触れる GIS のための３次元地理情報プラットフォームの開発
－Three.js と AR マーカーを用いた Web ベース実装－

石黒 希海*・中村 暢佑・中山 俊*****

Development of a 3D Geospatial Information Platform for Tangible GIS
- A Web-based Implementation with Three.js and AR Markers -

Nozomi ISHIGURO*, Yosuke NAKAMURA**, Shun NAKAYAMA***

Abstract: Traditional GIS requires specialized knowledge and offers limited intuitiveness for public participation in urban planning. We developed a web-based tangible GIS enabling direct manipulation of physical models to control 3D representations. ArUco markers on building models are detected by Python/OpenCV and streamed via WebSocket to a Three.js browser interface. We evaluated the platform in an evacuation-planning case study: participants manipulated models and immediately observed changes in routes and shelter capacities. Results showed real-time synchronization with ~0.2-s latency between physical operations and visualization, enabling non-experts to understand spatial relationships and evacuation scenarios. Limitations include performance on large-scale models and marker-recognition instability under poor lighting. Findings indicate a simple, generalizable foundation for tangible GIS; future work will pursue LOD-based performance and hybrid sensing (computer vision and RFID) to scale collaborative urban-planning workflows.

Keywords: 触れる GIS（tangible GIS）, Three.js, WebGL, AR マーカー（AR markers）, 市民参加（public participation）

１．はじめに

近年，都市計画や防災分野において，住民参加型の意思決定支援を目的としたインタラクティブなツールの研究が進んでいる．その中でも「触れる GIS（Tangible GIS）」は，物理的な模型を操作対象とすることで，専門知識を持たない参加者にも直感的な理解や体験を提供できる仕組みとして注目されている．模型の操作に応じて映像や情報を重ね合わせることにより，GIS を画面上で操作するだけでは得にくい「納得感」や「共感」を伴った議論を可能にする点が大きな特徴である．

触れる GIS の開発の多くは，既存の WebGIS に使用される技術やサービスを転用したものである．そのほとんどは 2 次元にしか対応しておらず，せっかく物理的な模型が 3 次元で作成されていることの強みを活かせているとは言い難い．ウェブブラウザ上で動くウェブアプリケーションは，ネイティブアプリケーションに比べて，高速で高画質なグラフィックス処理を苦手とすることがその一因である．しかし近年では，ウェブブラウザ上でも高速に動作させるために，GPU を使ったグラフィック描画を可能にする WebGL の技術が開発されている．主流のウェブブラウザには軒並み WebGL が導入されている．

現代都市とその機能は立体的に構成されている．複雑な 3 次元空間における人々の活動を適切に分析・表現することは時代の要請とも言える．そのために，触れる GIS のシステム基盤も，3 次元も対応することが求められる．そこで本研究では，WebGL に基づくオープンソースのライブラリである Three.js に着目し，この要素技術を活用して触れる GIS のシステム基盤を開発する．

２．システム基盤の開発

２．１　インプット側の要件

本システムは，従来の GIS が 2 次元画面上での操作に留まっていた限界を超え，物理的な模型操作を通じて 3 次元都市空間を直感的に操作できる環境を提供する．これを実現するため，本研究では Three.js をコアエンジンとした Web ベースの触れる GIS プラットフォームを構築した．システム全体は，マーカー認識モジュール，Three.js 描画エンジン，空間解析モジュール，リアルタイム通信層の 4 つの主要コンポーネントから構成される．

第一のモジュールは物理模型に貼付された正方形パターンの二次元マーカー（ArUco）をカメラで撮影し，コンピュータが自動的に読み取るマーカー認識機能である．従来の GIS における「属性データの入力」に相当する．マーカー認識は認識 PC 上の Python と OpenCV（cv2.aruco）で実行し，検出した ID（必要に応じて姿勢）を WebSocket でブラウザへ送信する．ブラウザ側は追加のネイティブ

*　学生会員　専修大学ネットワーク情報学部 (Senshu University)
〒214-8580　神奈川県川崎市多摩区東三田 2-1-1　Tel：044-712-7772 または E-mail：ne231163@senshu-u.jp
**　学生会員　専修大学ネットワーク情報学部 (Senshu University)
***　正会員　専修大学ネットワーク情報学部 (Senshu University)

アプリを必要とせず，受信した JSON（軽量なデータ交換形式）を Three.js のシーンに反映する．実測では，マーカー検出完了までの遅延は約 0.2 秒（端末構成・照明条件に依存）であった．

第二のモジュールは，マーカー認識で得られた情報を即座に 3 次元表示システムに伝達するリアルタイム通信層である．従来の GIS では「データ更新→再描画」に数秒を要することが多いが，本システムは WebSocket による常時接続の双方向通信を採用し，認識 PC からの検出イベントを JSON でプッシュ配信する．ポーリングを回避できるため，検出後の更新は次フレームで画面に反映される．その結果，参加者が模型を動かした瞬間に 3D 表示が追従し，物理操作とデジタル表示を一体化できる．

第三のモジュールは，Web ブラウザ上で 3 次元グラフィックスを高速描画するための Three.js 描画エンジンである． 従来の WebGIS は主に 2 次元の地図表示が中心であったが，Three.js は WebGL を活用してブラウザ上で高品質な 3 次元表現を可能にする．本研究では，地形は平面ジオメトリを基盤とし，建物は主に BoxGeometry を用いて属性値に応じた押し出し高さを付与して配置し，都市空間を構成する．また，GIS の「レイヤ管理」に相当する構造はシーングラフで実現し，地形・建物・道路・経路・注記などを機能別の Group／オブジェクトとして分離して保持することで，表示・非表示やスタイル（色，不透明度）をレイヤ単位で即時に切り替えられる．必要に応じて複雑形状（glTF〔GL Transmission Format，3D モデル交換形式〕等）にも拡張可能である．

第四のモジュールは，3 次元都市空間上での経路探索を実行する機能である．建物配置の変更に応じて，建物外縁とその間隙を通行可能領域として離散化し，交点からなる道路グラフを自動更新する．出発地と目的地は最寄りノードへスナップし， A*（ユークリッド距離ヒューリスティック）で最短経路を計算する． 復元した経路は 3D ポリラインとして即時に可視化され，参加者は経路の立体的関係を直感的に理解できる．

２．２　AR マーカーを利用したインプット

本システムのインプット手法は,2.1 で述べたように，物理模型の操作をデジタル空間に反映させるための AR マーカー認識技術を基盤とする．各建物模型には固有の ID を持つ AR マーカーを貼付し，カメラ画像から位置と向きを推定し，種別は ID と属性マスタを照合して取得する．マーカーは ArUco ライブラリの正方形パターンを採用し，辞書に応じて最大約 1000 の ID を利用できる．マーカーサイズは模型に合わせて調整可能で，本研究では一辺 3 センチメートルを標準とした．これは貼付作業が容易である一方，約 1 メートルの撮影距離でも安定して認識できる目安のサイズである．

カメラによるマーカー認識処理は，一般的な Web カメラから取得した映像を用い,認識 PC 上の Python と OpenCV の cv2.aruco で実行する．映像は連続フレームとして取り込み，各フレームごとにマーカーの検出と必要に応じて姿勢推定を行う．検出では，まず正方形候補を抽出し，内部パターンを照合してマーカーID を特定する．さらに，四隅の画素座標と事前に求めたカメラキャリブレーション結果を用いて，3 次元の位置と向きを推定し，レンズ歪みを補正した上で後段へ渡す JSON を生成する．

認識されたマーカー情報は， ブラウザ側の属性マスタと照合される．例えば ID 42 のマーカーが検出されると，当該 ID に紐づく「住宅建物，人口 150 人，3 階」などの属性が即時に引き当てられる．属性は設定 UI または JSON で編集可能で，同一の物理模型をシナリオ切替で再利用できる．照合結果は次フレームの描画と評価に反映され，建物の色や高さ，収容力の算出，経路の再計算が更新される．

２．３　３次元情報のアウトプット

触れる GIS はインターフェースの特殊性が取り上げられがちだが，情報のアウトプットまで含めて成立する．どう情報を伝達するかは触れる GIS の共通課題であり，方式は大きく(1)模型上への直接投影と(2)モニター表示に分けられる．前者は暗室環境や厳密な投影調整を前提とするため，公開討論やワークショップのような開かれた場への適用には制約がある(Yan et al.， 2021)．対して後者は， 市民参加を促進するプラットフォームとして成果を示している（Noyman et al.，2017）．一方で，物理模型と 2 次元画面の往復は表現の不一致を生み，理解の連続性が途切れやすいと指摘される（Noyman et al.，2018）．そこで本研究では，モニター上に物理模型を再現し，評価結果を可視化することで理解を支援する．Three.js のシーングラフ（Group）で地形・建物・道路・経路・注記をレイヤ分離し，状態切替に応じて可視／不可視やスタイル（色・不透明度）を即時更新する実装とした．図 2 に，ブラウザ側の可視化例を示す．建物は属性に応じて押し出し,歩行可能領域から生成したノード・エッジおよび最短経路を重ね合わせている．Three.js による出力例を図 1 に示す．

３．ケーススタディ

３．１　シナリオの設定

本研究ではケーススタディとして,「新しいマンションの建設に伴う避難経路および避難場所の収容力の検証」を題材とした．都市部では再開発や人口増加により建物の配置が変化し，災害時の避難経路や避難所のキャパシ

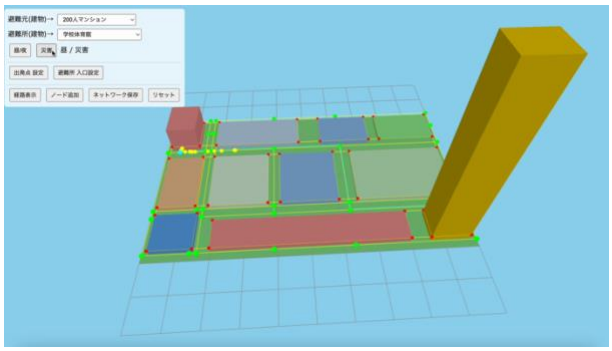


図 1　Three. js による出力例

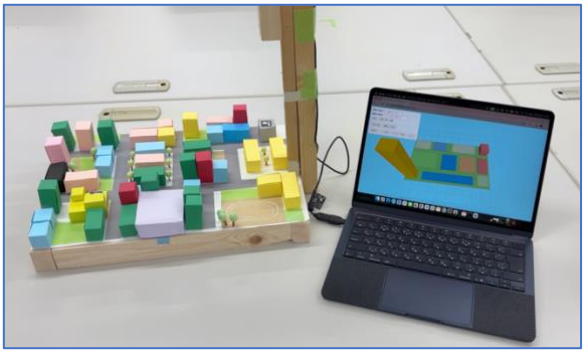


図 2　模型と画面の連動

ティに影響が生じることがある．従来の 2D 地図上での議論では直感的な理解が得にくく，参加者間での認識のずれも生じやすい．そこで，触れる GIS を用いて建物の追加や配置変更を物理模型上で行い，その影響を 3D で直感的に確認するシナリオを設定した．

３．２　実装手順

実装は次の流れで構成した.まず建物パーツに AR マーカーを貼付して物理模型上に配置する．ついでカメラでマーカーを認識し，ID と座標（必要に応じて姿勢）を取得する．得られた情報を Three.js に渡して地形モデル上に対応する建物を生成し，人口属性を高さに写像して押し出し（extrusion）により立体化する．さらに避難経路を再計算し，避難所キャパシティの変化とあわせて画面に即時表示する．

モニター側は建物表現では，AR マーカーにより認識された建物を立方体や角柱として可視化し，人口や収容能力などの属性データを建物の高さや色に反映させる．これにより，数値データを視覚的に把握しやすくしている．道路網の表現は，建物配置に基づいて自動生成される歩行可能領域を白線で表示する．建物が移動や追加された場合，道路網も即座に再計算されて表示が更新される．経路表現では，出発地から目的地への最短経路を赤線で表示し，複数の経路候補がある場合は色分けして同時表示する．経路上には距離や所要時間の情報も文字で表示される．群集の動きは，経路に沿って移動する小さな点やアイコンで表現される．避難シミュレーションの場合，避難者の流れが視覚的に理解できるよう，点の移動速度や密度を調整している．

３．３　実装結果

実装の結果，物理模型に新しい建物を配置すると，Three.js 上の都市モデルに即時に建物が追加されることを確認した．さらに，建物の高さが人口属性に基づき extrusion 処理で反映され，収容規模の大小が視覚的に表現された．その結果，避難経路の混雑や避難所の収容人数の変化を画面上で直感的に把握することが可能となり，非専門家を含む参加者にとっても避難シナリオを理解しやすい環境が実現できた． 図 2 に，物理模型と Web ブラウザ画面の連動構成を示す．

４．考察

４．１　Three. js の可能性と限界

本研究で用いた Three. js は，WebGL を基盤とする JavaScript ライブラリであり，Web ブラウザ上で高度な 3D 表現を実現できる．この特性により，触れる GIS に必要とされる「直感的かつ視覚的に理解しやすい空間表現」を比較的容易に実装できるという利点がある．具体的には，地形や建物を動的に生成・変形できる柔軟性があり，ユーザーが操作した結果を即座に視覚化できる．また，Web 技術であることから専用アプリケーションを導入する必要がなく，多様な端末で利用できるアクセシビリティの高さも大きな強みである．さらに，オープンソースとして拡張機能が豊富であり，外部データとの統合も容易であるため，将来的には交通シミュレーションや環境データの重畳表示などへの発展も期待できる．

加えて，GIS のコア機能であるレイヤの重ね合わせについては，Three. js のシーングラフ（Group）とマテリアル制御によりレイヤを分離できる．また，可視／不可視の切替や表示順序，色・不透明度などのスタイル変更をレイヤ単位で行うことで実現できる．本研究でも状態切替に応じてレイヤのスタイルを即時に割当てし直し，議論の焦点に応じた重ね合わせを運用した．なお，半透明レイヤの重畳では深度テスト／描画順に依存するため，レンダリング順やマテリアル設定の調整が必要となる．

一方で，Three. js には限界も存在する．都市全体のような大規模 3D モデルでは描画負荷が高く，リアルタイム性が損なわれやすい．このため LOD（Level of Detail，表示距離等に応じて詳細度を切替える手法）の導入やデータ軽量化が不可欠である（Luebke et al.， 2002；Losasso・Hoppe，2004）．また，Three. js は GIS 専用ライブラリではないため，空間解析機能や地理的正確性の面で制約がある．したがって，本格的な空間解析を行う際は既存の GIS 基盤との連携を前提とする．

本研究のケーススタディにおいても，人口属性を用いた建物の立体化は容易に実現できた一方で，建物数の増加に伴う描画負荷が課題となる可能性が示唆された．したがって，Three. js は「触れる GIS」を構築する上で直感的な可視化基盤として有効であるが，都市スケールへの適用を視野に入れる場合には，描画最適化や他システムとの統合を含めた設計が求められることが明らかとなった．

４．２　AR マーカーの課題

本研究では，物理模型の識別および位置姿勢の推定のために AR マーカー（Garrido-Jurado et al., 2014）を用いた. AR マーカーは印刷物を貼付するだけで導入可能であり, オブジェクトごとに固有の ID を割り当てられる点で利便性が高い．さらに，カメラを介して位置と姿勢を推定できるため，模型の移動や回転を即座に 3D 空間に反映でき，触れる GIS の基本要件である「直感的な操作と即時的な可視化」を実現できた．

一方で, AR マーカーにはいくつかの制約も確認された. 第一に，照明条件やカメラ角度によって認識精度が変動し，特に反射や影の影響で一時的にマーカーが検出されない場合があった．第二に，模型パーツが増えるとマーカー同士が互いに隠れ合う「オクルージョン」が発生し，認識が不安定になる傾向が見られた．第三に，マーカーのサイズを小さくすると模型への貼付は容易になるが，認識の安定性が低下することが確認された．これらの問題は既存研究でも指摘されており, AR マーカーは低コストかつ汎用性に優れる一方で，複雑な模型環境や長期的な運用には限界があることが示唆されている（Garrido-Jurado et al.， 2014）.

以上より，AR マーカーは触れる GIS の実装において導入のしやすさと即時性に優れるが，実環境に近い利用を考えると認識の安定性やスケーラビリティに課題を残すことが明らかとなった．今後は，カメラベースの物体認識や RFID など他技術とのハイブリッド化により，認識の堅牢性を高める方向性が有効であると考えられる．

５．おわりに

本研究では，Three.js と AR マーカーを組み合わせることで，物理模型の操作を Web ブラウザ上の 3D 都市モデルに即時的に反映する触れる GIS のプロトタイプを構築した．ケーススタディとして新規マンション建設に伴う避難シナリオを設定し，建物配置や人口属性を反映した立体化により，避難経路の混雑や避難所収容力の変化を直感的に把握できることを確認した．

その結果, Three.js は触れる GIS における直感的な可視化基盤として有効であること，また AR マーカーは導入の容易さと即時性に優れることが示された．一方で，大規模モデルの描画負荷や，照明条件・遮蔽による認識の不安定さといった課題も明らかとなった．

以上より, 本研究は触れる GIS における汎用的かつ簡易な基盤構築の可能性を提示したといえる．今後は，認識技術の多様化や描画最適化に加え, RFID や画像認識など他方式とのハイブリッド化を進めることで，より堅牢かつスケーラブルな都市計画支援システムへ発展させていくことが期待される．

謝辞

本研究を進めるにあたり，専修大学中山研究室の皆様には多大なるご協力を賜った．ここに記して深く感謝の意を表する．

参考文献

Garrido-Jurado，S.，Muñoz-Salinas，R.，Madrid-Cuevas，F. J.，＆ Medina-Carnicer, R.（2014）Automatic generation and detection of highly reliable fiducial markers under occlusion．Pattern Recognition，47（6），2280-2292.

Losasso，F.，＆ Hoppe，H.（2004）Geometry clipmaps：Terrain rendering using nested regular grids． ACM Transactions on Graphics，23（3），769-776.

Luebke，D.，Reddy，M.，Cohen，J. D.，Varshney，A.，Watson，B.，＆ Huebner，R.（2002）Level of Detail for 3D Graphics．Morgan Kaufmann.

Noyman，A.，Holtz，T.，Kröger，J.，Noennig，J. R.，＆ Larson，K.（2017）Finding Places：HCI platform for public participation in refugees’ accommodation process. Procedia Computer Science，112，2463-2472.

Noyman，A.，Sakai，Y.，＆ Larson，K.（2018）CityScope：A data-driven interactive simulation for urban design．『Proceedings of the 9th International Conference on Complex Systems（ICCS 2018）』，253-261．Springer.

Yan，W.，Murakami，Y.，Yasuda，A.，Makihara，T.，Fujimoto，R.，＆ Nakayama，S.（2021）Developing the eXtended Tangible User Interface as an experimental platform for Geo-CPS.『Emerging Technologies for Disaster Resilience : Practical Cases and Theories』, 11-32. Springer, Singapore.