist in the vicinity of each point in the reconstructed Geometry. Note that if the Geometry is encoded using lossless encoding, the recolor process is unnecessary and is therefore omitted. The point cloud with attribute values for each point in the reconstructed Geometry calculated by the recolor process is used as input, and the attributes are encoded using RAHT or Predicting/Lifting Transform and transmitted to the decoder as an attribute bitstream. The decoder first generates a reconstructed Geometry from the Geometry

bitstream. Next, the attributes for each point in the reconstructed Geometry are generated from the attribute bitstream to obtain the final reconstructed point cloud.

Of the multiple coding tools defined in G-PCC, we will provide an overview of Octree and RAHT, which are primarily used in the G-PCC real-time codec described below.

Octree is a method that assumes a rectangular parallelepiped that contains the point cloud to be processed and recursively divides that space. It is also used for 3D point cloud processing other than encoding. When used for encoding, the rectangular parallelepiped (node) is divided into two in the x, y, and z directions, resulting in eight regions (child nodes), each of which is divided into eight regions.

An occupancy code is calculated that indicates whether or not the region contains even one point (occupied/unoccupied). Unoccupied regions are no longer divided, and occupied regions are divided again. This process is carried out recursively until the size of the divided regions is $1 \times 1 \times 1$. This allows us to determine with integer coordinate precision ($1 \times 1 \times 1$ voxel precision) whether or not there is a point at each coordinate position.

The decoder side can encode the occupancy data transmitted from the encoder side using the occupancy code.

By reproducing the division on the encoder side based on the code, it is possible to restore the positions where points exist.

RAHT20) is a method for efficiently encoding attribute information such as color and reflection coefficients by recursively converting them into low- and high-frequency components using a transform similar to the Haar-Wavelet transform. In the case of images, the number of pixels in adjacent regions is always the same, so the weights used during transformation are equal. However, in the case of point clouds, the number of points contained in adjacent regions is not necessarily the same. Therefore, a technique is devised to weight the transformation based on the number of points contained in each region when performing the transformation. In addition to simply encoding transform coefficients, it is also possible to use intra-prediction, which performs spatial prediction based on transform coefficients in higher layers that have already been decoded, and decodes only the prediction residual.

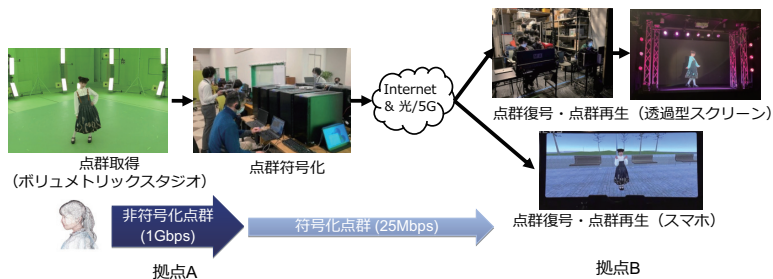


図 5 V-PCC 伝送実験構成

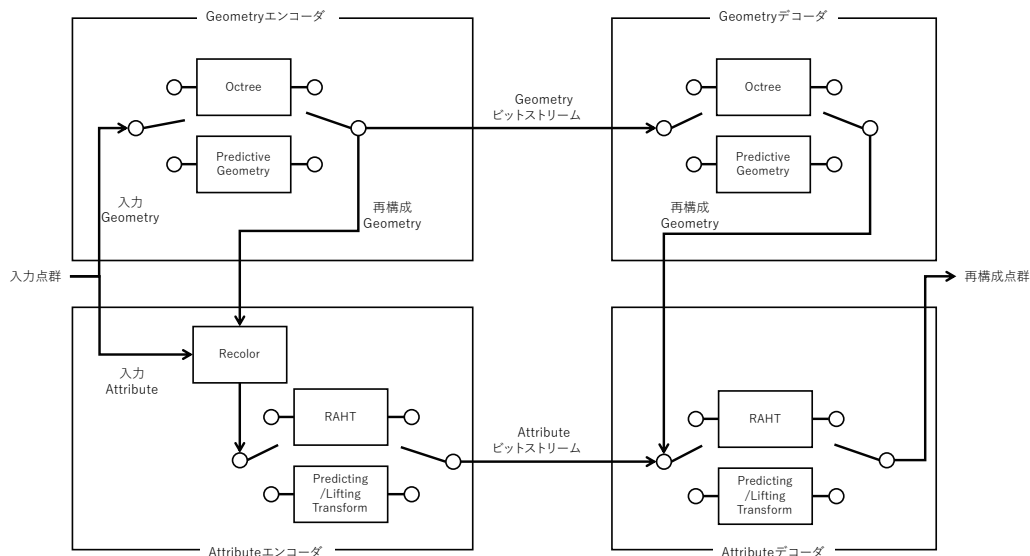


図 6 G-PCC のエンコードおよびデコード処理構成図

の各点の座標に応じて Attribute を再計算する処理である。具体的には、再構成 Geometry の各点の近傍に存在する入力点群の Attribute の値を用いてフィルタリングをすることで、再構成 Geometry の各点の Attribute 値を算出する。なお、Geometry の符号化がロスレス符号化であった場合は、Recolor 処理は不要であるため省略される。Recolor 処理によって算出された、再構成 Geometry の各点に Attribute 値を持つ点群を入力として、RAHT または Predicting/Lifting Transform によって Attribute を符号化し、Attribute ビットストリームとしてデコーダ側に伝送する。

デコーダ側では、まず Geometry ビットストリームから再構成 Geometry を生成する。次に再構成 Geometry の各点の Attribute を Attribute ビットストリームから生成することで、最終的な再構成点群を得る。

G-PCC に定義されてる複数の符号化ツールのうち、例として後述する G-PCC リアルタイムコーデックで主に使用している Octree と、RAHT について概要を紹介する。

Octree は、処理対象の点群を包含する直方体を仮定し、その空間を再帰的に分割する手法であり、符号化以外の三次元点群処理でも活用されている。符号化に用いる場合は、前記の直方体（ノード）を x , y , z 方向にそれぞれ 2 分割してできる 8 つの領域（子ノード）ごとに、各

領域内に 1 点でも点が含まれているかどうか（占有/非占有）を示す Occupancy Code を算出する。非占有の領域は分割を終了し、占有の領域は再度分割を行う。これらの処理を分割後の領域のサイズが $1 \times 1 \times 1$ になるまで再帰的に行う。これにより、整数座標精度で（ $1 \times 1 \times 1$ のボクセル精度で）、各座標位置に点があるか無いかを Occupancy Code によって符号化することができる。デコーダ側では、エンコーダ側から伝送された Occupancy Code に基づいてエンコーダ側での分割を再現することで、どの位置に点が存在するかを復元できる。

RAHT²⁰⁾ は、Haar-Wavelet 変換と同様の変換を行い色や反射係数などの属性情報を低周波成分と高周波成分に再帰的に変換することで効率的に符号化する手法である。画像の場合は隣り合う領域内に存在する画素数は常に等しいため変換時の重みは等しいが、点群の場合は隣り合う領域内に含まれる点の数が必ずしも同一であるとは限らない。よって、変換を行う際の領域ごとに含まれる点数によって重みづけをして変換する工夫が行われている。また、単純に変換係数を符号化するだけでなく、すでに復号が完了した上位階層の変換係数に基づいて空間方向の予測を行い、予測残差のみを復号するイントラ予測を用いることが可能である。

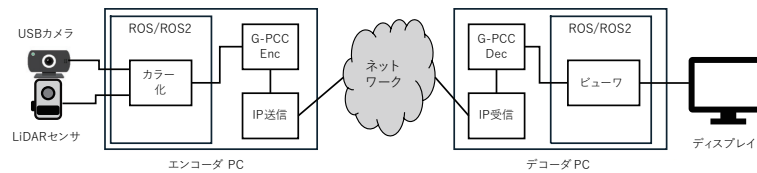


Figure 7 G-PCC real-time codec configuration

3.2 Development of G-PCC real-time codec

The authors have developed the world's first end-to-end G-PCC real-time codec¹¹⁾. Here, real-time is defined as the throughput of the encoder and decoder being equal to or greater than the input rate of the input data. Figure 7 shows the system configuration of the G-PCC real-time codec. The encoder PC receives the point cloud acquired by the LiDAR sensor and the RGB image acquired by the USB camera. Within the encoder PC, the point cloud and RGB image data are processed by ROS (Robot

The point cloud acquired by LiDAR usually does not have color information, so color is added to each point (colorization) based on the color information acquired by the USB camera. The colorized point cloud data is then output as a ROS topic and input to the G-PCC encoder. After the encoder encodes the geometry (coordinate information) and attribute (color information), the data is transmitted to the G-PCC decoder via the IP transmission/reception module and the network. The G-PCC decoder outputs the decoded point cloud as a ROS topic, and the decoded point cloud is displayed using a viewer provided with ROS such as rviz. Here, the encoder and decoder are

Based on the G-PCC reference software developed by MPEG, we newly implemented frame-by-frame parallel processing to speed up processing. Here, the term frame is defined the same as for video data. That is, when point clouds are acquired continuously in time, a frame is defined as a point cloud acquired at the same time (or considered to have been acquired at the same time). It is also possible to divide a frame into multiple regions and consider each divided point cloud as a pseudo-frame. With this ingenuity, we confirmed that the throughput was approximately six times faster (up to approximately 2.4 million points/second) by processing eight parallel threads, compared to a maximum of approximately 400,000 points / second in sequential processing (single-threaded). Note that throughput varies slightly depending on the input point cloud, so the above figures are for the point cloud data (MPEG test data) used for speed measurement. As mentioned above, ROS is used for the input and output of point cloud data. Although ROS is not essential for real-time processing itself, many LiDAR sensors are compatible with ROS, so ROS was adopted from the perspective of interoperability. Although ROS was used at the beginning of development, it is now being used as shown in the figure.

It also supports ROS2. In the initial development, both the encoder and decoder were implemented on laptops.

Currently, the encoder can run on a small computer such as the NVIDIA Jetson Orin Nano, and in the demonstration experiment described below, a Jetson Orin Nano was installed on a drone and a dog-shaped robot to transmit point cloud data acquired in real time. In addition, an iPhone app equipped with a real-time encoder function has been developed, which makes it possible to transmit point cloud data acquired by the iPhone's LiDAR sensor in real time.

3.3 Demonstration experiment

This section introduces a demonstration experiment using the G-PCC real-time

codec. First, in January 2023, we demonstrated that a pre-generated point cloud of a building could be time-divided at approximately 300,000 points per second, compressed using a real-time encoder, and transmitted stably via 5G /LTE. 10) Specifically, the input point cloud was divided into approximately 30,000 points per frame and input to the encoder at 10 fps. In this experiment, stable transmission of the point cloud was not possible when processed using a single thread. However, we confirmed that stable transmission was possible when parallel processing was performed using two threads. We also confirmed that the real-time codec could compress the point cloud file size from approximately 26 Mbps to approximately 1.2 Mbps (approximately 1/21 of the size).

Furthermore, a demonstration experiment was conducted in August 2024 to transmit point cloud data acquired in real time from the construction site to a remote location.²¹⁾ Li-

We equipped a quadruped robot and a drone with a small computer (Jetson Orin Nano) equipped with a DAR sensor, a USB camera for coloring point clouds, a communications module, and a real-time encoder, and confirmed that real-time point cloud acquisition and transmission to a remote location (Shimizu Corporation's Onko Soushin no Mori NOVARE, Koto-ku, Tokyo) was possible (Figure 8). This technology significantly reduces the time required to acquire point clouds at construction sites, etc., from remotely capturing the point cloud to confirming the data, which previously took several hours, to less than 10 seconds. In the future, remotely controlling, semi-automating, or autonomous quadruped robots and drones will enable remote construction management, monitoring, and inspection, which is expected to improve the efficiency of quality and progress management as well as safety and productivity.

4. Conclusion

This paper outlines the development of the world's first real-time point cloud compression and transmission system that complies with the MPEG point cloud coding standard (PCC).

[Copyrights to Machine-Translated Content]

The copyright of the original papers published on this website belongs to the Institute of Image Information and Television Engineers. Unauthorized use of original papers or translated papers is prohibited. Please be sure to cite the original publication when referencing. For details, please refer to the copyright regulations of the Institute of Image Information and Television Engineers.

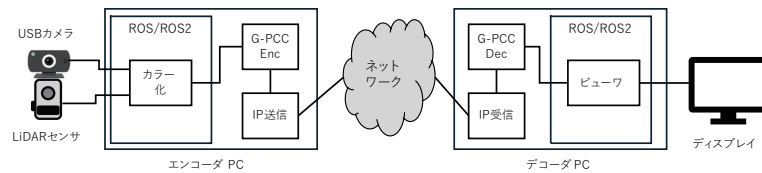


図 7 G-PCC リアルタイムコーデック構成

3.2 G-PCC リアルタイムコーデックの開発

筆者らは、世界で初めて End-to-End の G-PCC リアルタイムコーデックを開発した¹¹⁾。ここでも、リアルタイムとはエンコーダ、デコーダのスループットが入力データの入力レート以上であることと定義する。G-PCC リアルタイムコーデックのシステム構成を図 7 に示す。エンコーダ PC には、LiDAR センサで取得した点群と USB カメラで取得した RGB 画像を入力する。エンコーダ PC 内では、前記の点群と RGB 画像データを ROS (Robot Operating System) のトピックとして扱う。LiDAR で取得した点群には通常色情報が無いので、USB カメラで取得した色情報に基づいて各点に色を付加 (カラー化) する。その後、カラー化後の点群データも ROS のトピックとして出力し、G-PCC エンコーダに入力する。エンコーダにおいて Geometry (座標情報) と Attribute (色情報) を符号化した後、IP 送受信モジュールとネットワークを経由して G-PCC デコーダにデータを伝送する。G-PCC デコーダは、復号した点群を ROS のトピックとして出力し、rviz などの ROS に備えられたビューワを用いて復号した点群を表示する。ここで、エンコーダとデコーダは MPEG が開発した G-PCC 参照ソフトウェアをベースとして、処理を高速化するために、フレームごとの並列処理を新規に実装した。ここでフレームとは、映像データの場合と同一の定義である。すなわち、時間的に連続して点群を取得する場合、同一時刻に取得した (あるいは同一時刻に取得したとみなす) 点群をフレームと定義する。なお、一つのフレームを複数の領域に分割し、分割後の点群それぞれを擬似的なフレームとみなすことも可能である。この工夫により、逐次処理の場合 (シングルスレッドの場合) のスループットが最大で 40 万点/秒程度であったのに対し、8 並列で処理することで約 6 倍の速度 (最大 240 万点/秒程度) を実現することを確認した。なお、スループットは入力する点群に応じて多少変動するため、前述の数値は速度測定に使用した点群データ (MPEG のテストデータ) における数値である。また、上記の通り、点群データの入出力部分等には ROS を用いている。リアルタイム処理そのものに ROS は必須ではないが、多くの LiDAR センサが ROS に対応しているため、相互接続性の観点で ROS を採用した。なお、開発当初は ROS を用いていたが、現在では図中にも示す通り ROS2 にも対応している。また、開発当初はエンコーダ側、デコーダ側ともにノート PC で実現していたが、現

在ではエンコーダ側は NVIDIA Jetson Orin Nano などの小型コンピュータでの動作も可能であり、後述する実証実験では Jetson Orin Nano をドローンや犬型ロボットに搭載してリアルタイムに取得した点群の伝送実験を行った。さらに、リアルタイムエンコーダ機能を搭載した iPhone アプリも開発しており、iPhone の LiDAR センサで取得した点群データをリアルタイムに伝送することも可能である。

3.3 実証実験

G-PCC リアルタイムコーデックを用いた実証実験を紹介する。

まず、事前に生成した建造物の点群を約 30 万点/秒に時分割したうえでリアルタイムエンコーダを用いて圧縮し、5G/LTE を経由して安定的に伝送できることを 2023 年 1 月に実証した¹⁰⁾。具体的には、入力点群を約 3 万点/フレームに分割し、10fps でエンコーダに入力した。本実験においては、シングルスレッドで処理をした場合は安定的に点群を伝送することができなかった。一方、2 スレッドで並列処理を行うことで安定的に伝送できることを確認した。また、当該リアルタイムコーデックによって点群のファイルサイズを約 26Mbps から約 1.2Mbps (約 1/21 のサイズ) に圧縮できることを確認した。

さらに、建設現場からリアルタイムに取得した点群を遠隔地に伝送する実証実験を 2024 年 8 月に実施した²¹⁾。建設中の北海道新幹線渡島トンネル坑内外において、LiDAR センサ、点群に色を付けるための USB カメラ、通信モジュール、およびリアルタイムエンコーダを実装した小型コンピュータ (Jetson Orin Nano) を四足歩行ロボットとドローンにそれぞれ搭載し、点群の取得と遠隔地 (清水建設 温故創新の森 NOVARE (東京都江東区)) への伝送をリアルタイムに実現できることを確認した (図 8)。この技術により、従来は建設現場等で点群を取得する場合に遠隔での点群の撮影からデータ確認まで数時間かかっていたが、10 秒以内まで大幅に短縮することが可能となる。将来的には四足歩行ロボットやドローンなどを遠隔操作/半自動化/自律化することで、遠隔からの施工管理や監視・検査を実現し、品質・進捗管理の効率化や安全性・生産性の向上が期待できる。

4. む す び

本稿では、MPEG の点群符号化方式 (PCC) に準拠した世界初のリアルタイム点群圧縮伝送システムの開発概

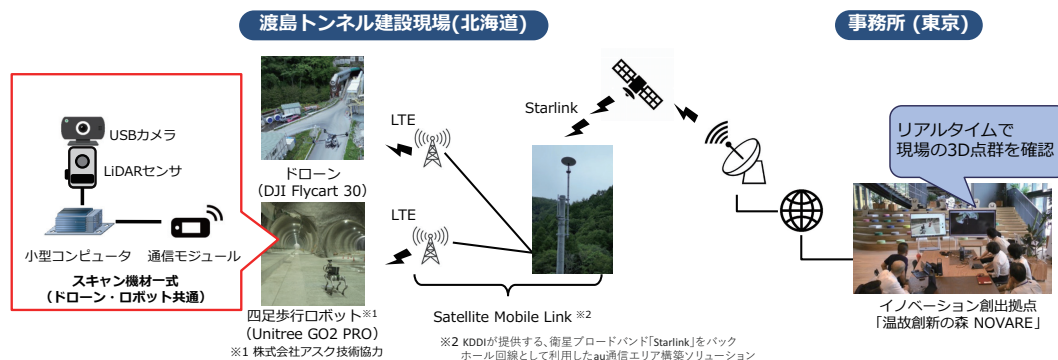


Figure 8. G-PCC transmission experiment configuration

We have introduced the key points of this standard and the demonstration experiments that have been carried out using it. Based on these results, the authors are currently leading the standardization of a successor to PCC in MPEG. We will continue our research and development in the future, aiming to solve social issues by distributing point cloud data in a wide range of fields, from entertainment fields such as the metaverse to construction, civil engineering, and disaster response. Acknowledgments This research and development was supported by the Ministry of Internal Affairs and Communications 'SCOPE (International Standard Acquisition Type) This work was commissioned under the JPJ000595. We would also like to express our gratitude to Shimizu Corporation and ASK Corporation for their cooperation in the real-time point cloud transmission experiment from a construction site (August 2024).

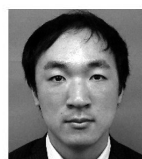
(文 献)

- 1) ISO/IEC 23090-5: "Information technology - Coded representation of immersive media Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)", in Edition 1.0 (2021)
- 2) ISO/IEC 23090-9: "Information technology - Coded representation of immersive media Part 9: Geometry-based point cloud compression", in Edition 1.0 (2023)
- 3) Recommendation ITU-T H.265: "High efficiency coding", in Edition 1.0 (2013)
- 4) Recommendation ITU-T H.266: "Versatile video coding", in Edition 1.0 (2020)
- 5) Y. Kim and Y. -H. Kim: "Real-Time Video-based Point Cloud Encoding System on a Distributed Platform", in proceedings of 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 593-595 (2023)
- 6) InterDigital: "V-PCC: The First MPEG Codec for Point Cloud Compression", <https://www.interdigital.com/videos/v-pcc-the-first-mpeg-codec-for-point-cloud-compression> (2019) (accessed 2024.12.17)
- 7) The Draco authors: "Draco 3D Graphics Compression", <https://google.github.io/draco/> (2017) (accessed 2024.12.17)
- 8) J. Prazeres, M. Pereira and A. M. G. Pinheiro: "Subjective Quality Evaluation of Point Clouds with 3D Stereoscopic Visualization", in proceedings of 2022 IEEE International Conference on Image Processing (ICIP), pp. 2861-2865 (2022)
- 9) KDDI 総合研究所: "点群圧縮技術の最新の国際標準方式に対応したリアルタイムエンコーダーを開発", <https://www.kddi-research.jp/newsrelease/2022/102401.html> (2022) (accessed 2024.12.17)
- 10) KDDI 総合研究所: "点群圧縮技術の最新の国際標準方式に対応したリアルタイムコーデックによる伝送実験に成功", <https://www.kddi-research.jp/newsrelease/2023/012401.html> (2023) (accessed 2024.12.17)
- 11) 辻智弘, 河村圭: "G-PCC 対応 3D 点群リアルタイム伝送システムの試作開発", 電子情報通信学会大会講演論文集 (2023)
- 12) 中塚智尋: "点群圧縮技術の国際標準方式に対応したリアルタイムエンコーダ", 映像情報メディア学会誌, **78**, 2, pp. 10-13 (2024)
- 13) S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P.A. Chou, R.A. Cohen, M. Krivokuc' a, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan,

- A. Tabatabai, A.M. Tourapis and V. Zakharchenko: "Emerging MPEG Standards for Point Cloud Compression", in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, **9**, 1, pp.133-148 (2019)
- 14) C. Cao, M. Preda, V. Zakharchenko, E.S. Jang and T. Zaharia: "Compression of Sparse and Dense Dynamic Point Clouds-Methods and Standards", in Proceedings of the IEEE, **109**, 9, pp. 1537-1558 (2021)
- 15) "V-PCC codec description", ISO/IEC JTC1/SC29/WG7 N0100 (2020)
- 16) 海野恭平: "3次元グラフィックス符号化", 映像情報メディア学会誌, **77**, 5, pp. 43-47 (2024)
- 17) 足立翔平, 木谷佳隆, 渡辺裕, 河村圭: "V-PCC エンコーダにおけるテキスト画像のパッチ間平滑化に関する一検討", 画像符号化シンポジウム/映像メディア処理シンポジウム, P2-09 (2022)
- 18) 木谷佳隆, 海野恭平, 河村圭: "H.266 | VVC 対応 4K/8K リアルタイムコーデックの開発", 映像情報メディア学会誌, **78**, 1, pp. 115-123 (2024)
- 19) "G-PCC codec description", ISO/IEC JTC1/SC29/WG7 N0217 (2022)
- 20) R. L. de Queiroz and P.A. Chou: "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform", in IEEE Transactions on Image Processing, **25**, 8, pp. 3947-3956 (2016)
- 21) KDDI 総合研究所: "Starlink 活用によるトンネル建設現場からの 3D 点群データのリアルタイム伝送に成功", <https://www.kddi-research.jp/newsrelease/2024/090201.html> (2024) (accessed 2024.12.17)



うんの きょうへい
海野 恭平 2009 年, 東京理科大学理工学部電気電子情報工学科卒業。2011 年, 同大学院理工学研究科修士課程了。2021 年, 同大学院理工学研究科博士課程了。現在, (株) KDDI 総合研究所 3D 空間伝送グループコアリサーチ。主に動画画像符号化と点群符号化の研究・開発および国際標準化に従事。博士(工学)。正会員。



きだに よしたか
木谷 佳隆 2010 年, 名古屋大学工学部電気電子情報工学科卒業。2012 年, 同大学院工学研究科修士課程了。2024 年, 早稲田大学大学院基幹理工学研究科博士課程了。2012 年, KDDI(株) 入社。現在, (株) KDDI 総合研究所先端映像通信グループエキスパート。主に, 動画画像符号化方式の研究・開発および国際標準化に従事。博士(工学)。正会員。



つじ ともひろ
辻 智弘 1997 年, 富山県立大学工学部電子情報工学科卒業。1999 年, 同大学院工学研究科修士課程了。(株) KDDI 総合研究所にて動画画像の符号化・編集・伝送に関する研究開発やプロダクト開発に従事後, 現在, KDDI(株) DX・IoT サービス企画部コアスタッフ。主にデジタルツインに関わる DX サービスの企画・提供に従事。正会員。

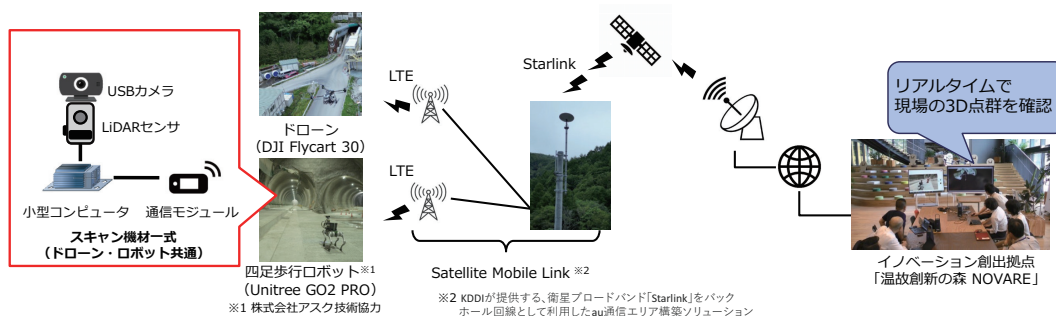


図 8 G-PCC 伝送実験構成

要, およびこれを活用して実施した実証実験について紹介した。筆者らは, これらの成果を基に, 現在は MPEG における PCC の後継方式の標準化を牽引している。今後も研究開発を継続し, メタバースなどのエンターテインメント分野から, 建設, 土木, 災害対応などの幅広い分野で点群データを流通させて, 社会課題の解決を目指していきたい。

謝辞 本研究開発は総務省 SCOPE (国際標準獲得型) JPJ000595 の委託を受けたものである。また, 建設現場からのリアルタイム点群伝送実験 (2024 年 8 月) における清水建設 (株) 様および (株) アスク様のご協力に謝意を表する。

(文 献)

- 1) ISO/IEC 23090-5: "Information technology – Coded representation of immersive media Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)", in Edition 1.0 (2021)
- 2) ISO/IEC 23090-9: "Information technology – Coded representation of immersive media Part 9: Geometry-based point cloud compression", in Edition 1.0 (2023)
- 3) Recommendation ITU-T H.265: "High efficiency coding", in Edition 1.0 (2013)
- 4) Recommendation ITU-T H.266: "Versatile video coding", in Edition 1.0 (2020)
- 5) Y. Kim and Y. -H. Kim: "Real-Time Video-based Point Cloud Encoding System on a Distributed Platform", in proceedings of 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 593-595 (2023)
- 6) InterDigital: "V-PCC: The First MPEG Codec for Point Cloud Compression", <https://www.interdigital.com/videos/v-pcc-the-first-mpeg-codec-for-point-cloud-compression> (2019) (accessed 2024.12.17)
- 7) The Draco authors: "Draco 3D Graphics Compression", <https://google.github.io/draco/> (2017) (accessed 2024.12.17)
- 8) J. Prazeres, M. Pereira and A. M. G. Pinheiro: "Subjective Quality Evaluation of Point Clouds with 3D Stereoscopic Visualization", in proceedings of 2022 IEEE International Conference on Image Processing (ICIP), pp. 2861-2865 (2022)
- 9) KDDI 総合研究所: "点群圧縮技術の最新の国際標準方式に対応したリアルタイムエンコーダーを開発", <https://www.kddi-research.jp/newsrelease/2022/102401.html> (2022) (accessed 2024.12.17)
- 10) KDDI 総合研究所: "点群圧縮技術の最新の国際標準方式に対応したリアルタイムコーデックによる伝送実験に成功", <https://www.kddi-research.jp/newsrelease/2023/012401.html> (2023) (accessed 2024.12.17)
- 11) 辻智弘, 河村圭: "G-PCC 対応 3D 点群リアルタイム伝送システムの試作開発", 電子情報通信学会大会講演論文集 (2023)
- 12) 中塚智尋: "点群圧縮技術の国際標準方式に対応したリアルタイムエンコーダー", 映像情報メディア学会誌, **78**, 2, pp. 10-13 (2024)
- 13) S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P.A. Chou, R.A. Cohen, M. Krivokuc' a, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan,

- A. Tabatabai, A.M. Tourapis and V. Zakharchenko: "Emerging MPEG Standards for Point Cloud Compression", in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, **9**, 1, pp.133-148 (2019)
- 14) C. Cao, M. Preda, V. Zakharchenko, E.S. Jang and T. Zaharia: "Compression of Sparse and Dense Dynamic Point Clouds-Methods and Standards", in Proceedings of the IEEE, **109**, 9, pp. 1537-1558 (2021)
- 15) "V-PCC codec description", ISO/IEC JTC1/SC29/WG7 N0100 (2020)
- 16) 海野恭平: "3次元グラフィックス符号化", 映像情報メディア学会誌, **77**, 5, pp. 43-47 (2024)
- 17) 足立翔平, 木谷佳隆, 渡辺裕, 河村圭: "V-PCC エンコーダにおけるテクスチャ画像のパッチ間平滑化に関する一検討", 画像符号化シンポジウム/映像メディア処理シンポジウム, P2-09 (2022)
- 18) 木谷佳隆, 海野恭平, 河村圭: "H.266 | VVC 対応 4K/8K リアルタイムコーデックの開発", 映像情報メディア学会誌, **78**, 1, pp. 115-123 (2024)
- 19) "G-PCC codec description", ISO/IEC JTC1/SC29/WG7 N0217 (2022)
- 20) R. L. de Queiroz and P.A. Chou: "Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform", in IEEE Transactions on Image Processing, **25**, 8, pp. 3947-3956 (2016)
- 21) KDDI 総合研究所: "Starlink 活用によるトンネル建設現場からの 3D 点群データのリアルタイム伝送に成功", <https://www.kddi-research.jp/newsrelease/2024/090201.html> (2024) (accessed 2024.12.17)



うんの きょうへい
海野 恭平 2009 年, 東京理科大学理工学部電気電子情報工学科卒業。2011 年, 同大学院理工学研究科修士課程了。2021 年, 同大学院理工学研究科博士課程了。現在, (株) KDDI 総合研究所 3D 空間伝送グループコアリサーチ。主に動画像符号化と点群符号化の研究・開発および国際標準化に従事。博士 (工学)。正会員。



きだに よしたか
木谷 佳隆 2010 年, 名古屋大学工学部電気電子情報工学科卒業。2012 年, 同大学院工学研究科修士課程了。2024 年, 早稲田大学大学院基幹理工学研究科博士課程了。2012 年, KDDI (株) 入社。現在, (株) KDDI 総合研究所先端映像通信グループエキスパート。主に, 動画像符号化方式の研究・開発および国際標準化に従事。博士 (工学)。正会員。



つじ ともひろ
辻 智弘 1997 年, 富山県立大学工学部電子情報工学科卒業。1999 年, 同大学院工学研究科修士課程了。(株) KDDI 総合研究所にて動画像の符号化・編集・伝送に関する研究開発やプロダクト開発に従事後, 現在, KDDI (株) DX・IoT サービス企画部コアスタッフ。主にデジタルツインに関わる DX サービスの企画・提供に従事。正会員。