

# 点群データに対する階層型 H.265/HEVC 圧縮符号化

佐藤 和也<sup>\*1</sup> 上倉 一人<sup>\*2</sup>

## Layered Video Coding Using H.265/HEVC for Point Cloud Data

Kazuya Sato<sup>\*1</sup> Kazuto Kamikura<sup>\*2</sup>

### Abstract

Point cloud data include information about accurate surface shape of objects. In the case of colored point cloud data, RGB information at each coordinate point is included in addition to XYZ positional information. Point cloud has been applied to 3D modeling, medical image processing, DEM (Digital Elevation Model), etc. On the other hand, data size of point cloud can be massive depending on the acquisition of coordinate values with wide range and high accuracy. Therefore an efficient compression method is necessary for storing or transmitting the data. In this paper, we propose a point cloud data format translation method for layered video coding using H.265/HEVC (High Efficiency Video Coding) in order to not only efficiently compress the data, but also easily retrieve the data.

### 1 はじめに

点群データ (Point Cloud) とは、様々な物体の表面形状を 3 次元座標による位置データの集合で表現したものであり<sup>1)</sup>、3 次元レーザスキャナや赤外線を使った奥行きセンサなどによって取得できる。点群データは実際の物体と同じスケールで計測され、医療分野や地理情報システム (Geographic Information System: GIS) における数値標高モデル (Digital Elevation Model: DEM) などで利用されている。点群データは、対象となる撮影範囲が広がったり計測精度が高い場合にはデータサイズが膨大となる。そこで、データサイズを削減するための圧縮符号化技術が有効となる。データ圧縮符号化の代表的な技術として LZSS (Lempel Ziv Storer Szymanski)がある<sup>2)</sup>。LZSS はファイルフォーマットの 1 つである ZIP ファイルを生成する際に用いられている可逆圧縮アルゴリズムである。可逆圧縮とは、データを圧縮符号化した後に復号した際、全くノイズが含まれず元のデータと同一のデータが再現できることを意味する。可逆圧縮により削減できるデータ量は、対象データの種類にもよるが一般的に 1/2 程度である。

一方、ノイズは含まれるが大幅な圧縮率を実現できる方式 (非可逆圧縮) として、国際標準化機関で規格化された H.265/HEVC (High Efficiency Video Coding)がある<sup>3)</sup>。この方式はカラー映像を圧縮符号化するものであり、数十から数百分の 1 のデータ量での圧縮が可能である。本研究で対象とするデータは色付きの点群データである。すなわち、3 次元座標による位置情報に加えて、その位置での色情報も含まれている。色付き点群データのうち色情報については H.265/HEVC を用いることで効率的な圧縮が実現できると考えられる。そこで筆者らは先に、色情報だけでなく位

置情報も合わせた点群データ全体を H.265/HEVC で統一的に圧縮符号化するための変換手法を提案した<sup>4)</sup>。この手法によりデータ量を約 1/10 に削減できたが、変換時の誤差が比較的大きいことによって点群画像に位置ずれが発生してしまうことが課題であった。また、GIS などで利用される点群データは撮影が広範囲に渡っておりデータ量は膨大となる。その中から所望のデータを高速に検索できることが望まれる。そこで本研究では、点群データを高精度かつ階層的に擬似的な色情報に変換して圧縮符号化することにより、検索時には低階層のデータのみを復号することで高速再生を可能とし、所望の場所については高階層のデータも復号し追加することで高品質な点群画像を再生することが可能な手法を提案する。

### 2 関連技術

#### 2.1 点群データ

物体などの表面形状を 3 次元座標点の集合で表したデータである。以前は点群データを取得するために高価な 3D レーザスキャナを必要としたため、広く一般に知られているデータではなかった。しかし近年、Kinect や Xtion のような安価な奥行きセンサが普及したことや、奥行きセンサを実装したスマートフォンが発売されるなど、点群データは一般ユーザにも普及しつつあり、今後の幅広い応用が期待されている。

点群データのファイルフォーマットとして、Wavefront 社の OBJ 形式、3D Systems 社の STL 形式、Stanford の PLY 形式、LIDAR 出力に用いられる LAS 形式などがある。LAS 形式には位置情報や色情報の他に、レーザーポイントを発射したときの GPS 時間やスキャン角度、スキャン方向等

<sup>\*1</sup> 東京工芸大学大学院工学研究科電子情報工学専攻  
2017 年 9 月 25 日 受理

<sup>\*2</sup> 東京工芸大学工学部コンピュータ応用学科教授

```

1 ply
2 format ascii 1.0
3 element vertex 138183
4 property float x
5 property float y
6 property float z
7 property uchar red
8 property uchar green
9 property uchar blue
10 end_header
11 -0.925446 -0.480118 3.975000 243 244 243
12 -0.925446 -0.473160 3.975000 244 244 244
13 -0.925446 -0.466202 3.975000 230 244 245
14 -0.918487 -0.487077 3.975000 243 242 243
15 -0.918487 -0.480118 3.975000 244 243 243
16 -0.918487 -0.473160 3.975000 245 245 243
17 -0.918487 -0.466202 3.975000 226 236 244
18 -0.918487 -0.459244 3.975000 208 211 219
19 -0.918487 -0.452286 3.975000 214 209 204

```

図1 PLY フォーマットによる点群データの例

の情報も付加されている。図1にPLY形式の点群データの例を示す。同図において1～10行目はヘッダ情報であり11行目以降が実データである。11行目以降の各ラインにおいて、左の3つの数字が計測点の位置情報に相当するX, Y, Z座標値を表しており、右の3つの数字が計測点の色情報に相当するR(赤), G(緑), B(青)強度値を表している。本研究ではヘッダ情報を除いた実データのみを圧縮符号化の対象とする。

## 2.2 H.265/HEVC

H.265/HEVCは、国際標準化団体であるITU-T (International Telecommunication Union Telecommunication Standardization Sector)とISO (International Organization for Standardization), IEC (International Electrotechnical Commission)が共同で規格化した最新の動画圧縮符号化方式である。原画像の冗長部分を削除し、画質に影響の少ない誤差を許容することでデータサイズを削減する不可逆な圧縮技術である。H.265/HEVCに実装されている技術の基本要素として、図2に示すようにフレーム内/フレーム間予測、動き検出、DCT (Discrete Cosine Transform: 離散コサイン変換)/逆DCT、量子化/逆量子化、エントローピー符号化が挙げられる。また、予測やDCTは4×4画

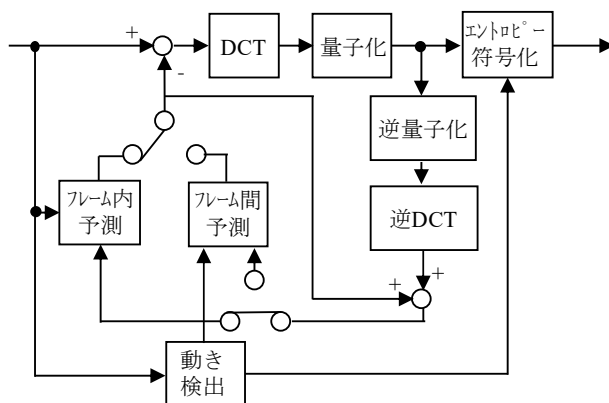


図2 H.265/HEVCの基本構成

素から64画素×64画素までの様々なサイズで処理することができる。これらの処理サイズを対象画像の性質に合わせて適切に選択することにより効率的な符号化が可能となっている。図2では省略されているが、上述した処理サイズを含めた様々な選択情報も付加情報としてエントローピー符号化される。

H.265/HEVCでは、YUVと呼ばれる色空間で表現された信号を処理対象としている。YUVの各値とRGBの各値との関係はITU-R (International Telecommunication Union Radiocommunication Standardization Sector)のBT.709<sup>5)</sup>という規格により式(1)のように定義されている。

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1146 & -0.3854 & 0.5000 \\ 0.5000 & -0.4542 & -0.0458 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

したがって本研究では、XYZ位置情報を擬似的なRGB色情報に変換した後、H.265/HEVCで圧縮符号化が行えるように式(1)によって擬似的なYUV値に変換する処理を行う。

## 3 提案手法

提案手法の概要を図3に示す。まず、点群データのXYZ値(位置情報)を16bitに量子化し、その値を擬似RGB値とする。X値を例にして具体的に述べると、点群データ内の全てのX値から最小値 $X_{min}$ と最大値 $X_{max}$ を求め、式(2)により定数 $a_X$ と $b_X$ を求める。

$$\begin{cases} 0 = a_X \cdot X_{min} + b_X \\ 2^{16} - 1 = a_X \cdot X_{max} + b_X \end{cases} \quad (2)$$

次に、点群データの各X値に対応する擬似的なR値を、定数 $a_X$ と $b_X$ の値を用いて式(3)により算出する。

$$R = a_X \cdot X + b_X \quad (3)$$

Y値、Z値に対してもX値と同様に定数 $a_Y$ と $b_Y$ 、および定数 $a_Z$ と $b_Z$ を求め、Y値に対して擬似的なG値、Z値に対して擬似的なB値を算出する。以上の処理により、ある一点の位置情報であるXYZ値が各々 $0 \sim 2^{16}-1$  (65,535)の範囲の

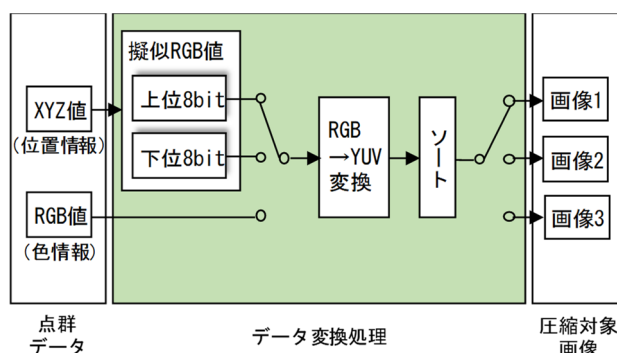


図3 提案手法の構成概要

RGB 値に変換される．これら 3 値のセットをここでは擬似 RGB 値と呼ぶ (図 3)．擬似 RGB 値は，各々が符号なし整数型 16 bit 精度であり，これを上位側である MSB (Most Significant bit) 8 bit と下位側である LSB (Least Significant bit) 8 bit に分ける (図 3)．これにより各値は符号なし整数型 8 bit 精度すなわち  $0 \sim 2^8 - 1$  (255) の値となる．XYZ 値から擬似 RGB への変換例を図 4 に示す．図 4(a) は元の XYZ 値であり，左端の値は計測点のインデックス番号，残りの 3 値は左から X, Y, Z 座標の値である．同図には先頭 10 点のデータのみを表示しているが，実際にはそれ以降にも多数のデータが存在している．これらの点に対して 16 bit 精度で擬似 RGB に変換したものを 10 進数表示したものが図 4(b) である．さらに，それらの上位 8 bit のみを取り出し 10 進数表示したものが図 4(c)，下位 8 bit のみを取り出し 10 進数表示したものが図 4(d) である．

上位 8 bit と下位 8 bit の擬似 RGB 値，および元々の色情報である RGB 値は，それぞれ式(1)により YUV 値に変換される (図 3)．ただし，式(1)のとおりに計算すると Y 値

1	-0.925446	-0.480118	3.975000
2	-0.925446	-0.473160	3.975000
3	-0.925446	-0.466202	3.975000
4	-0.918487	-0.487077	3.975000
5	-0.918487	-0.480118	3.975000
6	-0.918487	-0.473160	3.975000
7	-0.918487	-0.466202	3.975000
8	-0.918487	-0.459244	3.975000
9	-0.918487	-0.452285	3.975000
10	-0.911529	-0.487077	3.975000

(a) XYZ 値

1	0	26712	65535↓
2	0	26900	65535↓
3	0	27088	65535↓
4	212	26524	65535↓
5	212	26712	65535↓
6	212	26900	65535↓
7	212	27088	65535↓
8	212	27276	65535↓
9	212	27464	65535↓
10	425	26524	65535↓

(b) 擬似 RGB 値 (16 bit 精度)

1	0	104	255↓
2	0	105	255↓
3	0	105	255↓
4	0	103	255↓
5	0	104	255↓
6	0	105	255↓
7	0	105	255↓
8	0	106	255↓
9	0	107	255↓
10	1	103	255↓

(c) 擬似 RGB 値  
(上位 8 bit)

1	0	88	255↓
2	0	20	255↓
3	0	208	255↓
4	212	156	255↓
5	212	88	255↓
6	212	20	255↓
7	212	208	255↓
8	212	140	255↓
9	212	72	255↓
10	169	156	255↓

(d) 擬似 RGB 値  
(下位 8 bit)

の値域は  $0 \sim 255$ ，U 値および V 値の値域は  $-128 \sim 127$  となるため，U 値および V 値については値域が  $0 \sim 255$  となるように 128 を加算する．

前述の処理によって得られた擬似 YUV 値 (上位 8 bit，下位 8 bit)，および本来の色情報を表している YUV 値は，隣接点間の相関が高くなるようにソートが施される (図 3)．本手法では，擬似的な YUV 値 (上位 8 bit) に対してユークリッド距離が最短となる順に並び替えることによってソートを行う．すなわち，擬似的な YUV 値 (上位 8 bit) の 1 番目 (1 行目) の点とユークリッド距離が最も近い点を探し，その点を 2 番目 (2 行目) の点として並び替える．ユークリッド距離  $d$  は図 5 において式(4)により求められる．

$$d = \sqrt{(Y_2 - Y_1)^2 + (U_2 - U_1)^2 + (V_2 - V_1)^2} \quad (4)$$

図 6 にソートの例を示す．ソートを行う前 (図 6(a)) では，例えば 3 番目と 4 番目のように  $d$  が比較的大きいデータが隣接しているが，ソートを行った後 (図 6(b)) では  $d$  が小さくなるように並び替わっていることが分かる．

擬似 YUV 値 (下位 8 bit) と，本来の色情報を表している YUV 値については，擬似 YUV 値 (上位 8 bit) のソー

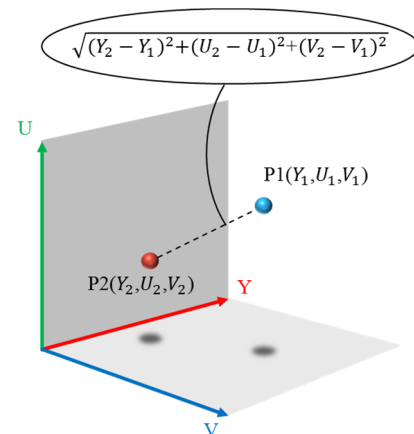


図 5 ユークリッド距離

1	93	215	69↓
2	94	215	69↓
3	94	215	69↓
4	92	216	70↓
5	93	215	69↓
6	94	215	69↓
7	94	215	69↓
8	94	215	68↓
9	95	214	68↓
10	92	216	70↓

(a) ソート前

1	93	215	69↓
2	93	215	69↓
3	94	215	69↓
4	94	215	69↓
5	94	215	69↓
6	94	215	69↓
7	94	215	69↓
8	94	215	69↓
9	94	215	68↓
10	95	214	68↓

(b) ソート後

図 4 XYZ 値から擬似 RGB 値への変換例

図 6 ソートの例



ト結果と同じ順序で並び替える。このようにすることによって、元の点群データにおける各測定点の座標情報と色情報の対応関係を保つことが可能となる。

ソートされた3種類の擬似 YUV 値は、それぞれラスタースキャンの順序に従って2次元画像の画素値として配置される(図3)。擬似 YUV 値(上位8 bit)を配置したものを画像1、擬似 YUV 値(下位8 bit)を配置したものを画像2、本来の色情報を表す YUV 値を配置したものを画像3とする。画像のサイズについては、代表的な画像フォーマットの中から選択する。具体例を図7に示す。図7(a)は圧縮符号化対象の点群データから構築した3D画像であり、データ総数は222,328である。それに対して、図7(b)～(d)では代表的な画像フォーマットの中からHVGAW(Half VGA Wide; 縦640画素×横360)を選択して画像1～画像3を生成した。このフォーマットにおける画素数は230,400/フレームであるため、各々の画像は1フレームずつ生成すれば良い。なお、残りの(230,400-222,328)画素には最後の値をそのまま入れる。図7(b)は変換後の画像1である。画像1は位置情報を擬似 YUV 値に変換し上位8 bitのみを抜き出したものであるため、その情報に含まれる雑音は比較的多いが相関は高いと考えられる。実際、図7(b)では類似した値が横方向に続いており、画素間の相関が高いことが分かる。したがって、画像1のみを圧縮符号化することで点群画像の概観(雑音が多く含まれている画像)を少ないデータで確認することができるため、高速な検索が可能となる。図7(c)は変換後の画像2である。画像2は擬似 YUV 値の下位8 bitであるため、このデータを画像1に加えることによって位置情報はより正確になるが、画像2そのものの画素間相関はほとんど無いと考えられる。実際、図7(c)では画素の色や明るさが細かく変化しており、画素間の相関が低いことが分かる。したがって、画像2に対して圧縮符号化を行ってもデータ圧縮の効果がない、もしくは逆にデータ量が増加してしまうという結果であった。図7(d)は変換後の画像3である。画像3は、通常の画像圧縮符号化と同様に、隣接する位置の色情報を表しているため、圧縮符号化の効果が高いと考えられる。

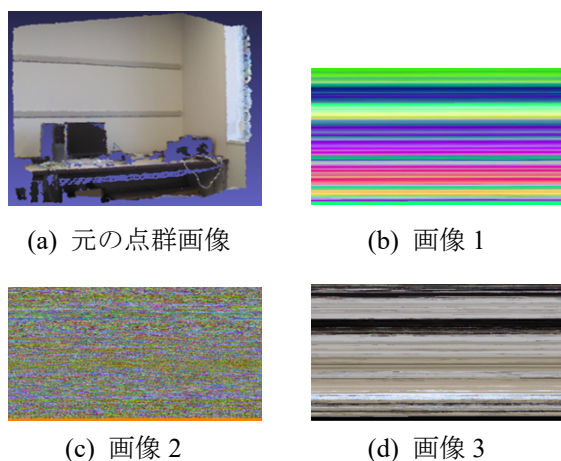


図7 点群画像と変換後の画像1～画像3

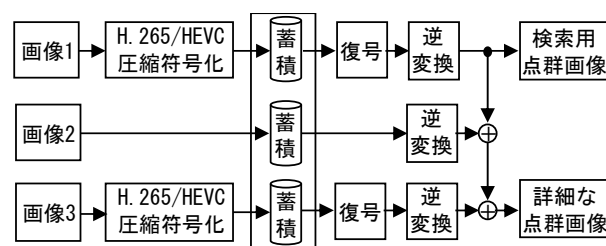


図8 圧縮符号化・蓄積と復号・再生の構成

上記の記述をまとめると、本手法では、画像1と画像3に対してはH.265/HEVCによる圧縮符号化処理を行い、画像2に対しては圧縮符号化処理を行わない。画像1～3に対する圧縮符号化・蓄積処理と、そのデータを復号して点群データによる画像に再構築するための構成を図8に示す。前述したとおり、画像1と画像3に対してはH.265/HEVCによる圧縮符号化を行い、そのデータを蓄積する。画像2に対しては圧縮符号化を行わず、そのままのデータを蓄積する。点群データを利用する際には、まず画像1の圧縮データのみを復号し図3と反対の手順で逆変換することにより、粗い点群データ画像を高速に表示しながら検索することができる。所望のデータを見つけたら、それに対応する画像2、画像3のデータを復号または逆変換して画像1のデータに加えることにより、詳細な点群データ画像を再生することができる。

#### 4 実験と考察

Microsoft社のKinectを用いて大学研究室内を撮影し、MATLABでascii plyフォーマットで出力することにより、実験に用いる点群データを作成した。ascii plyフォーマットはテキストデータである。点群総数は138,183、座標値は32ビット(4バイト)の浮動小数点型で保持されている。作成した点群データから構築した3D画像を図9に示す。このplyファイルからヘッダ部分を削除して入力点群データとした。点群データを変換・逆変換するソフトウェアは、統合開発環境Microsoft Visual Studioとプログラミング

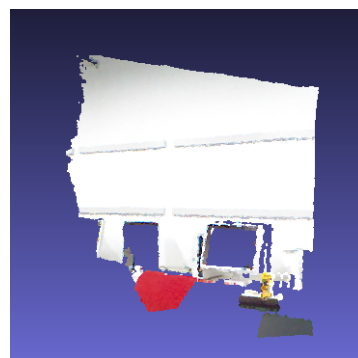


図9 実験に用いた点群データ

表 1 各処理によるデータサイズ (単位: kB)

元の点群データ	2,072.7	
	YUV 変換後	圧縮符号化後
画像 1	460.8	65.3
画像 2	460.8	(460.8)
画像 3	460.8	132.2
トータル	1,382.4	658.3

グ言語 C++を使用して開発した。点群データ変換後の画像 1～画像 3 のフォーマットは HVGA (Half VGA; 横 480 画素×縦 320 画素)を用いた。その画素数は 153,600/フレームであるため、実験で用いた点群データは 1 フレームに収まる。

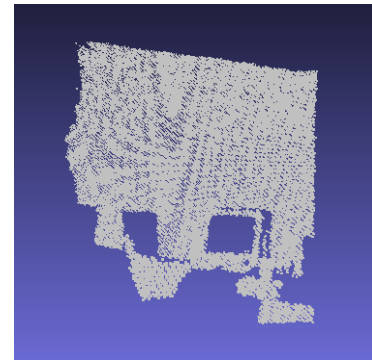
変換した画像に対する H.265/HEVC の圧縮符号化には、国際標準化機関 ISO/IEC で作成・維持管理されているソフトウェアエンコーダ HM(HEVC Test Model)<sup>6)</sup>を用いた。HM において、圧縮画像品質を決めるパラメータである QP は 0 (最高品質) に設定した。

各々の処理結果によるデータサイズを表 1 に示す。また、画像 1 のみから再構築された点群画像 (検索用)、および画像 1～3 から再構築された点群画像を図 10 に示す。表 1 において、元の点群データのサイズは XYZ 値が各々 4 バイト、RGB 値が各々 1 バイトとして算出している。画像 2 については圧縮符号化を行わないため YUV 変換後のデータサイズと同一としている。表 1 より、画像 1 のデータは元の点群データの 1/30 程度に削減されていることが分かる。図 10(a)より、画像 1 のみから再構築された点群画像は点の欠落が多いが、概観を知ることは可能であり、検索用としては問題ないといえる。一方、画像 1～3 のトータルデータは元の点群データの 1/3 程度に削減できている。そのデータから再構築された点群画像 (図 10(b)) は、点の欠落が減り色情報も保持されていることが分かる。しかしながら、点群データでは点の欠落が許容されない用途もあると思われる、今後は、そのような用途に対応するための改善が必要と考えられる。

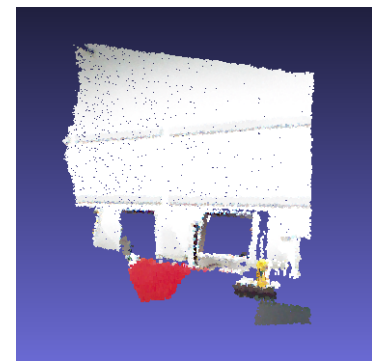
## 5 むすび

点群データを画像形式に変換する過程においてデータを階層構造とすることにより、点群データの概観を高速に検索し、所望のデータ部分について更に高品質な点群データを再構築する手法を提案した。実験の結果、検索用データは元の約 1/30、高品質な点群データは元の約 1/3 程度のサイズで再構築できることを確認した。

本提案手法では、データ変換におけるソート処理に時間を要している。その処理に対する負荷軽減が今後の課題である。また、検索時の高速化に関する実証実験も今後の課題といえる。



(a) 検索用点群画像  
(画像 1 のみから再構築)



(b) 詳細な点群画像  
(画像 1～3 から再構築)

図 10 再構築した点群データ

## 参考文献

- 1) Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," Proceedings of SIGGRAPH 2000, pp. 131-144, July 2000.
- 2) James Andrew Storer and Thomas Gregory Szymanski, "Data Compression via Textual Substitution," J. ACM, 29, 4, pp.928-951, October 1982.
- 3) ITU-T Recommendation "H.265: High efficiency video coding" October 2016.
- 4) 佐藤和也, 上倉一人, 清水淳, "点群データに対する H.265/HEVC 圧縮のためのデータ形式変換," 画像関連学会連合会第 3 回秋季大会予稿集, PC19, 2016 年 11 月.
- 5) ITU-R Recommendation "BT.709: Parameter values for the HDTV standards for production and international programme exchange" June 2015.
- 6) ITU-T Recommendation "Reference software for ITU-T H.265 high efficiency video coding" October 2016.