

S3M 空間 3D モデルデータ形式概要

2025 年 2 月作成
日本スーパーマップ株式会社
事業統括部

S3M (Spatial 3D Model) データ形式は、空間 3 次元モデルデータの伝送、交換、共有を目的とした、**拡張可能でオープンなデータ形式**です。この形式は、さまざまな端末 (モバイルデバイス、ブラウザ、デスクトップコンピュータ) の地理情報プラットフォームでの 3 次元空間データの保存、効率的な可視化、共有、相互運用を支援することを目的としています。S3M は、多種多様な 3 次元データを統合的に扱い、効率的なデータ管理と利用を促進します。

【概要】

S3M のコンセプト

- データ形式:** S3M は、3 次元空間モデルを表現するためのファイル形式であり、単一のデータタイプに限定されず、複数のデータソースを統合できます。
- オープン性と拡張性:** S3M はオープンな仕様であり、開発者や研究者が自由に利用、拡張できます。
- 相互運用性:** 異なるシステムやプラットフォーム間でのデータの交換や共有を容易にし、データの再利用性を高めます。

S3M が対応する主なデータタイプ

- 従来のモデル:** 人工的にモデリングされた 3 次元モデルデータ
- 実景 3 次元:** 3 次元都市モデルデータと点群データを含む
- BIM (Building Information Modeling) データ:** BIM 設計ソフトウェアで作成された 3 次元モデルデータ
- ベクタデータ:** 2 次元および 3 次元の点、線、面データ、3 次元パイプラインデータを含む

S3M の優位性

- オープン仕様:** S3M は、オープンソースのデータ規格であり、2 次元と 3 次元のデータを統合的に扱うことができます。
- 高いパフォーマンス:** LOD、バッチ描画、インスタンス化などの技術により、大規模な 3 次元シーンを効率的にレンダリングできます。

- **コンパクトなデータ:** データはバイナリ形式で保存され、ファイルサイズが小さく、転送速度が向上します。
- **多様なデータ形式のサポート:** さまざまなデータソース（BIM、点群、都市モデル、人工モデルなど）をサポートしており、異構造データの統合が可能です。
- **マルチ端末対応:** モバイルデバイス、ブラウザ、デスクトップなど、様々なプラットフォームで利用可能です。
- **拡張性:** データ構造がモジュール化されており、新しいデータ型や機能を簡単に追加できます。

S3M の活用例

S3M は、以下のような様々な分野で活用されています:

- **都市計画:** 3次元都市モデルの作成と管理、都市開発シミュレーション、景観解析などに利用されています。
- **建築:** BIM データの可視化、設計レビュー、施工管理、施設管理などに利用されています。
- **交通:** 道路、橋梁、鉄道などのインフラモデルの可視化、交通シミュレーション、運行管理などに利用されています。
- **エネルギー:** 発電所、送電線、パイプラインなどのモデル化、資源管理、リスク管理などに利用されています。
- **防災:** 地形モデル、災害シミュレーション、避難経路解析などに利用されています。
- **デジタルツイン:** 現実世界の物理オブジェクトやプロセスを仮想空間で再現し、可視化、解析、最適化、シミュレーションを行うために利用されています。

S3M と SuperMap GIS の関係

S3M は、**SuperMap GIS** プラットフォームとの連携も可能です。

- SuperMap GIS は、BIM データや点群データなどの様々なデータを S3M 形式に変換する機能を提供します。
- SuperMap GIS の高性能な 3D レンダリングエンジンを使用して、S3M データを高品質に可視化できます。
- SuperMap iClient3D for WebGL, WebGPU, Cesium などのクライアントで利用可能です。
- SuperMap Hi-Fi 3D SDK を使ってゲームエンジンで S3M を利用できます。

SuperMap GIS による S3M のデータ変換

S3M は、様々なデータ形式を変換して利用できます。

- BIM データ（IFC, RVT, DWG など）を S3M に変換する自動化されたワークフローが提供されています。
- 3D 都市モデルデータから S3M への変換も可能で、大規模な都市モデルや地形モデルを高速に可視化できます。

SuperMap GIS による S3M のパフォーマンス

S3M 形式は、大規模な 3 次元シーンのレンダリングにおいて、優れたパフォーマンスを発揮します。

- Cesium などのプラットフォームと比較して、S3M は、より安定したフレームレートとより高速なロード時間を実現できます。
- 大規模な 3D 都市モデルデータや BIM データのリアルタイムレンダリングが可能です。
- 大量のデータを効率的に処理し、スムーズな視覚体験を提供します。

SuperMap GIS による S3M の解析機能

S3M データは、様々な空間解析に使用できます。

- **可視域解析:** 特定の場所から見える範囲を解析します。
- **スカイライン解析:** 建物や地形の輪郭を解析します。
- **見通し解析:** 特定の地点間の視界を解析します。
- **断面解析:** モデルを任意の断面で切り出して内部構造を観察します。
- **衝突解析:** 複数のオブジェクト間の干渉を検出します。
- **開放度解析:** ある場所の開放度を解析します。
- **浸水解析:** 浸水する範囲をシミュレーションします。
- **重畳解析:** 複数のデータを重ね合わせて解析します。
- **建築物高さ制限解析:** 建物の高さ制限を解析します。
- **その他:** 切り盛り、傾斜角度・傾斜方向、等高線解析。

【仕様】

ファイル構成

S3M データ形式は、以下の主要なファイルタイプで構成されています。

- **記述ファイル (.scp):**
 - データセット全体に関するメタデータを JSON 形式で記述します。
 - TileTree のルートノードのパス、データの空間範囲 (geoBounds)、高さ範囲 (heightRange)、座標系情報 (crs)、LOD タイプ、データタイプなどの基本情報を含みます。
 - **TileTreeSetInfo** オブジェクトとして表現され、データの基本情報、バージョン、データタイプ、空間分割タイプ、LOD 細分化タイプ、地理的範囲、高さ範囲、座標系情報、TileTree 情報、ユーザー拡張情報などを定義します。
 - ファイル名は任意ですが、拡張子は必ず".scp"とする必要があります。
- **データファイル (.s3mb):**
 - タイルごとの実際の 3 次元モデルデータを格納します。
 - 各タイルは、**.s3mb ファイル**として保存され、**PatchLOD** のデータに対応します。
 - バイナリ形式でデータが格納され、データ構造には、ヘッダー、予約領域、Shell、ModelEntities が含まれます。
 - ファイル名は任意ですが、拡張子は必ず".s3mb"とする必要があります。
- **インデックツリーファイル (.json):**
 - TileTree の各 LOD (Level of Detail) 層のタイルファイル情報を JSON 形式で記述します。
 - タイルのバウンディングボックス、LOD 切り替え情報、子ノードファイルパスなどの情報を含みます。
 - **データファイルの検索効率を向上**させるために、実際のデータをロードせずに、必要な情報にアクセスできるようにします。
 - ファイル名は TileTree のルートノード名と同じで、拡張子は必ず".json"とする必要があります。
 - タイルの名称、タイルの情報 (lodNum、modelPath、rangeMode、rangeValue、boundingBox)、ステータス (lodCount、tilesCount) などが含まれます。
- **属性ファイル:**
 - 属性記述ファイル (attribute.json):
 - 各データセットの属性に関するメタデータ (レイヤー名、ID 範囲、フィールド情報など) を記述します。
 - ファイル名は "attribute.json" に固定され、記述ファイル(.scp)と同じディレクトリに保存されます。
 - 属性データファイル (.s3md):

- TileTree 内のすべてのオブジェクトの属性データを格納します。
- JSON 形式で記述され、zip 圧縮されています。
- ファイル名は TileTree のルートノード名と同じで、拡張子は必ず".s3md"とする必要があります。
- 各レイヤーの属性データ (ID 範囲、フィールド情報、レコード) を含みます。

データ構造

S3M データ形式は、**TileTree** という階層構造を用いて 3 次元空間データを整理します。

- **TileTree:**
 - 単一のルートノードから始まり、複数の LOD レベルで構成されるツリー状の空間データ構造です。
 - 各ノードは**タイル**を表し、それぞれの空間範囲と詳細度を持ちます。
 - ルートノードの空間範囲は、すべての子ノードの空間範囲の和集合です。
- **タイル:**
 - 2 次元の矩形または 3 次元の直方体の空間に対応します。
 - 隣接するタイル間では空間範囲が重複することがあります。
 - 各タイルは、より詳細なデータを表現するために、さらに小さな単位に分割されることがあります。
- **PatchLOD:**
 - TileTree 内の特定の LOD 層のデータ分割を表します。
 - 1 つまたは複数の Patch を含みます。
- **Patch:**
 - PatchLOD のデータ分割で、親 Patch と子 Patch のツリー構造を形成します。
 - **親 Patch の空間範囲**は、すべての子 Patch の空間範囲の和集合です。
 - 各 Patch は 0 個または 1 個の親 Patch、0 個または複数の子 Patch を持つことができます。
- **Geode:**
 - Patch に対応するデータパッケージを表します。
 - スケルトン、マテリアル、テクスチャの 3 つのエンティティオブジェクトを含みます。
- **ModelEntity:**
 - Geode の基本データ構成要素で、スケルトン、マテリアル、テクスチャの 3 つのエンティティタイプを含みます。

データタイプ

- **スケルトン (Skeleton):**
 - 頂点、頂点インデックス、テクスチャ座標などの幾何学的データを格納します。
 - 頂点データパッケージ (VertexDataPackage) と 1 つ以上の頂点インデックスパッケ

ージ (IndexPackage) で構成されます。

- 頂点データパッケージは、頂点の座標、法線、色、テクスチャ座標などを記述します。
- 頂点インデックスパッケージは、骨格構造の構成を記述し、各頂点のレンダリング方法を識別します。

- **マテリアル (Material):**

- **モデルオブジェクトの表面の視覚的属性の集合**で、色彩、テクスチャ、滑らかさ、透明度、反射率、屈折率、発光度などを含みます。
- Pass で構成され、各 Pass はマテリアルが使用するテクスチャオブジェクトの名前を記録します。
- Pass には、環境光、拡散光、鏡面光の色、光沢度、透明度ソートの有無、テクスチャ情報などが含まれます。
- テクスチャ情報には、テクスチャ名、URL、テクスチャ座標の U/V/W 方向のアドレスモード、フィルタリングオプション、テクスチャ行列などが含まれます。

- **テクスチャ (Texture):**

- テクスチャマッピング情報を格納します。
- 幅、高さ、圧縮方式、テクスチャバイナリデータなどを含みます。
- mipmap レベルとテクスチャデータ (TextureData) を含みます。
- テクスチャデータには、横方向と縦方向のピクセル数、圧縮タイプ、データサイズ、ピクセル形式、バイナリデータが含まれます。

.s3mb ファイルの詳細

.s3mb ファイルは、3 次元モデルデータを格納する主要なファイルです。ファイル構造は以下のようになっています。

- **ヘッダー:**

- s3mb ファイルのバージョン番号
- 解凍後のパッケージのバイト数
- 圧縮されたパッケージのバイト数

- **圧縮されたパッケージ (zippedPackage):**

- **Reserved:** 予約済みの 4 バイト
- Shell: PatchLOD、Patch、Geode オブジェクトを格納
 - **streamSize:** シェルのバイナリストリームのバイト数
 - **patchCount:** Patch オブジェクトの数
 - **patches:** Patch オブジェクトの配列
 - **lodFactor:** LOD 切り替え因子
 - **rMode:** LOD 切り替えモード
 - Distance_From_EyePoint: 視点からの距離に基づく切り替え

- **Pixel_Size_OnScreen**: 画面上のピクセルサイズに基づき切り替え
 - **boundingSphere**: バウンディング球
 - **strChildTile**: 子ファイルの相対パス
 - **geodeCount**: Geode オブジェクトの数
 - **geodes**: Geode オブジェクトの配列
 - **matrix**: スケルトンの姿勢に適用される 4x4 行列
 - **skeletonCount**: スケルトンオブジェクトの数
 - **skeletonNames**: スケルトン名の配列
- **ModelEntities**: スケルトン、テクスチャ、マテリアルなどの実体データを格納
 - **skeletonStreamSize**: スケルトンデータストリームのバイナリ長
 - **skeletonCount**: スケルトンオブジェクトの数
 - **skeletons**: スケルトンオブジェクトの配列
 - **name**: スケルトン名
 - **dataPack**: 頂点データパッケージ
 - **indexpackCount**: 頂点インデックスパッケージの数
 - **indexPacks**: 頂点インデックスパッケージの配列
- **textureStreamSize**: テクスチャデータストリームのバイナリ長
- **textureCount**: テクスチャオブジェクトの数
- **textures**: テクスチャオブジェクトの配列
 - **strName**: テクスチャ名
 - **mipMapLevel**: mipmap レベル
 - **texData**: テクスチャデータ
- **materials**: マテリアルデータ (JSON 形式の文字列)

基本データ型と構造

S3M は、様々な基本データ型を利用してデータを表現します。

- **数値型**:
 - **byte**: 1 バイトの符号なし整数
 - **bool**: 1 バイトのブール値 (0 または 1)
 - **int16**: 2 バイトの符号付き短整数
 - **uint16**: 2 バイトの符号なし短整数
 - **int32**: 4 バイトの符号付き整数
 - **uint32**: 4 バイトの符号なし整数
 - **int64**: 8 バイトの符号付き長整数
 - **uint64**: 8 バイトの符号なし長整数
 - **float**: 4 バイトの単精度浮動小数点数

- **double**: 8 バイトの倍精度浮動小数点数
- **wchar**: 2 バイトのワイド文字
- **文字列型**:
 - **String** オブジェクトで記述され、**UTF8 エンコーディング**を使用します。
 - 長さ（バイト数）とデータ内容を含みます。
- **JSON 形式**:
 - UTF8 エンコーディングを使用し、BOM ヘッドは含みません。

UML 表記

S3M データ形式の構造は、UML (Unified Modeling Language) の静的構造図を用いて記述されます。UML 図における記号の意味は以下の通りです。

- **双方向関連**: 2 つのクラス間の一般的な関係
- **単方向関連**: あるクラスが別のクラスの属性とメソッドを知っている関係
- **集約**: あるオブジェクトが別のオブジェクトを所有しているが、それらのライフサイクルは独立している関係
- **コンポジション**: あるオブジェクトが別のオブジェクトを所有し、それらのライフサイクルが一致する関係
- **汎化**: あるオブジェクトが別のオブジェクトを継承する関係
- **依存**: あるクラスが別のクラスに依存し、依存されるクラスの変更が依存するクラスに影響を与える関係
- **多重度**: UML 図における多重度の記述は、関係する要素の数を表します。
 - **0..1**: 0 個または 1 個
 - **1**: 1 個のみ
 - **0..n**: 0 個以上
 - **1..n**: 1 個以上