

クライアントサイド描画手法を利用した地図と Google ストリートビューを融合した WebGIS

篠田 陽平[†] 井上 潮[‡]

[†] 東京電機大学大学院工学研究科 〒101-8457 東京都千代田区神田錦町 2-2

E-mail: [†] 09kmc20@ms.dendai.ac.jp, [‡] inoue@c.dendai.ac.jp

あらまし 近年、高速なインターネット環境や高性能で安価な PC の普及を背景として WebGIS の高度化が進んでいる。中でも Google マップのストリートビューは、従来の上空から見た視点を人間の目の高さの視点に変えたという点で直観性が飛躍的に向上しており、利用者も多い。しかし、表示範囲の広域性や位置座標の正確性においては従来の地図よりも大幅に劣っている。本稿では、描画の自由度が高いクライアントサイド描画手法を利用した地図と、視覚情報として人間が認識しやすいストリートビューを融合して、広域的な情報と局所的な情報を一つの画面に連動して表示可能な WebGIS について述べる。

キーワード WebGIS, 地理情報

A WebGIS combining Client-Side-Drawing Maps and Google Street View

Yohei SHINODA[†] Ushio INOUE[‡]

[†] Graduate School of Engineering, Tokyo Denki University

2-2 Kanda-nishikicho, Chiyoda-ku, Tokyo, 101-8457 Japan

E-mail: [†] 09kmc20@ms.dendai.ac.jp, [‡] inoue@c.dendai.ac.jp

1. 序論

近年、FTTH (Fiber To The Home) に代表される大容量で高速なインターネット環境が普及し、同時に半導体技術の向上により高性能で安価な PC が普及している。それに伴い、画像や動画コンテンツなどの通信データ量が大きく、JavaScript, Flash などのクライアントでの処理量が大きい Web ページが増加している。

WebGIS(Geographic Information System on the Web)もそのような Web ページを提供するサービスの一つであり、Yahoo! Japan 地図¹や Google マップ²をはじめとして多くのものが利用されている。我々も、高性能で安価な PC の特性を生かした自由度の高い地図表示を可能とするクライアントサイド描画手法を用いた CMap[1]の開発を進めてきた。

WebGIS が普及した理由として、インターネット環境と PC さえあれば誰でも無料で利用できるサービスが出現したことと、紙媒体の地図では実現できず、コンピュータならではの付加価値を持ったサービスが出現したことが挙げられる。Google ストリートビュー³はそのようなサービスの中の一つであり、従来の上空から見た視点を人間の目の高さの視点に変えたという点

で、直感性を画期的に向上させた。

しかし、Google ストリートビューは、構造物の陰になっている部分が見えない、空間全体での現在位置が分からない、という従来の地図表示にはなかった新たな問題が発生している。

2. 目的

本研究の目的は、直感性の高い Google ストリートビューで広域の地理情報を取得可能にすることである。これを実現することにより、直感性は高いが広域の地理情報を持たない Google ストリートビューで、全体の地理的な位置関係を把握しやすくする。具体的には Google ストリートビューをベースとして表示し、その上にクライアントサイド描画手法で作成した従来の地図表示を重ねる方法により、目的の達成を図る。

3. 既存研究

3.1. Emma

Emma[2]は、動的に地図表示を変えることが可能な WebGIS である。ある範囲全体と、拡大表示される複数の注目地域とを一画面に表示することにより、地図移動の操作量削減と注目地域間の位置関係の把握が容易な地図の生成が可能である。Emma は地図データとして画像データとベクトルデータを併用している。そのため、2 つのデータ間の境界付近で画面表示が乱れる問題がある。

¹ <http://map.yahoo.co.jp/>

² <http://maps.google.co.jp/>

³ <http://maps.google.com/intl/ja/help/maps/streetview/>

3.2. fMap

fMap[3]は、視認性の向上を目的としてフィッシュアイモデルを用いたクライアントサイド描画方式の WebGIS である。フィッシュアイモデルを用いることにより、利用者が注視している中心付近の地図情報を拡大表示し、注視していない周辺の地図情報を収縮させ、なおかつ広い領域の表示を可能とした。作成したシステムで本学付近を表示させた例を図 1 に示す。このシステムは CMap をベースとしており、利用者の画面操作に応じて動的に表示を変化させることができる。

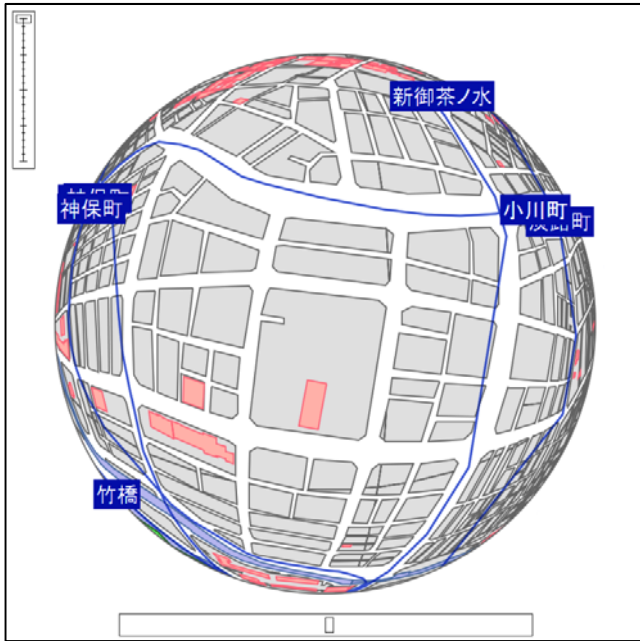


図 1 fMap の表示画面

3.3. Google ストリートビューに関連する研究

本稿で利用する Google ストリートビューに関しては既に様々な研究がなされている。ここではそれらの一部を紹介する。

小林ら[4]は、ユーザによる WebGIS と Google ストリートビューの操作から意図を推定してユーザが必要としているであろう Google ストリートビューの表示を自動生成するシステムを提案している。このシステムを利用することにより、ユーザは WebGIS と Google ストリートビューの操作から目的物の Google ストリートビューの表示を得ることができ、煩雑な Google ストリートビューの操作を軽減させることができる。

また、西尾ら[5]はインターネット上の情報や、スマートフォンなどの端末から投稿される実世界の情報を Google ストリートビュー上に表示させ、それをユーザ間で共有する CMS の構築をしている。Google ストリートビューを CMS の基盤にすることで、ユーザは従来の地図上で情報を共有するよりも直感的に目的地の状況を認識することが可能となる。

4. ベースとなる研究

CMap は、動的な地図表示の描き換えによる自由度の向上、および、サーバ-クライアント間の通信量削減による高速化の 2 点を目指した WebGIS である。本稿の提案手法を実装するためのベースシステムとして CMap を利用するため、その概要を説明する。

多くの既存 WebGIS はサーバサイド描画手法を利用している。この手法では、クライアント（ウェブブラウザ）の要求に応じて、サーバは適切な地図画像を描画して送信する。クライアントは受信した地図画像をそのまま表示する。

それに対し、CMap ではクライアントサイド描画手法を利用する。この手法では、クライアントの要求に対し、サーバはベクトル情報の地理データを送信する。クライアントは受信したベクトルデータを基に地図の描画、表示を行う。

クライアントサイド描画手法を用いるとサーバサイドの実装が簡素化され、既存の WebGIS のように大規模なサーバを用意する必要がなくなる。さらに、地図の動的生成により柔軟な地図表示が可能となり、この手法で実装された CMap は地図表示の縮尺や回転（上部が北を指す通常の地図を上部が東や南南西など自由に変更できる）の無段階変化や、各地物（道路や鉄道、建物など）ごとに表示と非表示を選択可能なレイヤー切り替えをリアルタイムに行う機能を有している。

また、クライアントサイドで描画を行うため、サーバとクライアント間の通信形式が既存の WebGIS と異なる。既存 WebGIS が通信形式に画像データを用いるのに対し、クライアントサイド描画手法で実装された CMap は JSON 形式のテキストデータを用いる。通信形式にテキストデータを用いることにより通信量の削減が可能となる。そして、画像データは一部のみを拡大して利用するなどの操作が難しいが、テキストデータではそれが可能であるため地図データの再利用性が向上する。

表 1 にクライアントサイド描画手法と Google マップなどの既存 WebGIS との比較を示す。

表 1 描画手法の比較

	クライアントサイド 描画手法	サーバサイド 描画手法
地図生成の タイミング	リアルタイム	事前に用意
リクエスト時の 負荷の有無	あり (クライアント側)	なし
通信データ	テキスト	画像

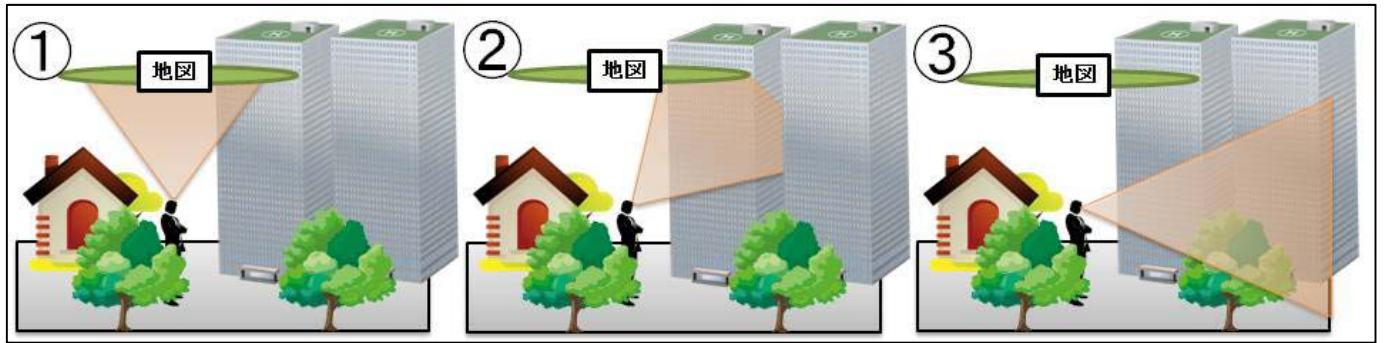


図 2 イメージ図

5. 提案手法

地図情報は上空からの視点で作成された二次情報であり、広域の地物に関する情報を収録し、かつ高い地理的精度をもっているが、人間が直感的に理解できるという視認性は劣る。一方ストリートビューに代表される画像は、人間が普段活動している地上で撮影した一次情報であるため、直感的かつ視認性が高い。しかし、市街地では視界が建物によって遮られるため広域を見渡すことはできず、また距離や方角などの地理的な位置関係を正しく把握することも困難である。

そこで、一次情報を軸とし、そこに二次情報を付与することにより、ユーザにとって視認が容易な一次情報で画面を操作しつつ広域の二次情報を利用できる新しい WebGIS の表示方法を提案する。

5.1. 表示方法

ユーザが普段、街中を散歩する際に有効な視野角は左右に広く上下に狭い。つまり、街の景観の情報の多くは下部にあり、上部には情報が少ない。この特性を利用し、ユーザが居る地点を中心とした二次情報を視野上部に付与する表示方法を考案した。

具体的には、まず一次情報を画面全体に表示し、視点を天上にしている場合は一次情報の表示域全てに二次情報を重ねるように表示し(図 2.①)、視点を下げるとともに二次情報の表示域を狭くし(図 2.②)、視点が下を向いているときは二次情報を非表示にする(図 2.③)。

5.2. システム構成

一次情報に Google 社が提供する Google Maps API[6] から使用できる Google ストリートビューを、二次情報に昭文社のデジタル地図 Mapple10000[7] を搭載した CMap を利用し、提案手法を実装したプロトタイプを作成した。システムの仕様と作成したプロトタイプで本学付近を表示した画面を表 2、図 3 に示す。

基本操作は Google ストリートビューの操作法と同じである。表示域をドラッグして視点を移動させるとそれに連動して CMap (二次情報) の表示が変わる。視点を右に変化させれば天上にある CMap が時計回り

の方向に変化し、視点を上に変化させれば CMap の表示域が徐々に広がり、視点を真上に向ければ全表示域に CMap が表示される。これらを全てリアルタイムに行う。表 3 に操作と機能を示す。

表 2 プロトタイプの仕様

画面サイズ	幅 1000[px] 高さ 600[px]
入力機器	マウス
CMap の透過率	80[%]
CMap に表示される構造データとその配色	一般道 灰色 県道 黄色 国道 黄色
CMap に表示される文字情報とその配色	駅名 赤色 交差点名 赤色

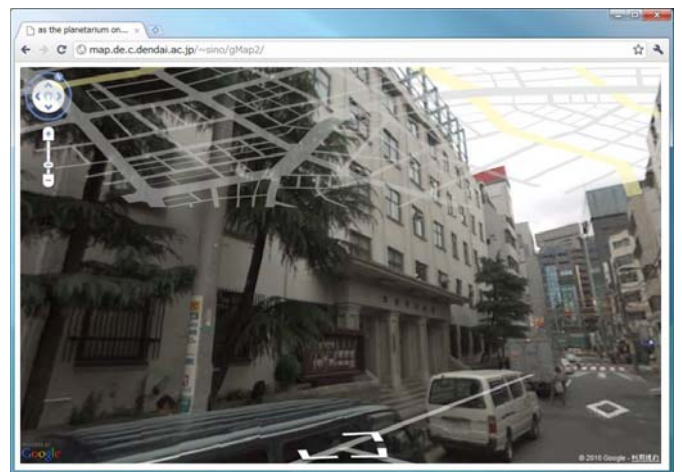


図 3 プロトタイプの表示画面

表 3 操作と機能の一覧

	操作	機能
GSV ¹	GSV 画面のドラッグ	視点移動
	矢印のクリック	現在地の移動(短距離)
CMap	CMap 画面のドラッグ	現在地の移動(長距離)

5.2.1. クライアントサイドの実装

ユーザ視点の一次情報と上空視点の二次情報は異なる視点であるため二つの情報を関連付けなければなら

¹ Google ストリートビューの略記

らない。つまり、一次情報の変化に合わせ二次情報の表示をリアルタイムに生成する必要がある。

本手法ではこの問題をクライアントサイド描画手法で実装された CMap を二次情報に用いることで解決する。

4 章で記したようにクライアントサイド描画手法は地図のリアルタイム生成により柔軟な地図表示が可能となる。そのため、Google ストリートビューの視点変化に対応した二次情報の表示が可能となる。

具体的には、JavaScript を用いて Google Maps API からユーザの操作で生じた視点（水平角と上下角）の変化を取得し、新しい視点の真北からの角度と水平からの角度を取得、真北からの角度から CMap の回転角度を、水平からの角度から CMap の表示域と遠近度を設定し CMap を再描画する。

そして、ユーザが Google ストリートビューを操作し位置を移動すると、JavaScript が Google Maps API から新しい位置情報を取得し、その情報を基に新しい地図データをサーバに要求し、返ってきた地図データを基に CMap を再描画する。

地図描画は HTML の Canvas エlement を使用する。サーバから送信された地図データを基に JavaScript から Canvas の API を呼び出すことにより地図を描画している。また、地図データの要求は非同期に行う。非同期で行うことによりユーザが地図を操作中であっても、それを阻害することなく地図データを取得することが可能となる。

クライアントサイドの実装図を図 4 に示す。

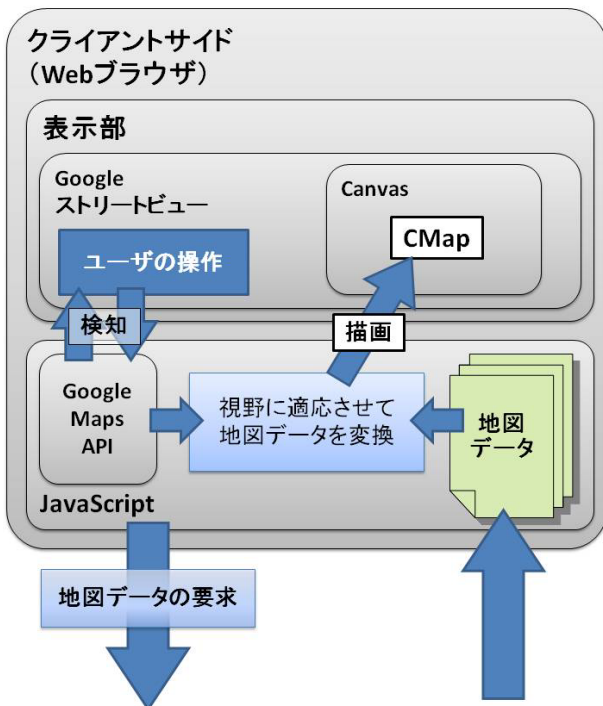


図 4 クライアントサイドの実装図

5.2.2. サーバサイドと通信部の実装

サーバサイドでは CGI として Perl を利用し、クライアントからの地図データ要求を受け取る。それを分析し、要求に合った地図データをデータベースから検索して取得、JSONP 形式に変換してクライアントに送信している。データベースには PostgreSQL とその拡張モジュールの PostGIS を利用している。そして、Perl からデータベースに接続するコネクタとして CPAN の DBI モジュールを利用している。また、通信部は JSONP 形式のテキストデータを利用する。図 5 にサーバサイドの実装図を、表 4 にサーバサイドの環境を示す。

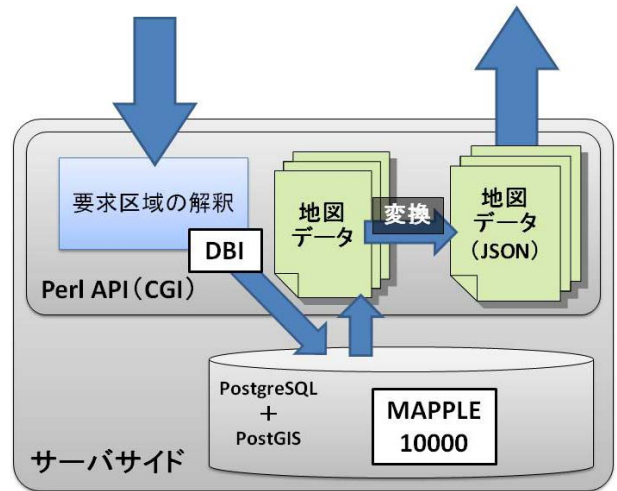


図 5 サーバサイドの実装図

表 4 サーバ環境

サーバ OS	Ubuntu Linux 9.04 32bit
CPU	Intel Core Duo T2500 @2.00GHz
Memory	2GB
Web サーバ	Apache 2.2.11
データベース	PostgreSQL 8.3.12 PostGIS 1.3.3
プログラム言語	Perl 5.10
地図データ	MAPPLE デジタルデータ 10000

6. 評価

作成したプロトタイプが円滑に利用可能かを判断する応答性と、CMap と Google ストリートビューの二つの地理情報を連携させることによって生じる誤差がないかを判断する整合性について評価を行った。

表 5 に評価に用いたクライアントサイドのコンピュータ環境を示す。なお、サーバサイドの環境は表 4 のコンピュータを利用した。

表 5 クライアント環境

OS	Microsoft Windows Vista Business 32bit
CPU	Intel Core2 Duo E8400 @3.00GHz
Memory	4GB
ウェブブラウザ	Google Chrome 8
表示場所	本学付近

6.1. 応答性

6.1.1. 評価項目とその方法

評価する項目は、初期表示にかかる時間とユーザによるマウス操作で生じる画面変移にかかる時間である。表 6 に測定する動作とその説明を示す。

測定方法は JavaScript と Chrome の Developer Tools¹ を使用して、測定する動作が始まるタイミングから画面変移が終了するまでの時間を ms (millisecond) 単位で 10 回測定し、その平均をとった。

表 6 測定項目

測定する動作	詳細
初期表示	最初にアクセスしてから画面が表示されるまでの時間
GSV の視点変更	GSV の操作によって生じる画面変移にかかる時間
GSV の位置移動	GSV の操作によって GSV の表示位置を移動したときに生じる画面変移にかかる時間
CMap の移動	CMap の地図移動操作によって生じる画面変移にかかる時間

6.1.2. 結果

各動作項目の測定結果を表 7 から表 10 に示す。なお、Google ストリートビューによる視点変更は視点の上下角で CMap の描画領域が変化するため、前方と上部の表示で描画時間を測定した。

表 7 初期表示の結果

画面表示にかかる合計時間 [ms]	内 CMap の描画時間 [ms]
785	17

表 8 視点変更の結果

画面表示にかかる合計時間 [ms]	内 CMap の描画時間 [ms]	
	前方表示	上部表示
280	15	21

表 9 位置移動の結果

画面表示にかかる合計時間 [ms]	内 CMap の描画時間 [ms]
524	17

表 10 CMap 移動の結果

画面表示にかかる合計時間 [ms]	内 CMap の描画時間 [ms]
750	16

6.1.3. 考察

いずれの動作項目でも応答速度は 800[ms]以内であり、プロトタイプの応答性は高く、円滑に利用可能と言える。

動作項目ごとに比較すると、Google ストリートビューによる視点変更が高い応答性を示しており、続いて Google ストリートビューによる位置移動、最後に初期表示と CMap による位置移動の順で応答性が悪くなる。

これは Google ストリートビューにキャッシュ機能と先読み機能が搭載されているためだと思われる。これらの機能により、Google ストリートビューはユーザが閲覧している場所から見える場所の画像で未取得のものをサーバから先行して取得する。従って、Google ストリートビューによる視点変更の応答性が高い。次に応答性が高い Google ストリートビューによる位置移動もキャッシュ機能による効果である。Google ストリートビューによる位置移動は、ユーザが閲覧している場所から見える近隣の場所への移動である。従って、画像を流用することによりサーバから画像を取得する時間が減少する。以上のことから、初期表示と CMap による位置移動の応答性が同等である理由は、Google ストリートビューが視界にある全ての画像を取得するからである。若干 CMap による位置移動の応答性が高い理由は HTML や JavaScript など、初期表示に必要なデータの取得が無いためである。

なお、CMap の描画速度は非常に高速であり、最も遅い処理時間で 21[ms]と人間ではほとんど視認できない速度で動作する。

6.2. 整合性

6.2.1. 評価項目とその方法

位置の誤差と向きの誤差について検証を行う。位置の誤差は、CMap による位置移動で交差点や駅を表示し、Google ストリートビューと CMap の両方で表示されているかを判断する。

向きの誤差は、従来の地図表示による地物の位置関係を利用する。例えば、従来の地図表示では本学から北の方角にはお茶の水駅がある。この位置関係を利用し、Google ストリートビューで北を表示したときに CMap でお茶の水駅が表示されるかを検証する。

表 11 に検証する項目とその説明の一覧を示す。

表 11 検査項目

検証する項目	詳細
位置の誤差	GSV と CMap の位置情報に誤差が無いか (GSV が東京駅を表示している場合、CMap も東京駅を表示しているか)
向きの誤差	GSV と CMap が同一の方角を指しているか (GSV の表示が北を指している場合、CMap も北を指しているか)

6.2.2. 結果と考察

位置と向きの両方で目立った誤差は生じなかった。Google ストリートビューの仕様上、道幅の広い道路では中央を表示しないため若干の誤差が生じたが通常の利用では影響が無い程度である。

図 6 に交差点「淡路町」で位置の誤差を確認する表示例を、図 7 に本学と「お茶の水駅」の位置関係による向きの誤差を確認する表示例を示す。

¹ <http://code.google.com/intl/en/chrome/devtools/>



図 6 位置の誤差を確認する表示例

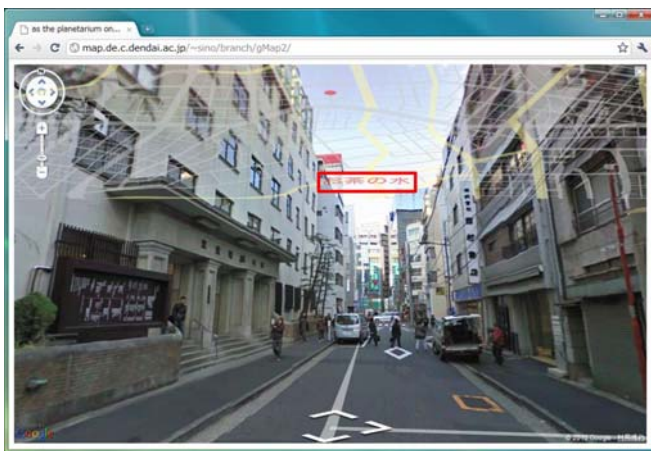


図 7 向きの誤差を確認する表示例

7. まとめと今後の課題

Google ストリートビューは直感性が高いが、広域の地理情報を持たず、全体の地理的な位置関係が把握し難いという問題があった。この問題に対し、本研究では、上空から俯瞰した地図情報である従来の地図表示と地上で撮影した画像情報である Google ストリートビューの表示を同一の画面上で合成表示することで解決を試みた。

そして、提案手法の有効性を確かめるために、プロトタイプを作成した。プロトタイプは Google Maps API から利用できる Google ストリートビューをベースとして表示し、その上にクライアント描画手法で実装された WebGIS である CMap を重ねて表示させることにより実装した。

プロトタイプを利用することによりユーザは、着目している地点の詳細情報を Google ストリートビューの表示から直感的に得ると同時に、全体の地理的な位置関係を従来の地図表示から正確に把握することが可能となった。

さらに、プロトタイプがインタラクティブに利用可能かを判断する応答性と、Google ストリートビューと CMap の二つの地理情報に矛盾が生じていないかを判

断する整合性の二つの項目について実測評価を行った。結果はいずれも良好であり、プロトタイプの表示は正確で高速に動作することを確認した。

以上のことから、本研究の目的である Google ストリートビューで広域の地理情報を取得可能にすることを達成したと言える。

今後の課題として、まず、ユーザインタフェースの改良が挙げられる。

現在のプロトタイプ of CMap の表示は道路ポリゴンをメインとして、追加の文字情報に交差点名と駅名を表示させている。これは、多くの情報を表示させると煩雑になりユーザが混乱することを防ぐためであるが、表示が単純で自由度が低い。この問題は、表示させる項目を選択可能なユーザインタフェースを搭載するなどの方法で解決可能である。

また、他分野への応用として、スマートフォンを利用して、現地を歩く人にその地点の映像と地図を統合表示する AR (拡張現実) の利用が考えられる。

本研究では一次情報として Google ストリートビューを利用したが、Google ストリートビューは情報の更新頻度が低く、例えば、表示される店舗 A が現実では別の店舗 B に入れ替わっていたなどリアルタイム性に欠ける。また、Google ストリートビューはその仕様上、道路からの視点であり、さらに時間帯も日中に限定されている。AR とプロトタイプを併用利用すればそのような限定条件は無く、任意の場所、時間で利用可能となる。

参考文献

- [1] 矢島健太郎, 山崎優, 井上潮 “クライアントサイド描画手法を用いた WebGIS の提案と実装”, DEIM 2009.
- [2] 小関章太郎, 山本大介, 北神慎司, 高橋直久, “Emma における Focus+Glue+Context 型地図の実現法”, DEIM 2009.
- [3] 篠田陽平, 井上潮, “フィッシュアイモデルを用いた地図表示システムの提案”, DEIM 2009.
- [4] 小林加織里, 北山大輔, 角谷和俊, “オンライン地図におけるユーザ操作に基づくストリートビュー自動生成システム”, DEIM 2010.
- [5] 西尾信彦, 坂本憲昭, 新井イスマイル, “Google Street View とマッシュアップした実世界指向 CMS”, 情報処理学会 2009.
- [6] Google, Google Maps API ドキュメント, <http://code.google.com/intl/ja/apis/maps/>
- [7] 昭文社, マップルデジタル地図データ, <http://www.mapple.co.jp/corporate/product/01.html>