

# TTA Standard

정보통신단체표준(국문표준)

TTAx.xx-xx.xxxx/R1

제정일: 2019년 6월 28일

다중 패치로 구성된 스캔 포인트  
클라우드 데이터 압축 파일포맷

A Compression File Format for 3D Scanned  
Point Cloud Data Composed with Multi-Patch



한국정보통신기술협회  
Telecommunications Technology Association

표준초안 검토 위원회 디지털콘텐츠 프로젝트그룹(PG610)

표준안 심의 위원회 소프트웨어/콘텐츠 기술위원회(TC6)

	성명	소속	직위	위원회 및 직위	표준번호
표준(과제) 제안	이승욱	한국전자통신연구원	책임연구원	PG610 위원	
표준 초안 작성자	이승욱	한국전자통신연구원	책임연구원	PG610 위원	
	남현우	동덕여자대학교	교수	PG610 의장	
	이범렬	한국전자통신연구원	책임연구원	PG610 부의장	
	정상권	조이편	대표이사	PG610 위원	
사무국 담당	김찬영	TTA	전임연구원	-	

본 문서에 대한 저작권은 TTA에 있으며, TTA와 사전 협의 없이 이 문서의 전체 또는 일부를 상업적 목적으로 복제 또는 배포해서는 안 됩니다.

본 표준 발간 이전에 접수된 지식재산권 협약서 정보는 본 표준의 '부록(지식재산권 협약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 협약서는 TTA 웹사이트에서 확인할 수 있습니다.

본 표준과 관련하여 접수된 협약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장

발행처 : 한국정보통신기술협회

13591, 경기도 성남시 분당구 분당로 47

Tel : 031-724-0114, Fax : 031-724-0109

발행일 : 2019.6.28

# 서 문

## 1 표준의 목적

이 표준의 목적은 스캐너를 이용하여 다중 패치로 생성된 포인트 클라우드 데이터를 저장할 때 압축하는 방법을 제공함으로써, 효율적으로 파일을 저장하는데 있다.

## 2 주요 내용 요약

이 표준은 다중으로 스캔된 데이터를 저장하는 바이너리 기반의 파일 형식을 정의하고 관련된 비트스트림 신택스와 시멘틱스를 정의한다.

## 3 인용 표준과의 비교

### 3.1 인용 표준과의 관련성

해당사항 없음

## Preface

### 1 Purpose

The standard is aimed to store the multi-patch scanned point cloud data efficiently by providing a compression method.

### 2 Summary

The standard defines a binary fileformat that stores the multi-patch scanned data and the related syntax and semantics.

### 3 Relationship to Reference Standards

N/A

## 목 차

1 적용 범위 .....	1
2 인용 표준 .....	1
3 용어 정의 .....	1
4 약어 .....	1
5 다중패치로 구성된 포인트 클라우드의 압축 파일포맷 .....	2
5.1 포인트 클라우드 파일의 비트스트림 구조 .....	2
5.2 비트스트림의 신택스 및 시맨틱스 .....	2
부속서 A (자유 작성 부속서) 제목 .....	7
부록 I 서비스 시나리오 및 파일예제 .....	8
부록 II-1 지식재산권 협약서 정보 .....	10
II-2 시험인증 관련 사항 .....	11
II-3 본 표준의 연계(family) 표준 .....	12
II-4 참고 문헌 .....	13
II-5 영문표준 해설서 .....	14
II-6 표준의 이력 .....	15

# 다중 패치로 구성된 스캔 포인트 클라우드 데이터 압축 파일포맷 (A Compression File Format for 3D Scanned Point Cloud Data Composed with Multi-Patch)

## 1 적용 범위

본 표준은 스캐너로 생성한 포인트 클라우드 데이터를 저장할 때 사용되는 비트스트림의 구조, 헤더정보, 데이터 정보 및 관련된 신택스와 시맨틱스를 기술한다. 일반적으로 사용자는 다중 패치를 생성하게 되는데, 본 표준을 통해 다중패치를 압축하여 저장할 수 있다.

## 2 인용 표준

해당사항 없음

## 3 용어 정의

### 3.1 포인트 클라우드 (Point Cloud)

동일한 좌표계에서 정의되는 점들의 집합

※ 일반적으로 2D, 3D 포인트 클라우드 등을 정의하며, 본 표준에서는 3D 포인트 클라우드를 대상으로 하며, 각 점들을 연결하는 삼각형 등으로 만들어 기존의 컴퓨터 그래픽스 작업을 수행함.

### 3.2. 스캔 패치

3D 스캐너에서 한 번의 연속적인 작업을 통해 생성되는 포인트 클라우드 데이터의 집합

### 3.2. 패치 정합

각각의 스캔 패치를 동일한 좌표계로 정렬하여 하나의 포인트 클라우드 데이터를 만드는 과정

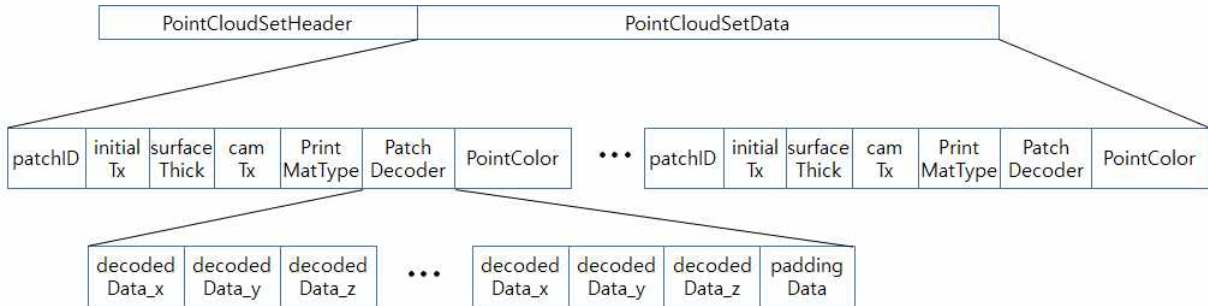
※ 일반적으로 스캐닝 작업시 다수의 스캔 패치를 생성하고, 이를 연결하는 작업, 즉 패치 정합을 통해 하나의 포인트 클라우드 데이터를 생성

## 4 약어

FDM Fused Deposition Modeling

## 5 다중패치로 구성된 포인트 클라우드의 압축 파일포맷

### 5.1 포인트 클라우드 파일의 비트스트림 구조



(그림 5-1) 비트스트림 구조

그림 5-1은 전체 파일의 스트림 구조를 나타낸다. 일반적인 비트스트림 형태와 같이 헤더(PointCloudHeader) 및 데이터(PointCloudData)로 구성되며, 데이터는 패치 개수만큼의 정보가 반복된다. 헤더에는 비트스트림을 구성하는 일반적 내용이 기술되며, 데이터 부분에 실제 패치로 구성된 포인트 클라우드 정보가 기술된다.

### 5.2 비트스트림의 신택스 및 시맨틱스

본 절에서는 바이너리 기반의 포인트 클라우드 파일포맷의 신택스 및 시맨틱스를 정의한다.

#### 5.2.1 PointCloudSet

포인트 클라우드 데이터 파일포맷의 최상위 노드정보이다.

##### 5.2.1.1 신택스

```

class PointCloudSet {
    PointCloudSetHeader    header;
    PointCloudSetData      data;
}
    
```

##### 5.2.1.2 시맨틱스

- PointCloudSet : 바이너리 기반 포인트 클라우드 데이터의 최상위 노드
- header: PointCloudSet의 헤더정보
- data: PointCloudSet의 실제 데이터 정보

#### 5.2.2 PointCloudSetHeader

##### 5.2.2.1 신택스

```

class PointCloudSetHeader {
    unsigned int(32)    streamSize;
}
    
```

```

unsigned char(1024)  copyRight;
unsigned char(1024)  acquisitionAlgo;
bit(8)  unit;
float(32)  minimumVerticalResolution;

unsigned int(32)  numOfPatch;
if (numOfPatch>0)
{
    for (int i = 0; i < numOfPatch; i++)
    {
        unsigned int(32)  numOfPoint[i];
        unsigned int(32)  numOfColor[i];

        if (numOfPoint[i] > 0)
        {
            for (j = 0; j<3; i++) {
                float(32)  quantMinPoint[i];
                float(32)  quantRangePoint[i];
            }
        }
    }

    bit(8)  QPforPoint;
}

```

### 5.2.2.2 시멘틱스

- streamSize: PointCloudSet의 전체 파일 크기를 바이트 단위로 나타 냄
- copyRight: 스트링 형태로 정의된 저작권 정보를 나타낸다. 3D 스캐너의 경우 데이터만 있으면 제품을 생산 할 수 있기에 저작권의 정보가 중요하다.
- aquisitionAlgo: 스트링 형태로 정의된 포인트 클라우드 생성 알고리즘을 정의한다. 레이저 방식, 패턴 방식, 스테레오 방식 등 다양한 부가 정보를 삽입할 수 있다.
- unit: 8비트의 정보로 실제 사용되는 포인트의 3차원 좌표에 정의된 값의 단위(unit)을 정의한다. 만약 이 값이 mm로 정의되면, 모든 좌표는 mm로 해석된다.
- minimumVerticalResolution: 3D 모델 획득시의 해상도를 나타낸다. 디자이너에 의해 생성된 모델이 아닌 스캐너를 통해 만들어진 모델의 경우에서 사용되는 스캔 해상도를 의미한다. 이 값이 0.1로 설정된 경우 이 값보다 작은 해상도의 출력 해상도(y-축 적층 해상도)는 의미가 없다. 즉, 이 값은 수직방면의 최소 출력 해상도를 나타낸다.
- numOfPatch: 패치의 개수를 정의한다. 만약 이 값이 1이면 정합된 패치 혹은 한번에 스캔된 모델을 의미하고, 이 값이 2 이상이면, 패치 정합 과정을 통해 하나의 포인트 클라우드로 정렬하는 작업이 필요하다.
- numOfPoint: 각 패치에 존재하는 포인트의 개수
- numOfColor: 각 패치에 존재하는 포인트 컬러의 개수, 이 값은 numOfPoint와 같은 값이거나 '0'이다. 컬러를 지원하지 않는 포인트 클라우드 데이터도 존재하기 때문이다.
- quantMinPoint : 양자화 과정에서 사용되는 포인트의 최소값을 정의한다.
- quantRangePoint : 양자화 과정에서 사용되는 범위값을 정의한다.



<표 5-1. unit에 대한 시멘틱 정보>

값	정보
0	millimeter
1	centimeter
2	meter
3	inch
4	feet
5	yard
나머지	reserved

### 5.2.3 PointCloudSetData

#### 5.2.3.1 선택스

```
class PointCloudSetData {
    if (numOfPatch>0) {
        for (int i = 0; i < numOfPatch; i++)
        {
            unsigned int(32) patchID;
            for (int j = 0; j < 12; j++)
            {
                float(32) initialTransform[j];
            }
            float(32) surfaceThickness;
            for (int j = 0; j < numOfPoint[i]; j++)
            {
                float(32) camTransform[j];
            }

            for (int j = 0; j < numOfPoint[i]; j++)
            {
                bit(8) printingMaterialType[j];
            }

            EachPatchDecoder(numOfPoint[i], 3, QPforPoint);
            if (numOfColor[i] > 0)
            {
                for (int j = 0; j < numOfColor[i]; j++)
                    bit(32) decodedcolorOfPoint[j];
            }
        }
    }
}
```

#### 5.2.3.2 시멘틱스

- patchID : 각 패치의 ID 값을 나타낸다. 이 ID는 한 포인트 클라우드내에서 중복되지 않는다.
- initialTransform : 패치를 스캔할 때 초기 카메라의 위치를 정의하는 3x4 외부 캘리브레이션 파라미터이며, column-base 형태로 저장된다. 이 값은 후처리 과정을 통해 구해질 수도 있음

나, 본 표준에서는 센서등의 값에 의해 스캔할 때 바로 정의하는 값이다.

- surfaceThickness : 모델의 추천 두께를 나타낸다. 예를 들어 두께가 5인 실린더의 경우 실린더 내부를 비우고 두께는 5로 프린트 하고, 이때의 단위는 unit의 단위를 사용한다.
- camTransform : 스캔된 패치내의 정점정보를 스캔한 스캐너의 위치 정보를 나타내며, coord에 정의된 정점 각각에 하나씩 정의된다. 즉 정점의 개수와, camTransform내의 1x12 벡터의 개수는 같다. 스캐너의 위치 정보는 3x4의 외부 캘리브레이션 정보가 column-base 형태로 저장된 것이다.
- printingMaterialType : 프린팅 재질 정보를 나타내는 값으로, 사용자의 선택에 따라 옵션으로 사용하며, 상호간 미리 정의된 정보를 입력한다.
- decodedColorOfPoint : 각 포인트에서 스캔된 포인트 클라우드 데이터의 컬러 정보를 나타낸다. RGBA 순서로 값을 정의된다. 알파의 경우 1이 완전 투명한 것이다. 각각의 값은 다음과 같이 계산된다.

R(Red value) = decodedColorOfPoint & 0xFF000000) >> 24  
 G(Green value) = (decodedColorOfPoint & 0x00FF0000) >> 16  
 B(Blue value) = (decodedColorOfPoint & 0x0000FF00) >> 8  
 A(Alpha value) = (decodedColorOfPoint & 0x000000FF) >> 0

## 5.2.4 EachPatchDecoder

### 5.2.4.1 신택스

```
class EachPatchDecoder(numData, dim, QP) {
    for (int i = 0; i<numData; i++) {
        for (int j = 0; j<dim; j++) {
            bit(QP) decodedData[j];
        }
    }
    unsigned int (32) numPaddingBits;
    bit(numPaddingBits) paddingData;
}
```

### 5.2.4.2 시멘틱스

- decodedData : 각 x, y, z 좌표계 상에서의 디코딩된 포인트 값을 정의한다. 양자화 및 차분 코딩을 이용하여 복원한다. 즉, 첫 번째 디코딩 되는 경우는 역양자화만 이용하고 두 번째 부터는 역양자화를 수행한 후 이전 값을 더한다. 첫 번째 복원되는 값인지 여부에 따라 다음과 같이 계산된다.

case 1. 첫번째 값인 경우:

$$V_r = [v_q / (2^{QP} - 1)] * quantRange + quantMin$$

case 2. 첫 번째 이후의 경우

$$V_r = V_{r\_pre} + [v_q / (2^{QP} - 1)] * quantRange + quantMin$$

V\_r: 역 양자화 과정을 거친 최종 결과 값

v\_q: 양자화 된 입력 값

quantRange: 각 x, y, z축의 양자화 범위 값, 헤더 정보에 있는 quantRangePoint 값

quantMin: 각 x, y, z축의 최소값, 헤더 정보에 있는 quantMinPoint 값

V\_r\_pre: 이전의 디코딩된 값

- paddingData : numPaddingBits 비트만큼의 크기를 가지는 패딩 데이터로 8-bit alignment를 위한 패딩비트로 사용되며, 추출된 정보는 사용하지 않고 버린다

## 부 속 서 A

(본 부속서는 표준 내용의 일부임)

## 부 록 I

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 서비스 시나리오 및 파일예제

#### 1.1 스캐너를 이용한 다중 패치의 포인트 클라우드 생성 과정 및 시나리오

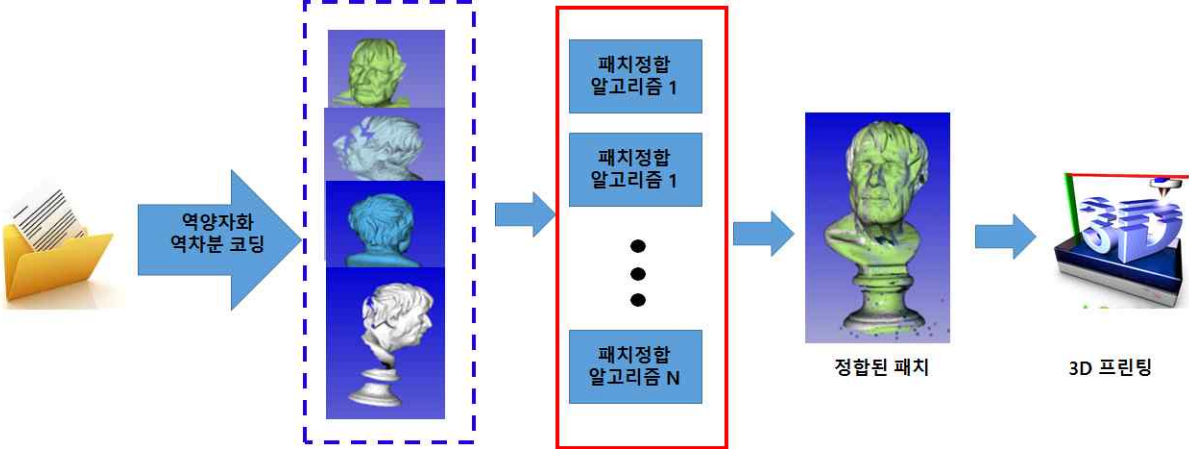
그림 1-1은 스캐너 기반의 포인트 클라우드 데이터 생성 과정을 설명한다.



(그림 1-1) 일반적인 스캐닝 작업 과정

먼저 스캐너를 이용하여 N번의 스캐닝 작업을 통해 N개의 스캔패치를 생성한다. 본 예제에서는 4개의 스캔 패치를 만든다. 각각의 스캔 패치를 연결하여 하나의 포인트 클라우드 데이터로 만들기 위해 패치 정합작업을 수행하는데, 이때 여러 가지 방법이 (그림 1-2의 붉은색 사각형) 사용될 수 있다. 패치 정합은 각 패치간의 행렬 관계를 계산하는 작업임으로, 각 패치에서 사용된 초기 행렬 정보를 스캐너의 센서로부터, 혹은 다른 방식으로 획득할 수 있으면 패치 정합과정이 쉬워진다.

그림 1-2는 본 표준에서 정의되는 포인트 클라우드 파일을 이용한 시나리오이다. 스캔을 통해 생성된 수천 패치의 모임을 하나의 표준 프로젝트 파일로 묶어 전송 및 공유하고, 이를 다양한 정합 알고리즘을 통해 최종 정합 파일을 만들어 3D 프린팅으로 진행할 수 있다. 정합하기 전에 역양자화 및 역차분 부호화를 통해 스캔된 패치를 복원하고, 정합을 수행한다.



(그림 1-2) 포인트 클라우드 프로젝트 파일을 이용한 프린팅 시나리오

## 부 록 II-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 지식재산권 협약서 정보

해당사항 없음

※ 상기 기재된 지식재산권 협약서 이외에도 본 표준이 발간된 후 접수된 협약서가 있을 수 있으니, TTA 웹사이트에서 확인하시기 바랍니다.

## 부 록 II-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 시험인증 관련 사항

#### II-2.1 시험인증 대상 여부

해당사항 없음

#### II-2.2 시험표준 제정 현황

해당사항 없음



## 부 록 II-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 본 표준의 연계(family) 표준

해당사항 없음

## 부 록 II-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 참고 문헌

해당사항 없음

## 부 록 II-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 영문표준 해설서

해당사항 없음

## 부 록 II-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

### 표준의 이력

판수	채택일	표준번호	내용	담당 위원회
제1판	2019.6.28	제정 TTAx.xx-xx.xxxx	-	디지털콘텐츠 프로젝트그룹 (PG610)