

WEB ブラウザを用いた CityGML の 3 次元表示システムに関する研究

野崎祐樹¹ 碓崎賢一² 荒木俊輔³

概要: 都市開発計画の分野では、都市の幾何情報だけでなく土地の用途、災害リスクなどの意味情報を CityGML という国際規格で符号化し、記述・交換に利用している。現実の都市を CityGML を用いて仮想空間上に再現することで、近年関心が高まっているメタバースやデジタルツイン、Society5.0 などの実現に活用できると期待されている。その一方で、WEB ブラウザを用いた CityGML の 3 次元表示システムは別のデータ形式に変換する必要があるものが一般的であり、データ変換に手間がかかる。本研究では、CityGML を他のデータに変換することなく読み込み、3 次元表示する WEB ブラウザシステムについて提案する。

キーワード: CityGML, GIS, 都市開発計画, 街づくり,

A Research on 3D display system of CityGML using web browser

Yuki NOZAKI¹ Kenichi KAKIZAKI² Syunsuke ARAKI³

Abstract: In the field of urban development planning, not only geometric but also semantic information, such as land use and disaster risk, is encoded in an international standard called CityGML and used for description and exchange. Society 5.0 has been the focus of increasing interest in recent years. However, most CityGML 3D display systems using a web browser need to be converted to another data format, which is time consuming. Therefore, this study proposes a web browser system that reads and displays CityGML 3D data without converting to another data.

Keywords: Geographical databases and spatial data management, Information visualization, Information extraction/summarization/restructuring

1. はじめに

近年では、多彩なデータを仮想空間上にて再現する取り組みが注目されている。都市開発の分野も例外ではなく、近年関心が高まっているメタバースやデジタルツイン、Society5.0 などの実現に活用するため、実際の都市の情報を仮想的な空間上に再現する活動を進めている。国土交通省では、projectPLATEAU [1]と呼ばれるプロジェクトを推進しており、日本各地の都市を CityGML (City Geography Markup Language) という国際符号化規格を用いて 3 次元データ化し、整備、管理を行っている。

都市計画や景観計画、スマートシティやデジタルツインなどの取り組みが注目されている中、地理情報都市を 3 次元のモデルとして管理、交換する場合、一般的な 3DCG で用いられる OBJ や FBX などのデータフォーマットによる表現を用いることが考えられる。しかし、都市開発や災害対策、シミュレーションなどの用途に利用する場合、上記のようなデータ形式では幾何形状の記述のみを扱うため、地理的なセマンティクスや地物間の関係性を記述する

ことができない。そのため、3 次元都市モデルの応用は可視化によるものに限られていた。3 次元都市モデルをより汎用的に利用するためには、純粋な幾何情報のみならず、都市活動に関わる意味情報も同時に扱えるデータ符号化基準が必要となる。また、都市レベルのモデルを作成するためには広範囲かつ複数種類のデータが必要となるため、莫大な時間と費用が掛かる。また、データの整備も容易ではない。そのため、データの作成や整備に関して、さまざまな分野の多様な利用者で共有利用し、効率的に維持更新することが望ましい。本研究では、この CityGML に着目し、CityGML の情報を WEB ブラウザ上で閲覧するシステムを提案する。

2. CityGML

CityGML は、3 次元都市モデルおよび地形モデルの表現、保存、交換のために Open Geospatial Consortium (OGC) [2] が策定した符号化標準である。物体の幾何情報を記述するためのジオメトリとトポロジーの表現、都市を構築する建造物や橋梁、トンネル、地形およびそれらの外見を定

1 九州工業大学 情報工学府 情報創成工学専攻
Graduate School of Computer Science and Systems Engineering Kyushu
Institute of Technology

2 九州工業大学大学院 情報工学研究院 教授・博士 (工学)
(〒820-8502 飯塚市川津 680-4)
Prof., Graduate School of Computer Science and Systems Engineering, Kyushu
Institute of Technology, Dr. Eng.

3 九州工業大学大学院 情報工学研究院 准教授・博士 (情報工学)
Assoc. Prof., Graduate School of Computer Science and Systems Engineering,
Kyushu Institute of Technology, Dr. Info. Eng.

義する様々なクラスが定義され、地理的なセマンティクスを考慮し地物同士の複雑な関係性や属性情報および3次元表現を提供する [3]。CityGML は 2008 年に最初のバージョンである 1.0 が公開され [4]、本稿執筆時点 (2023 年 2 月) ではバージョン 3.0 [5]まで公開されている。

CityGML は、GML (Geography Markup Language) と呼ばれる XML の文法に基づいたマークアップ言語で記述されており、データの構造を表すスキーマと実際のデータを格納するインスタンスで構成されている。

2.1 LOD

CityGML の大きな特徴の一つとして LOD (Level Of Detail) という概念がある。コンピュータグラフィックスの分野では、主に 3D モデルのポリゴンの細かさを表す指標であり、カメラから遠いオブジェクトの LOD を減らすことで描画負荷の軽減等に利用されるものであるが、CityGML における LOD は、地物セマンティクスを考慮したレベル設計となっており、地物の意味情報を、重要度を考慮してレベルを設定している。使用するデータを、LOD によって絞り込むことで、コンピュータグラフィックスの分野における LOD のように、描画システムにおける負荷の軽減が期待できることはもちろん、用途に合わせたモデルを使用することができる。例えば、地図上に意味情報を元に建物やエリアを色分けによって可視化するというシステムの場合、建物のモデルは、平面や箱型モデルのような単純なモデルの方が見えやすい場合がある。また、景観を考慮した都市開発に関するシステムのように、街の見た目を重視するシステムには、外観の情報も含まれた複雑なモデルが必要である。このように、ユーザーが LOD を元に使うモデルを選択することで、用途に合わせて様々な粒度のモデルを利用することができる。

LOD はバージョン 3.0 時点で、0 から 3 までの 4 段階ある。以前のバージョンでは、屋内の情報も含んだ LOD4 までが存在したが、LOD0~3 までのモデルで屋内要素を表現できるようになったため、バージョン 3.0 にて削除されている。例として、建物のデータの LOD によるモデルの抽象度の違いを比較したものを下の図に示す。

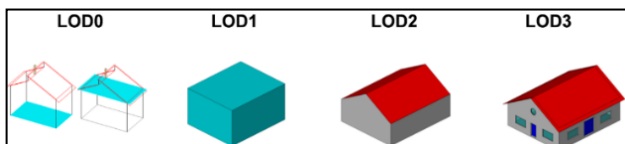


図 1 LOD による抽象度の比較 [6]

2.2 建物以外の地物

CityGML には、建物以外にも、道路や植生、地形などの地物や、洪水、津波、土砂災害などの災害想定区域にかかわる情報などを記述できる。

以下の章では本研究で対象とした地物のクラスについて詳細に説明する。

2.2.1 道路

一般交通に使われる道路を定義する。下の図に示すように、LOD0 では線で、LOD1 では一体的な面として取得される。また、単純な道路の外周の他にも、LOD2 では、歩道や自動車道の区別、さらに LOD3 では交通島や分離帯、路肩など、交通領域の機能を補助するための領域なども記述できる。

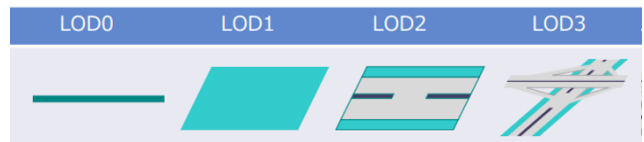


図 2 LOD 別の道路モデル [7]

2.2.2 植生

独立した樹木を定義する。下の図に示すように、LOD1 では樹冠を上に水平投影した面を樹高分立ち上げたような立体で定義する。また、LOD2 では楕円や球体、円錐、角錐、円柱などを組み合わせた簡易的なモデルで定義され、LOD3 ではさらに複雑に独立木の形状を立体として表現する。



図 3 LOD 別の樹木モデル [7]

芝生や茂みのように、植生に覆われた範囲も下の図のようなモデルで定義することができ、植被が占有している範囲を立体形状として表現する。



図 4 LOD 別の茂みモデル [7]

2.2.3 地形

地形の凹凸を表現するクラス。一つの地物あたり、基準地域メッシュ (第三次地域区画、1 辺の長さが約 1 km) の範囲で定義される。地形の表現には、TINRelief と MassPointRelief の 2 種類があり、前者は三角形ポリゴンの集合で地形の起伏を表現する。後者は点の集まりにより地形の起伏を表現する。

3. 表示システムの課題

3次元都市モデルの提供価値の一つとして、「ビジュアルライズ (可視化)」が挙げられる。都市空間を3次元的に可視化できるようになることで、都市に関する様々なデータを視覚的に分かりやすく表現できる。また、都市の現状や課題、将来像を分かりやすく提示できるようになるこ

とで、街づくりに求められる基本的な価値を生み出せる。しかしながら、CityGMLを3次元表示するには、いくつかの課題がある。

3.1 幾何形状の変換

CityGMLの幾何情報を一般的な表示システム (UnityなどのゲームエンジンやMayaなどの3DCGアニメーションソフトウェア) で表示する方法がある。そのためには、CityGMLを他のシステムを用いてOBJやFBXなどの一般的な3DCG用のデータ規格に変換する必要がある。変換手法については、稲留氏の手法 [8] のように幾何情報を抜き出して別フォーマットに従ってデータを作成する方法や、Safe Software社が開発したFME [9] と呼ばれる位置情報データ変換ソフトウェアにて、CityGML変換用ワークスペースを用いる方法などがある。しかし、その場合、前述したような、CityGMLが持っている地物の意味情報は欠落してしまう。また、表示システムに入力するデータをユーザーが変換する必要があるため手間がかかるという問題がある。

3.2 Cesium を利用したシステム

Cesiumは、WEBブラウザ上で動作するGISアプリケーション用の3Dビューアーエンジンである。国土交通省がPLATEAUで公開している「3D都市モデルの導入ガイド第2.0版」 [6] ではこのエンジンを利用した表示を推奨しており、CityGMLの3次元表示システムとして、同じく国土交通省が「PLATEAU VIEW」 [10] と呼ばれるWEBブラウザシステムを公開している。このシステムでは、3次元空間上で幾何形状の他、都市活動に関わる意味情報も扱える。しかしながら、CesiumはCityGMLには対応しておらず、CityGMLを前述したFMEなどで3D Tilesファイルに事前に変換し、データベース上に保管したものを利用する (図4-1)。そのため、厳密にはCityGMLをそのまま入力するシステムではなく、また、データサーバー上のデータではなく、ユーザーが自分で用意したCityGMLデータをそのまま入力することはできないため、4.1章で述べたような入力データの変換をしなければならない。

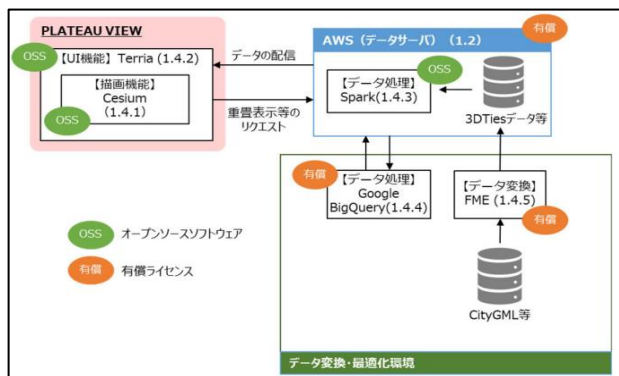


図5 PLATEAU VIEW のシステム構成 [6]

4. 提案手法

4.1 要件

以上の課題から、我々はCityGMLを他のデータフォーマットに変換せずとも利用できる3次元表示システムの開発に着目した。また、WEBブラウザ上で動作するシステムにすることで、プログラムをインストールすることなく、全国どこからでも利用できるように、WEBブラウザを用いたシステムの構築に着手した。

4.2 使用言語・ライブラリ

WEBブラウザ上で稼働するシステムを想定するので、HTMLとJavascriptを用いてシステムの構築を行った。また、幾何情報を3次元空間上に表示する方法として、WEBブラウザ用のグラフィックライブラリであるWebGLと、それらをJavascript上で扱いやすくするラッパーであるthree.jsを用いることにした。これにより、HTMLのCanvas要素に3次元のモデルを表示することが可能になる。

4.3 使用した都市モデル

PLATEAUでは、コミュニティとの連携、官民協働推進、地域間連携やプラットフォーム構築などの効果を期待して、作成した3次元都市モデルを容易に加工、編集、再配布等ができる形にオープンデータ化し、「G空間情報センター」 [11] と呼ばれるプラットフォームで無償公開されている。本研究ではシステムの入力データとしてPLATEAUのCityGMLデータを利用する。

CityGMLは以下の表のようにモジュールが分けられており、モジュール識別のための接頭辞が与えられ、接頭辞ごとにフォルダ分けされている。

表1 接頭辞と意味

接頭辞	意味
bldg.	建築物とその付属物、境界面、開口部
dem	地形(起伏)
tran	道路
veg	植生
ftn	都市設備
urf	都市計画決定情報
luse	土地利用
fld	洪水浸水想定区域
tnm	津波浸水想定区域
lsld	土砂災害警戒区域
htd	高潮浸水想定区域
ifld	内水浸水想定区域

今回の検証では、より都市の3次元表示において重要だと思われる建築物、地形、道路、植生を表示することを目的とした。

4.4 DOM

CityGMLのデータはXML形式で記述されているため、

Javascript 上で扱いやすくするために DOM(Document Object Model)に変換する。DOM はデータを、木構造を持つオブジェクトとして表現でき、オブジェクトのインタフェースを使って各ノードの情報を参照することができる。今回のシステムでは、XML 用のパーサーに CityGML を入力し、DOM に変換したものを利用する。

4.5 建築物の情報解析

CityGML の幾何情報をブラウザ上に表示するために、データ内から各情報を抜き出し、下の図のような、建築物の情報を扱うクラスの属性情報として格納する。

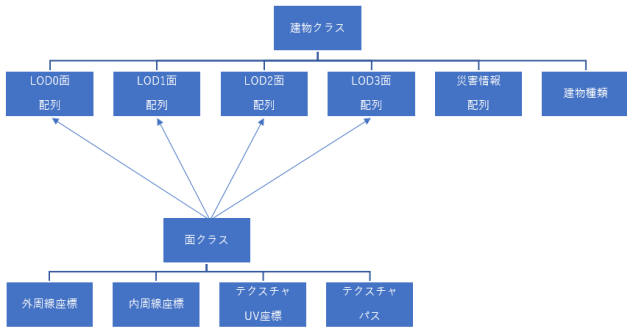


図 6 建造物のクラス

4.5.1 原点の情報

CityGML は対象となる都市の範囲がメッシュ状に分けられており、データをメッシュごとに分けて作成される。そのメッシュの緯度、経度の最低値と最大値が CityGML 内に記述されており、`gml:Envelope` タグ内に格納されている。

3次元空間上に表示する際、表示するモデルをワールド座標系のどこに置くかを定義する必要がある。そこで、データの緯度経度の最大値と最低値を取り、2つの座標の平均座標を求めた後、その座標がワールド座標系の原点に重なるようにモデルの配置を行う。

4.5.2 建造物の頂点座標

各建造物の幾何情報は LOD ごとに異なる名称のタグに格納されている。それぞれの LOD の幾何情報が格納されているタグの名称は以下の表である。

表 2 各 LOD の頂点座標のタグ

LOD	頂点座標が格納されたタグ
0	lod0RoofEdge
1	lod1Solid
2	lod2Geometry
3	lod3Geometry

例として、実際に CityGML 内に記述されている LOD1 の幾何情報の部分を抜粋したものが下図である。

```

exterior>
gml:CompositeSurface>
  gml:surfaceMember>
    gml:Polygon>
      gml:exterior>
        gml:LinearRing>
          gml:posList>33.633330888981945 130.68653619190016 18.753
33.633332152651064 130.68655201337214 18.753
33.633283930888965 130.6865575248402 18.753
33.633282657848795 130.6865415632811 18.753
33.633273645497844 130.68654256713572 18.753
          </gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </gml:surfaceMember>
  gml:surfaceMember>
    gml:Polygon>
      gml:exterior>

```

図 7 LOD1 幾何情報の記述例

建物を構成する面は `gml:surfaceMember` というタグで定義されており、その面を構成するポリゴン(多角形)の情報が `gml:Polygon` タグ内に記述されている。この情報は、建造物が持っている面の数だけ定義されている必要がある。`gml:exterior` は外周線を表し、このタグが `gml:interior` と記載されていた場合、内部には内周線の情報が定義されている。`gml:LinearRing` は多角形の境界を表す輪が定義されており、その直下にある `gml:posList` 内に実際の座標が緯度、経度、標高の順番で羅列してある。この値を取得し、それぞれ建造物オブジェクトのメンバ変数として格納する。

4.5.3 座標変換

Three.js を含めた多くの 3DCG ライブラリでは、幾何形状の表現に直交座標系を用いる。しかし、CityGML は、面の頂点座標などをすべて地理座標系で表現しているため、Three.js で直接扱うことができない。そこで、CityGML から取得した座標をメートル法で直交座標系に変換してから利用した。

4.5.4 属性情報

CityGML 内には幾何情報以外の情報もタグで定義することができる。以下の図は静岡県沼津市の CityGML 内のデータを一部切り取ったものである。

```

<gen:stringAttribute name="建物ID">
  <gen:value>22203-bldg-97752</gen:value>
</gen:stringAttribute>
<gen:genericAttributeSet name="狩野川水系狩野川洪水浸水想定区域(想定最大規模)">
  <gen:stringAttribute name="規模">
    <gen:value>L2</gen:value>
  </gen:stringAttribute>
  <gen:stringAttribute name="浸水ランク">
    <gen:value>3</gen:value>
  </gen:stringAttribute>
  <gen:measureAttribute name="浸水深">
    <gen:value uom="m">3.709</gen:value>
  </gen:measureAttribute>
  <gen:measureAttribute name="継続時間">
    <gen:value uom="hour">9.350</gen:value>
  </gen:measureAttribute>
</gen:genericAttributeSet>

```

図 8 属性情報の記述例

`gen:stringAttribute` タグ内にデータの名称、`gen:value` タグ内にそのデータの実際の値が格納される。また、属性情報のセットを定義することもでき、`gen:stringAttribute` タグの親として定義されている `gen:genericAttributeSet` タグが

それにあたる。この部分は都市ごとに定義されている属性が含まれるため、可変長のリストとしてクラスに格納する必要がある。

また、属性情報には下の図のようにコードリストと呼ばれるファイルのパスが格納されている場合がある。

```
<gen:genericAttributeSet name="津波浸水想定">
  <gen:stringAttribute name="説明">
    <gen:value>静岡県第4次地震被害想定（レベル2津波の重ね合わせ図）</gen:value>
  </gen:stringAttribute>
  <gen:stringAttribute name="浸水ランク">
    <gen:value>3</gen:value>
  </gen:stringAttribute>
  <gen:stringAttribute name="浸水ランクコードリスト">
    <gen:value>../codeLists/Tsunami_rank.xml</gen:value>
  </gen:stringAttribute>
</gen:genericAttributeSet>
```

図 9 コードリストパスの記述例

これは、属性情報の値をコードリストと呼ばれる別の XML ファイル上で定義するためのものである。図の例では、「浸水ランクコードリスト」という名前の属性内部に記述されている「Tsunami_rank.xml」のファイルに浸水ランクが XML で羅列されており、上の「浸水ランク」属性の gml:value タグ内には、3 のコードに対応するデータが実際の値であることを示している。

4.5.5 テクスチャ

LOD2 以上のモデルには、テクスチャが設定されており、CityGML 内の app:appearanceMember タグ内に画像のファイルパスやテクスチャを貼る際の UV 座標、テクスチャを貼る面の ID が記述されている。実際の記述例を下の図に示す。

```
<app:Appearance>
  <app:theme>rgbTexture://app:theme</app:theme>
  <app:surfaceDataMember>
    <app:ParameterizedTexture>
      <app:imageURI>52385608_bldg_6697_appearance/num286759_12.jpg</app:imageURI>
      <app:mimeType>image/jpg</app:mimeType>
      <app:target uri="#poly_NUM286759_p4157_1">
        <app:TexCoordList>
          <app:textureCoordinates ring="#line_NUM286759_p4157_1">0.9968576 0.4237288
            0.9968576 0.9769477
            0.9824154 0.9769477
            0.9824154 0.4237288
            0.9968576 0.4237288
          </app:textureCoordinates>
        </app:TexCoordList>
      </app:target>
    </app:ParameterizedTexture>
  </app:surfaceDataMember>
</app:Appearance>
```

図 10 テクスチャ情報の記述例

app:imageURI タグ内に保存されている画像ファイルのパス、app:target タグ内がテクスチャを貼る面の ID、app:textureCoordinates タグ内がテクスチャの UV 座標を表す。

4.6 道路、植生、地形の情報解析

建造物以外の地物に関しても同様にクラスを作成し、オブジェクトのメンバ変数に各情報を格納した。

4.6.1 道路

主に取得した情報が格納されるタグを以下の表に示す。

表 3 主に使用するタグと内容 (道路)

タグ名	内部の情報
tran:Road	一区画の道路
gml:name	道路の名前
tran:function	道路法及び建築基準法におけ

	る道路の区分 (高速道路, 一般国道など)
tran:usage	利用方法 (緊急輸送道路, 避難道路など)
tran:trafficArea	車や人が通行可能な領域 (車道, 歩道)
tran:auxiliaryTrafficArea	機能補助のための領域 (分離帯, 交通島, 路肩など)
tran:function	道路の区分
tran:lod1MultiSurface	LOD1 の幾何情報
tran:lod2MultiSurface	LOD2 の幾何情報
tran:lod3MultiSurface	LOD3 の幾何情報
gml:posList	頂点座標

道路の情報は、車や人が通行可能な領域と、交通領域の機能を補助する目的で設けられた領域でタグが分けられているため、それぞれを別のオブジェクトで定義する。道路クラスの構造は下の図のようになっている。



図 11 道路のクラス構造

4.6.2 植生

主に取得した情報が格納されるタグを以下の表に示す。

表 4 主に使用するタグと内容 (植生)

タグ名	内部の情報
veg:class	樹木や地被植物, 草花などの区分
veg:SolitaryVegetationObject	独立した樹木
veg:lod3Geometry	利用方法 (緊急輸送道路, 避難道路など)
veg:PlantCover	道路の区分
veg:lod3MultiSurface	LOD3 の幾何情報
gml:posList	頂点座標

植生は、街路樹のような独立した樹木と、植栽のような個々の樹木を識別しない植栽に分けられるため、別々のオブジェクトを定義する。植生クラスは下の図のようになっている。



図 12 植生のクラス構造

4.6.3 地形

主に取得した情報が格納されるタグを以下の表に示す。

表 5 主に使用するタグと内容 (地形)

タグ名	内部の情報
dem:ReliefFeature	一区画の地形
dem:lod	地形の LOD
gml:posList	頂点座標

地形は他の地物と違い、LOD を定義するタグが別で用意されている。

4.7 面の三角分割

CityGML の面は 4 点以上の点を持った多角形の面が多くある。4 点以上の点がある面は、同一平面上に点が無いような「非平面」が定義できてしまう。そのため、多くの 3DCG システムではポリゴンは 3 角形で定義されることが望ましい。そこで、本研究では、幾何情報からポリゴンを定義する際に、多角形を自動的に三角形分割するようにする。

5. 結果と評価

実際に構築したシステムについて、動作結果と以下の観点から、システムの評価を行う。

- WEB ブラウザ上で動作するか
- 形状は描画できるか
- テクスチャが適用できるか
- 属性情報を扱った機能があるか

5.1 WEB ブラウザ

実際にシステムを WEB ブラウザ上で動作させたものが下の画像である。

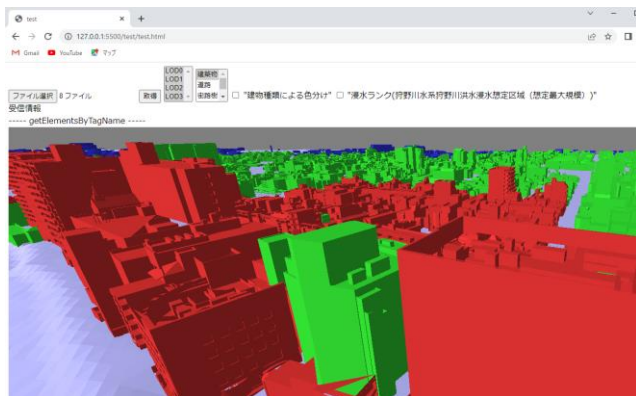


図 13 システムの実行結果

今回構築したシステムは、WEB ブラウザ上で利用が可能であることが確認できた。LOD ごとにモデルを色分けでき、なおかつ地形、道路、植生も描画でき、同時に重ねて表示することも可能になった。

また、以下の表に示す OS、ブラウザで動作することが確認できており、OS やデバイスの変化にも対応できていることが確認できた。

表 6 動作確認済みの OS、ブラウザ

OS	ブラウザ
Windows10	Microsoft Edge
Windows11	Mozilla Firefox
MacOS	Google Chrome
Linux	Safari

5.2 形状の評価

形状の評価では、LOD0 から LOD3 まで検証をするため、建造物が LOD3 まで対応している静岡県沼津市のデータを使用した。本研究手法で表示したものと、PLATEAU VIEW [10] を比較した画像を下の図に示す。

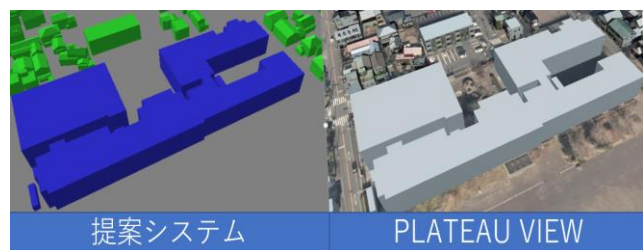


図 14 提案システムと PLATEAU VIEW 比較

以上の結果から、比較的正確に幾何情報を取得できていると考えられる。また、道路や地形についても同様に読み込み描画することができた。しかしながら、穴の空いた面の描画がうまくいかず、下図のように PLATEAU VIEW と比較すると穴が埋まってしまふ建物がいくつか見られた。

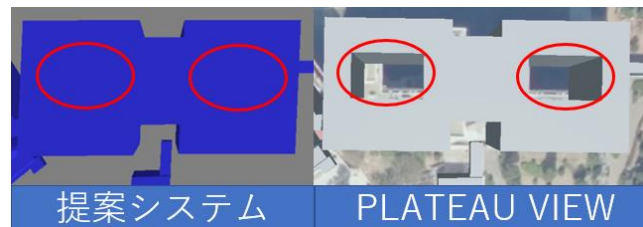


図 15 穴の空いた建物の比較

また、LOD3 モデルにおいて、下の図のようにポリゴンがうまく貼られていない建物がいくつか見られた。穴の空いていない LOD1 および LOD2 は正常に描画できていることから、頂点座標ではなく、頂点を結ぶインデックスの問題であると考察している。

5.3 テクスチャ

今回の提案手法と PLATEAU VIEW を比較したものが下の図である。



図 16 テクスチャの比較

テクスチャの適用に関して問題なく適用ができた。看板などの文字や絵が描かれた複雑な模様についてもある程度描画できた。

属性情報を利用するための仕組みとして、建物種類を使った色分けと、狩野川水系洪水浸水想定区域(想定最大規模)による色分けの二つを実装した。以下の図は、建物種類による色分けをしたものである。

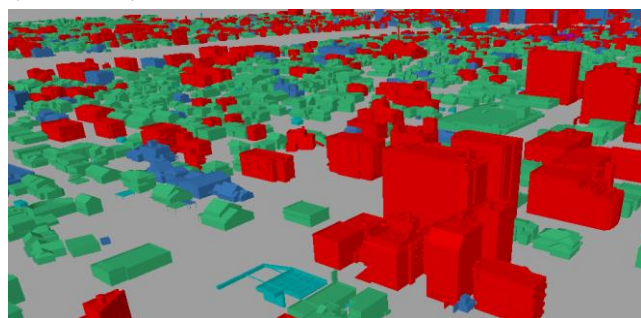


図 17 建物種類による色分け

建物を種類別に色分けすることで、分布が分かりやすくなる。

また、以下の図は狩野川水系洪水浸水想定による色分けを上から俯瞰した図と、沼津河川国道事務所が公開している浸水想定区域 [12]を比較した図である。浸水想定区域のリスクを分かりやすく確認することができる。

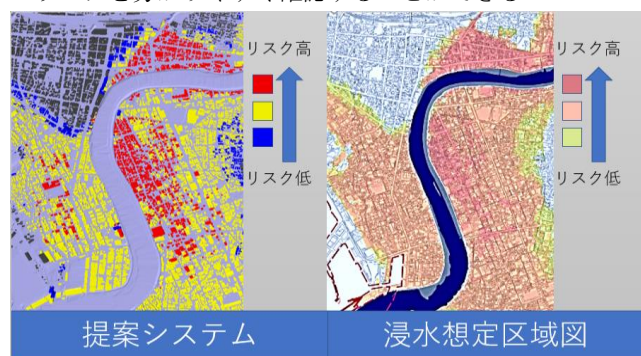


図 18 狩野川水系洪水浸水想定の比較

6. まとめ

本研究では、CityGML を WEB ブラウザ上で、変換をせずに直接入力し、3次元空間上に可視化するシステムを試作・評価した。提案システムによって、CityGML をユー

ザーがデータ変換の手間なく、建造物、道路、植生、地形の幾何情報の可視化をすることができた。テクスチャの適用が可能であることも確認できた。また、建物種類による色分け、河川洪水浸水想定による色分けを実装することによって、CityGML が持つ属性情報を利用することが確認できた。

参考文献

- [1] 国土交通省, “PLATEAU,” : <https://www.mlit.go.jp/plateau/>, (参照 2023-2-17).
- [2] Open Geospatial Consortium, “Open Geospatial Consortium,” : <https://www.ogc.org/>, (参照 2023-2-17).
- [3] Gerhard Gröger, Lutz Plümer, “CityGML - Interoperable semantic 3D city models,” 2012.
- [4] Open Geospatial Consortium, “OpenGIS® City Geography Markup Language (CityGML),” : <https://www.ogc.org/standards/citygml>, (参照 2023-2-17).
- [5] Open Geospatial Consortium, “OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard,” : <https://docs.ogc.org/is/20-010/20-010.html>, (参照 2023-2-17).
- [6] 国土交通省, “3D 都市モデルの導入ガイダンス,” : https://www.mlit.go.jp/plateau/file/libraries/doc/plateau_doc_0000_ver02.pdf, (参照 2023-2-17).
- [7] 国土交通省, “3D 都市モデル標準製品仕様書,” : <https://www.mlit.go.jp/plateaudocument/>, (参照 2023-2-17).
- [8] 稲留浩太郎, “CityGML 形式の三次元グラフィックスデータへの変換に関する研究,” 九州工業大学情報工学部知能情報工学科, 2021.
- [9] Safe Software, “Safe Software | FME | Data Integration Platform,” : <https://www.safe.com/about/>, (参照 2023-2-17).
- [10] 国土交通省, “Plateau View,” : <https://plateauview.mlit.go.jp/>, (参照 2023-2-17).
- [11] 社会基盤情報流通推進協議会, “G 空間情報センター,” : <https://front.geospatial.jp/>, (参照 2023-2-17).
- [12] 沼津河川国道事務所, “浸水想定区域図 狩野川水系狩野川洪水浸水想定(想定最大規模),” : https://www.cbr.mlit.go.jp/numazu/bousai/shinsui/pdf/01_soutei_kano.pdf, (参照 2023-2-17).