

# Leaflet を用いた Focus+Glue+Context マップインタフェースの開発

竹内 健祐<sup>1</sup> 山本 大介<sup>1</sup> 高橋 直久<sup>1</sup>

概要：OpenStreetMap といったオープンデータの地図データや、Leaflet などのオープンソースの地図制御用コードの登場により、Web マップは扱いやすいものとなった。我々は注目する地点を拡大表示する Focus 領域、その周辺の地域を表示する Context 領域、2つの領域間の縮尺の違いによる歪みを吸収する Glue 領域の3つからなる Focus+Glue+Context マップを用いた Web マップシステム Emma を提案してきた。しかし、EMMA は地図表示やレイヤー管理、タイリング処理などの基本的な制御部分も含めて全て我々独自で開発したクローズドなシステムであり、最新の Web マップ技術の取り込みや、機能拡張・操作性の向上などのカスタマイズが容易ではないという問題があった。上記の問題を解決するために本研究では EMMA の機能を Web マップのオープンデータ、オープンソースと共存できるようにして、機能拡張やカスタマイズを容易にした Open EMMA システムを提案する。また本論文では提案システムに基づくプロトタイプシステムを実装した。さらに、作成したプロトタイプシステムに対する評価実験を行い、提案手法の有用性を評価した。

## Development of Focus+Glue+Context Map Interface Using Leaflet

KENSUKE TAKEUCHI<sup>1</sup> DAISUKE YAMAMOTO<sup>1</sup> NAOHISA TAKAHASHI<sup>1</sup>

### 1. はじめに

OpenStreetMap [1] といったオープンデータの地図データや、Leaflet [2] などのオープンソースの地図制御用コードの登場により、Web マップは扱いやすいものとなった。Leaflet は Web マップのための軽量 JavaScript ライブラリで、多くの Web マップシステムと同様に、多数のレイヤーを位置情報に基づいて重ね合わせて描画することにより、道路網、鉄道路線、施設データなど多様な空間データを1つの地図画面に表示することができる。ユーザは、レイヤーを選択したり、新たなレイヤーを追加したりして、用途や趣向に応じた地図を容易に作成することができる。

また我々は Focus+Glue+Context 型のマップである EMMA システム [3-6] を提案してきた。Focus+Glue+Context マップは、注目する特定の地域を拡大して表示する Focus 領域、周辺の地域を表示する Con-

text 領域、スケールの異なる2つの領域間の縮尺率の違いによって生じる歪みを吸収する役割を果たす Glue 領域によって構成された、視認性と一覧性を両立させた Web マップである。Focus+Glue+Context マップの例を図1に示す。

従来の EMMA は商用地図データを用いているためオープンでなかった。その後、商用地図データ商用地図データから OpenStreetMap へ移行した [7,8] が、EMMA は地図表示やレイヤー管理、タイリング処理など基本的な制御部分も含めてすべて我々独自で開発した Java と ActionScript によるクローズドなシステムである。このため、既存システムとの連携やスマートデバイスに対応させるなど機能拡張やカスタマイズが容易ではないという問題がある。

この問題に対して、Leaflet と OpenStreetMap を用いて EMMA システムを再構築することで解決を図る。しかし、Leaflet を用いるには EMMA を ActionScript から JavaScript に移行し、それらすべての機能を書き換える必要がある。EMMA は場所により縮尺の異なる可変スケールマップであるため、各領域をそれぞれ異なるレイヤーの

<sup>1</sup> 名古屋工業大学大学院工学研究科  
Graduate School of Engineering, Nagoya Institute of Technology

ラスタマップとして Leaflet の機能を用いると、EMMA の機能と不整合が生じることがある。道路描画評価用 EMMA システム [8] では、可変スケールの領域では道路データのみを扱っていたため、不整合の問題が生じなかったが、任意のレイヤーを重ね合わせると不整合の問題が発生する可能性がある。

そこで、本稿では Focus と Glue の領域を画面にマッピングしたラスタマップからなる EMMA レイヤーを作成し、このレイヤーの中に、場所による縮尺の違いを閉じ込めて隠蔽し、EMMA の機能を Web マップのオープンデータ、オープンソースと共存できるようにして、機能拡張やカスタマイズを容易にした Open EMMA システムを提案する。



図 1 Focus+Glue+Context マップ

## 2. 提案システム

### 2.1 機能概要

提案システムの主な機能は次のとおりである。

#### EMMA レイヤー生成機能

サーバは、Focus の中心座標、半径、縮尺が与えられると、ラスタマップ DB から Focus のタイル画像を配信する。また、Glue の形状と境界の縮尺が与えられると、Glue の道路網を総描したラスタマップを生成して配信する。

クライアントは、Focus と Glue のラスタマップをそれぞれ個別にサーバからロードして合成し、EMMA レイヤーを作成する。これにより、Focus と Glue の移動特性に応じた描画が可能になる。

#### Focus と Context のタイリング機能

クライアントは、Leaflet のタイリング機能を用いて、Focus と Context の周辺区域のタイルのラスタマップを予めサーバからロードしておき、Focus と Context の移動時における描画の応答時間を短縮する。

#### レイヤー重ね合わせ機能

クライアントは、Leaflet の機能を用いて、必要なレイヤーの空間データをサーバからロードして画面上に重ね

て描画する。また、YOLP [9] など外部の開発者用ツールキットの Web マップ API を用いて空間データを入手してマッシュアップにより重ねることにより、ユーザは、状況、趣向に応じて空間情報を検索するシステムを容易に構築できるようにする。

#### 入力イベント操作機能

タッチ操作など端末の入力イベントから画面上の座標値を求めて、そのイベントに対応するアイコンなどをマップ上に描画する。EMMA レイヤーに座標変換機能を組み込んでおくことにより、イベント処理において、場所による縮尺の変化を考慮しないで描画しても、Focus と Glue 領域においても位置の整合性を保つことができる。

## 2.2 システム構成

提案システムの構成図は図 2 のようになっている。

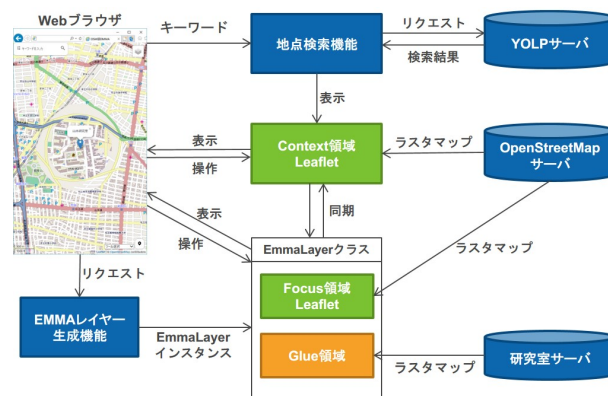


図 2 提案システムの構成図

ユーザは Web ブラウザ上で提案システムの HTML ファイルを開くとシステムが起動するようになっている。ユーザは EMMA レイヤー作成機能を使うことによって地図上の任意の場所に Focus+Glue 領域を生成することができる。生成された Focus+Glue 領域と Context 領域はいずれもドラッグ操作で移動させることができ、Focus+Glue 領域は領域の境界線をドラッグすることでサイズを変更することができる。Focus 領域と Context 領域は OpenStreetMap サーバから地図タイル画像を取得し、Glue 領域は研究室サーバからラスタマップ画像を取得して表示する。レイヤー重ね合わせ機能として今回は地点検索機能を実装しており、ユーザは近くのコンビニなど調べたいキーワードを入力することで検索結果を表示することができる。

## 3. 提案手法の実現法

本章では、提案システムの実現法について述べる。

### 3.1 EmmaLayer クラス

この節では leaflet を用いた Focus 領域表示方法ならびに Focus+Glue 領域を扱う EmmaLayer クラスについて詳しく

く説明する。EmmaLayer クラスの主な変数とメソッドを表 1,2 に示す。

表 1 EmmaLayer クラスの変数

変数	型	説明
focusGlueLayer	HTMLElement	Focus 領域と Glue 領域を含んだ HTML 要素
contextMap	map object	Context 領域を表す Leaflet のオブジェクト
focusLayer	HTMLElement	Focus 領域の HTML 要素
glueLayer	HTMLElement	Glue 領域の HTML 要素
circleLayer	HTMLElement	サイズ変更時に用いる円を表示する HTML 要素
options	構造体	Focus 領域の Leaflet を生成するとき用いる緯度経度や円の半径などのオプション群

表 2 EmmaLayer クラスのメソッド

メソッド名	戻り値	説明
onAdd (map object)	this	インスタンスが生成されたときに実行するメソッド
setSize (radius,radius)	this	引数から Glue 領域と Focus 領域のサイズを変更する
setPosition (radius,radius,point)	this	Glue 領域と Focus 領域の半径を用いて指定した緯度経度が中心に来るように Focus+Glue 領域の位置を変更する
createFocusMap (HTMLElement)	map object	Leaflet のインスタンスを生成する
updateFocus()	this	Focus 領域の緯度経度が指す Context 領域上の座標に Focus+Glue 領域を移動し、Glue 領域を更新する
update(latlng)	this	指定された緯度経度が Focus 領域の中心に来るようにし、それに応じて Glue 領域を更新する
updateFromMouse (event)	this	マウスイベント緯度経度を取得し、update に渡す

### 3.1.1 Focus 領域の作成

Focus 領域は Context 領域上のある地点を詳細に表示するために存在する。そのため Focus 領域は Context 領域上の詳細に表示したい地点の緯度と経度を取得し、Web ブラウザ上の適切な位置で表示させる必要がある。これについては以下のような手順で行う。

- (1) Leaflet のイベントハンドラを用いて、Context 領域において地図上の点をクリックしたときに緯度経度を取得する関数を作成する。これによって Focus 領域を置きたい地点の緯度経度を知る事が出来る。
- (2) 手順 1 で取得した緯度経度を元に Leaflet の latlngToLayerPoint 関数を用いて Web ブラウザ上での座標を取得する。latlngToLayerPoint コマンドは Leaflet での緯度経度を Web ブラウザ上での座標に変換してくれる関数である。これを用いて EmmaLayer クラスのインスタンスを生成する。

- (3) 各領域の HTML 要素を生成する。このときすべての領域は css の border-radius を用いて円形になるようにする。次にプロパティから半径の値を取得し、setSize メソッドで各領域のサイズを変更する。
- (4) 手順 2 で得た Web ブラウザ上の xy 座標に setPosition メソッドで Focus+Glue 領域を移動させる。Focus 領域は手順 2 で得た Web ブラウザ上の xy 座標からの半径の値をそれぞれ引いたもの Web ブラウザ上での絶対座標とする (図 3)。また Focus 領域の座標から Glue 領域の半径と Focus 領域の半径の値の差を引いたものを Glue 領域の Focus 領域からの相対座標とする (図 4)。
- (5) createFocusMap メソッドで Focus 領域の HTML 要素に Leaflet インスタンスを作成する。

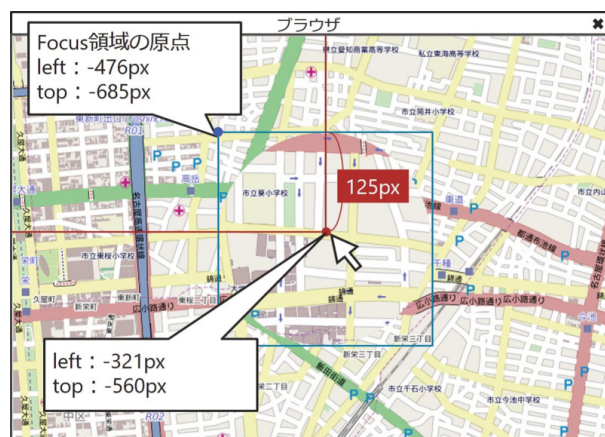


図 3 Focus 領域の原点



図 4 Glue 領域の原点

### 3.1.2 Glue ラスタマップ生成

Glue 領域の道路網を総描したラスタマップは先行研究である道路描画評価用 EMMA システム [8] によって開発されたものを使っている。サーバーは Focus 領域の中心座標、各領域の半径と縮尺、道路網の種類が与えられると、

道路網を総描したラスタマップを生成して配信する。この際サーバで生成されるラスタマップ画像は正方形である。クライアントは表示時に css の border-radius を用いて丸く切り取って表示している。

### 3.1.3 Focus + Glue 領域の移動

Focus + Glue 領域はユーザの操作によって移動できる必要がある。jQuery UI の Draggable [10] を使用してドラッグ開始時、ドラッグ中、ドラッグ終了時にそれぞれ以下のように処理することで移動処理を実現している。

**ドラッグ開始時** Focus + Glue 領域をドラッグした際に Context 領域も一緒に動かないように、Context 領域の Leaflet のドラッグイベントを一時的に取得しないように設定する。

**ドラッグ中** Context 領域の Leaflet のイベントハンドラでマウスポインタの座標を常に取得する。取得した座標をもとに Focus+Glue 領域を描画する。

**ドラッグ終了時** Context 領域の Leaflet のドラッグイベントを取得するように再設定する。

### 3.1.4 サイズの変更

提案システムでは Focus 領域や Glue 領域の縁をドラッグすることでサイズを変更することができる。サイズ変更に関しては以下の手順で行う。

- (1) 各領域の境界線上にマウスポインタがあるとき、サイズ変更フラグを true にする。
- (2) サイズ変更フラグが true の時に Focus または Glue 領域をドラッグをすると、領域が移動する代わりにサイズを変更する処理を開始する。
- (3) ドラッグ中は常にマウスポインタの座標を取得し、Focus 領域の中心座標からの距離を計算する。
- (4) この距離を一定時間前の各領域の半径と比較し、差分を各領域のサイズに適用する。
- (5) 変更したサイズ分の領域を表示するために Focus 領域の Leaflet について invalidateSize メソッドを実行する。invalidateSize メソッドは Leaflet に含まれている機能で、マップコンテナのサイズが変更されたかどうかを確認し、サイズ変更があれば更新するメソッドである。

### 3.1.5 縮尺の変更への対応

Context 領域の縮尺が変わったとき、Web ブラウザ上でのある地点を表示する位置は変化してしまうため、Focus+Glue 領域は Context 領域の縮尺の変化に合わせて動かななくてはならない。そこで EmmaLayer クラスは Context 領域の拡大縮小に対してのイベントリスナーを持つ。縮尺が変わるたびに縮尺変更後のブラウザ上での位置を計算する。

## 3.2 Focus+Glue 領域の Leaflet レイヤー化

Leaflet はいくつかのレイヤー (pane) から成り立ってお

り、表 3 のようになっている。Leaflet はこの多数のレイヤーを位置情報に基づいて重ね合わせて描画しているが、Focus+Glue 領域は Context 領域の Leaflet とは独立しているため、表示に不整合が生じてしまう。

そこで新たに Focus+Glue 領域を扱うレイヤーとして focusgluePane を表 4 の様に作成する。EmmaLayer クラスによって Focus+Glue 領域を focusgluePane に作成すると、Focus+Glue 領域は Leaflet においてその他のレイヤーと同じように位置情報に基づいて重ね合わせて描画されるようになる。

表 3 Leaflet の階層

Pane	Z-index	説明
tilePane	200	マップのタイル画像を表示する層
overlayPane	400	ポリゴンやパスなどを表示する層
shadowPane	500	影を表示する層
markerPane	600	マーカーを表示する層
tooltipPane	650	ツールチップを表示する層
popupPane	700	ポップアップを表示する層

表 4 提案システムでの Leaflet の階層

Pane	Z-index	説明
tilePane	200	マップのタイル画像を表示する層
overlayPane	400	ポリゴンやパスなどを表示する層
focusgluePane	450	Focus+Glue 領域を表示する層
shadowPane	500	影を表示する層
markerPane	600	マーカーを表示する層
tooltipPane	650	ツールチップを表示する層
popupPane	700	ポップアップを表示する層

## 3.3 領域間でのマーカー位置の補正

Context 領域・Glue 領域・Focus 領域はそれぞれ縮尺が異なるため、地図上にあるマーカーは各領域に対して表示位置を変更する必要がある。そこで提案システムではまずマーカーが本来指し示す緯度経度をマーカーのオプションとして保存しておく。Context 領域上のマーカーが Focus+Glue 領域に重なったときに、そのマーカーが Glue 領域ではどの位置に存在するかを Glue 領域を生成するときと同じ計算を使うことで計算し、その位置にマーカーを移動させる。また、マーカーが Focus+Glue 領域から外れた際にはオプションから本来の緯度経度を取得し、その位置にマーカーを移動させる。

## 3.4 地点検索機能

ユーザが入力したキーワードをもとに地図の中心から近い順に検索結果を表示する機能である。実装にあたっては YOLP [9] を用いた。YOLP とは Yahoo! Open Local Platform のことで、Yahoo!JAPAN がデベロッパー向けに

提供する地図・地域情報の API・SDK である。これについては以下のような手順で行う。

- (1) ユーザはキーワードを入力し検索ボタンをクリックする。プログラムは Ajax を使って YOLP にキーワード（クエリ）と地図の中心の緯度経度と検索件数を送り、json 形式で検索結果を受け取る。
- (2) 受け取った json ファイルから地点の名称と座標の組を取得する。
- (3) 取得した座標に Leaflet でマーカを打つ。
- (4) 名称一覧表を作成し、名称をクリックすることでその座標に固定された Focus+Glue 領域を生成する。ここでいう固定された Focus+Glue 領域とは、ユーザによってドラッグ操作による Focus+Glue 領域の移動が行えないものを指す。固定する理由は、ユーザの誤操作によって検索結果の地点から Focus+Glue 領域が外れてしまうことを防ぐためである。

### 3.5 スマートフォン向け UI

スマートフォンでの操作は PC とは異なるために、タッチ操作に対応していない従来システムではスマートフォンで利用する場合に操作できない場面など多くが存在した。そこで JavaScript のライブラリである jQuery [10] を用いて PC・スマートフォン問わず Emma システムが動くようにプログラムを組んだ。また、スマートフォンは画面が小さいためアイコンや機能などを前面に出して画面を圧迫するのではなく、使用頻度の高い機能を常に前面に出し、使用頻度の低いものを mmenu [11] を用いたスライドメニューの中に入れている。

## 4. プロトタイプシステム

システムの開発は、Windows10 で Microsoft Visual Studio Community 2015 環境のもと JavaScript と HTML と CSS を用いて開発した。また、動作確認には主に Google Chrome と Android 端末として Galaxy S7 edge, iOS 端末として iPad Air2 を用いた。

### 4.1 システムの動作

システムを起動すると、図 5 のように表示するようになっている。

### 4.2 Focus+Glue 領域の作成

注目したい地点に Focus+Glue 領域を作成するには画面右下のプルダウンメニューから可動 Focus 生成または固定 Focus 生成を選び、地図上の注目したい地点をクリックすることでその地点に Focus+Glue 領域を作成できる（図 6）。可動 Focus 生成はドラッグ操作によって自由に動かせる Focus+Glue 領域を作成するもので、固定 Focus 生成はドラッグ操作によって動かすことのできない Focus+Glue

領域を作成するものである。

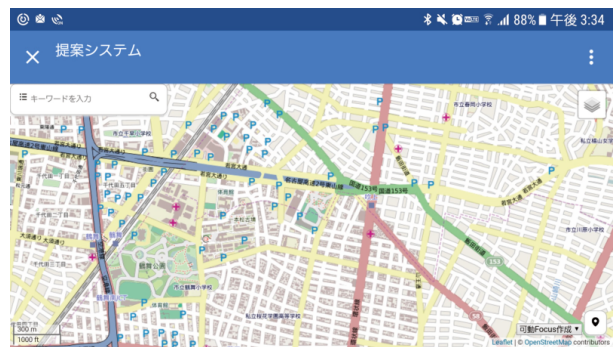


図 5 起動画面 Android

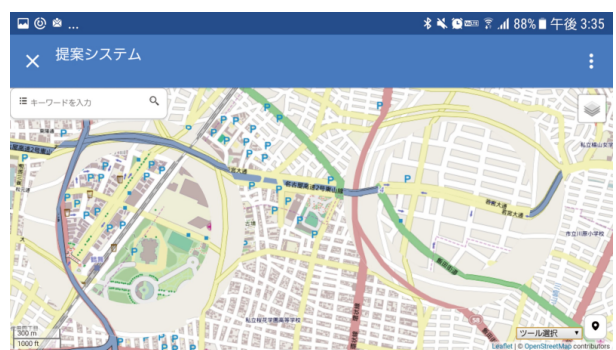


図 6 Focus+Glue 領域の作成

### 4.3 Focus+Glue 領域の削除

Focus+Glue 領域を削除するには画面右下のプルダウンメニューから Focus 削除を選び、削除したい Focus+Glue 領域をクリックすることで削除できる。

### 4.4 Focus+Glue 領域のサイズ変更

Focus+Glue 領域のサイズを変更するにはまず Focus 領域または Glue 領域の境界線上にマウスを持っていく。マウスが境界線上にあると領域に影のエフェクトがかかり、この状態の時にドラッグすることでサイズが変更できる。スマートフォンでは直接境界線上をタッチしてドラッグすることでサイズ変更ができる。デフォルトでは Focus 領域と Glue 領域は同時にサイズ変更されるようになっているが、4.6 小節で説明するスライドメニューからサイズ変更時の動作を変更し、別々にサイズを変更することもできる。

### 4.5 地点検索

左上の検索フォームにキーワードを入力することで地点検索を行う事が出来る。調べたいキーワードを入力し検索ボタンを押すと検索結果が表示される（図 7）。検索結果をクリックすることで図 8 のようにその地点に Focus+Glue 領域を作成する。

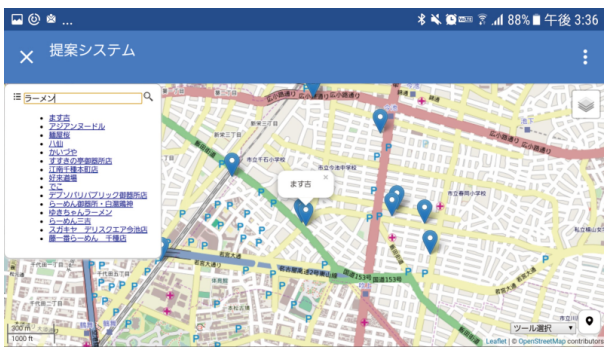


図 7 地点検索 1

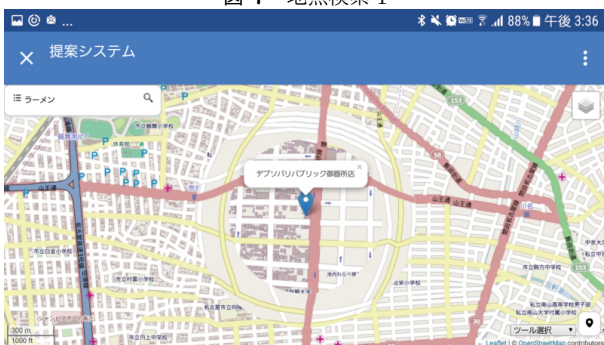


図 8 地点検索 2

#### 4.6 スライドメニュー

画面左上のメニューアイコンをクリックすることで図 9 のようにスライドメニューを表示させることができる。スライドメニューでは Glue 領域の処理方法の変更と Focus+Glue 領域のサイズ変更時の動作について変更できる。Glue 領域の処理方法は先行研究で開発された Glue 領域の描画方法を選ぶ事ができ、すべての道を表示させたり間引いて表示したりと変更することができる。Focus+Glue 領域のサイズ変更時の動作は Focus+Glue 領域のサイズ変更の際に Focus 領域と Glue 領域の両方を同時にサイズ変更するか個別に変更するかを選ぶ事が出来る。

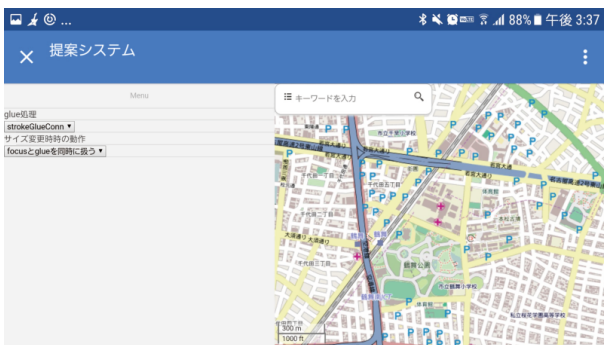


図 9 スライドメニュー

## 5. 評価実験

評価実験では、従来システムである道路描画評価用 EMMA システム [8] と、本論文の提案手法である Open EMMA システムを比較評価する。

### 5.1 実験方法

被験者(本学学生 5 名)に従来手法のシステムと提案手法のシステムの二つを体験してもらい、従来システムと提案システムを比べてもらう。また従来システムにない機能については、提案システムのみで評価をしてもらう。実験には比較しやすいように iPad でブラウザを 2 画面表示させ、左側に提案システム、右側に従来システムを表示させたものをを用いる(図 10)。

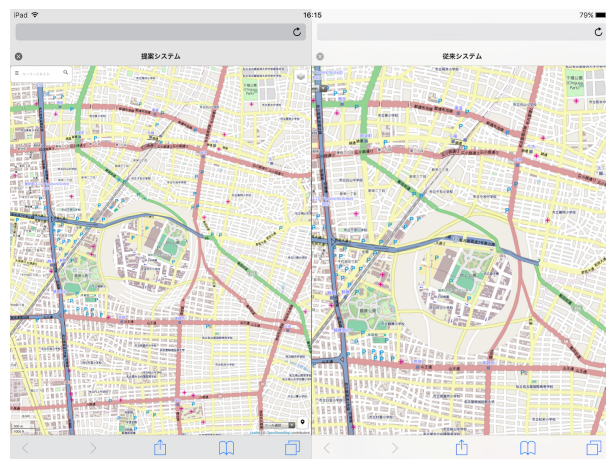


図 10 iPad で両システムを同時に表示

被験者に機能を一通り説明した後、自由に触ってもらい以下のアンケートに 5 段階評価で回答してもらおう。1 を「そう思わない」、5 を「そう思う」とし、質問の内容は 5 に近づくほど満足度が高くなっている。

- (1) Focus 領域の描画速度にストレスを感じない
- (2) 注目したい点に Focus 領域を簡単に持って来る事が出来る
- (3) 生成時の Focus+Glue 領域のサイズは適切である
- (4) Focus+Glue 領域を地図上に固定できるのは便利である
- (5) Focus+Glue 領域の作成・削除が容易である
- (6) 地点検索機能は使いやすい
- (7) 地点検索機能で調べた地点の位置は把握しやすい
- (8) タッチ操作は直感的である
- (9) タッチ操作でのサイズ変更がしやすい
- (10) スライドメニューは便利である

### 5.2 結果

アンケート調査の結果を図 11, 図 12 に示す。また、アンケート結果で得られたコメントを以下に記載する。

- 画面のサイズに対してのプルダウンメニューやボタンのサイズが小さいと感じた。高解像度デバイスでは繊細な操作が必要になりそう。
- ピンチ操作で Focus+Glue 領域のサイズを変更できると思った。

- Focus+Glue 領域のある地点に作っておいて、他の場所を見て回れるのは便利。
- 検索機能を使用した後に、Focus 領域とかが固定されているものを簡単に外せると便利だと思った。
- Focus 領域と Glue 領域のサイズ変更をするときにうまく縁をドラッグできずイライラした。
- 従来システムと比べると Focus 領域の動きが滑らかになった。
- Focus 領域が複数作れるのは非常に便利だし削除も簡単だった。
- スライドメニューは现阶段ではそんなにいらなと思う。

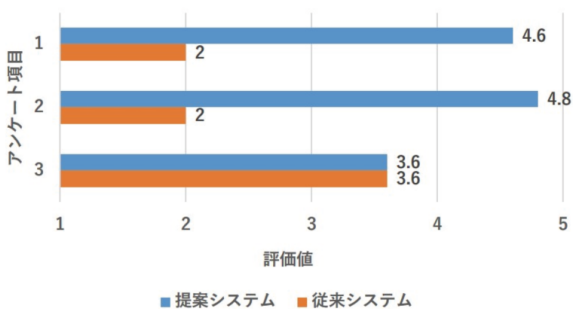


図 11 項目 1~3 の結果

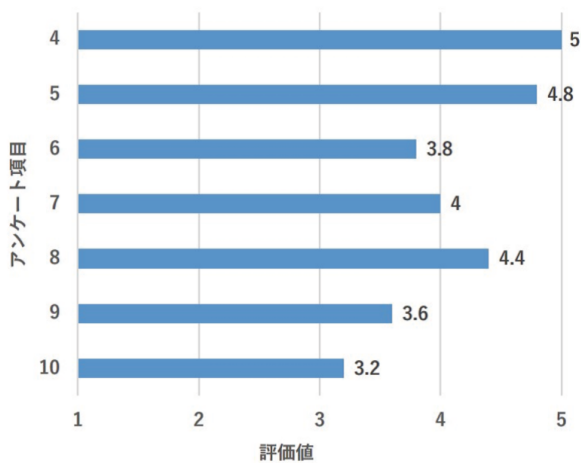


図 12 項目 4~10 の結果

### 5.3 考察

「Focus 領域の描画速度にストレスを感じない」という項目において提案システムは 4.6 であり、従来システムでは 2 と大きく差が表れた。Focus 領域を Leaflet に変えることによって描画速度が従来システムに比べて高速になったことが被験者の満足度に大きく影響しているのがわかる。「注目したい点に Focus 領域を簡単に持って来る事が出来る」という項目においても提案システムは 4.8 であり、従来シ

ステムでは 2 と大きく差が表れた。従来システムの Focus 領域を Web ブラウザ上の座標に固定する方式に対し、提案システムの Context 領域上の座標に固定する方式のほうが使い勝手が良いことがわかる。「生成時の Focus+Glue 領域のサイズは適切である」という項目においては、従来システム提案システムともに 3.6 と評価が一緒であった。提案システムでは Focus 領域の半径を 125px、Glue 領域の半径を 180px としているが解像度の高い画面では小さく見えてしまうとの声があった。画面の解像度に応じて Focus 領域と Glue 領域のサイズを変えるような機能の実装が必要不可欠であると考ええる。

「Focus+Glue 領域を地図上に固定できるのは便利である」という項目では 5 の評価を得た。被験者から「Focus+Glue 領域のある地点に作っておいて、他の場所を見て回れるのは便利」というコメントも得られた。「Focus+Glue 領域の作成・削除が容易である」という項目では 4.8 の評価を得た。しかし、被験者から「画面のサイズに対してのプルダウンメニューやボタンのサイズが小さいと感じた。高解像度デバイスでは繊細な操作が必要になりそう」とのコメントが得られたように、作成・削除ツールを選択するプルダウンメニューのサイズをもう一度考える必要がある。「地点検索機能は使いやすい」という項目では 3.8、「地点検索機能で調べた地点の位置は把握しやすい」という項目では 4 という結果が得られた。提案システムの地点検索機能は地図の中心からキーワードに当てはまるものを近い順に 15 件文字とマークで表示する固定された機能であり、検索オプションなど設定できないといったところが評価結果に表れているのではないかと考える。また被験者から「検索機能を使用した後に、Focus 領域とかが固定されているものを簡単に外せると便利だと思った」とのコメントもあり、検索機能とその結果を用いた Focus+Glue 領域の表現方法については今後の課題である。「タッチ操作は直感的である」の項目では 4.4 の評価を得ているが、「タッチ操作でのサイズ変更がしやすい」の項目では 3.6 と前項より評価が低い結果となった。被験者からは「Focus 領域と Glue 領域のサイズ変更をするときにうまく縁をドラッグできずイライラした」とのコメントもあり、サイズ変更の手法は今後改善が必要である。「スライドメニューは便利である」の項目については 3.2 の評価を得られたことに関して、提案システムではスライドメニューの中には Glue の描画方法とサイズ変更時の挙動の 2 つしか項目がなかったためか、被験者から「スライドメニューは现阶段ではそんなにいらなと思う」とのコメントが多く挙がった。

## 6. 関連研究

文献 [8] では、Focus+Glue+Context マップにストロークと呼ばれる、認知心理学に基づいた道なりに続く道を適用し、地図の利便性の向上を図る研究の一部として本研

究の土台となっている JavaScript 版の EMMA を実装している。Context 領域では Leaflet で OpenStreetMap を表示し、Focus+Glue 領域を研究室のサーバーから画像を取得して表示している。

文献 [20] では、モバイル端末向け Flash 版 Focus+Glue+Context マップにおいて主にフリック入力などを用いたモバイル端末での操作手法について述べている。フリックの他に長押しドラッグやダブルタップなどに対応させ、マウスで操作する EMMA とは異なった捜査手法を提案している。

## 7. おわりに

本稿では Focus と Glue の領域を画面にマッピングしたラスタマップからなる EMMA レイヤーを作成し、このレイヤーの中に、場所による縮尺の違いを閉じ込めて隠蔽し、EMMA の機能を Web マップのオープンデータ、オープンソースと共存できるようにして、機能拡張やカスタマイズを容易にした Open EMMA システムを提案した。Leaflet を用いた Focus 領域表示ならびに Focus+Glue 領域を扱う EmmaLayer クラスを作成し、Focus と Glue の移動特性に応じた描画を可能にした。領域間でのマーカー位置の補正機能や地点検索機能を実装し、またスマートフォンでも動くインタフェースを実装した。提案システムに基づいてプロトタイプシステムを実装し、Emma システムの利便性が向上したかどうかをプロトタイプシステムを用いて評価実験を行った。実験を通して、Focus 領域の描画速度や扱いやすさに関して従来システムよりも良い結果を得る事ができ、Focus+Glue 領域の生成機能についてもよい結果を得る事が出来た。しかしユーザインタフェースに関しては評価が低く、改善が求められる結果となった。

今後の課題として、Glue 領域の描画速度の向上や 2 つの Focus+Glue 領域が重なったときの各領域の挙動、解像度によって各領域やボタン等のサイズを変化させるユーザインタフェースの改良などが挙げられる。

## 謝辞

本研究は JSPS 科研費 26330136, 25700009, 科学技術振興機構 CREST, および、総務省 SCOPE の助成を受けたものです。この場を借りて、感謝の意を表します。

## 参考文献

- [1] OpenStreetMap Japan, 入手先 (<https://openstreetmap.jp/>) (参照 2017-04-24).
- [2] Leaflet - a JavaScript library for interactive maps, 入手先 (<http://leafletjs.com/>) (参照 2017-04-24).
- [3] 小関 章太郎, 高橋 直久, focus+context+glue マップの glue 領域における道路ネットワーク総描法, 第 6 回情報科学技術フォーラム一般講演論文集, D-025, pp.61-62, 2007/8.
- [4] Naohisa Takahashi, An Elastic Map System with Cognitive Map-based Operations, International Perspectives

- on Maps and the Internet, Springer-Verlag, pp.73-87, Feb. 12, 2008.
- [5] Daisuke Yamamoto, Shotaro Ozeki, Naohisa Takahashi, Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services, Proceedings of the ACM SIGSPATIALGIS 2009, Seattle, Washington, pp.101-110, 2009/11.
- [6] 山本 大介, 小関 章太郎, 高橋 直久, 道なり道路選別手法に基づく Fisheye View マップ総描手法, 電子情報通信学会論文誌. D, 情報・システム, Vol.J93-D, No.10, pp.1914-1925, 2010/10.
- [7] 大橋洋介, 山本大介, 高橋直久, "OpenStreetMap とイラストマップを用いた Focus+Context マップ描画手法," マルチメディア, 分散, 協調とモバイル (DICOMO2015) シンポジウム 論文集, pp.1619-1625, Jul.2015.
- [8] 村瀬正樹, 山本大介, 高橋直久, "Focus+Glue+Context マップのためのストロークに基づく道路ネットワーク総描システム," 第 8 回データ工学と情報マネジメントに関するフォーラム (DEIM2016), H7-3, Mar.2016.
- [9] YOLP(地図) - Yahoo!デベロッパネットワーク, 入手先 (<http://developer.yahoo.co.jp/webapi/map/>) (参照 2017-04-24).
- [10] jQuery, 入手先 (<https://jquery.com/>) (参照 2017-04-24).
- [11] jQuery.mmenu, app look-alike menus with sliding sub-menus, 入手先 (<http://mmenu.frebsite.nl/>) (参照 2017-04-24).
- [12] GoogleMaps, 入手先 (<https://www.google.co.jp/maps/>) (参照 2017-04-24).
- [13] Yahoo!地図 - 地図検索・雨雲レーダー搭載の多機能マップ, 入手先 (<http://map.yahoo.co.jp/>) (参照 2017-04-24).
- [14] MapFan (マップファン) - 地図・ルート検索・観光情報, 入手先 (<http://www.mapfan.com/>) (参照 2017-04-24).
- [15] 地図利用実態調査 | ゼンリン, 入手先 (<http://www.zenrin.co.jp/dl/map/16.html>) (参照 2017-04-24).
- [16] 谷口 和也, 高橋 直久, focus+context+glue 型モバイルマップにおけるランドマークの動的表示制御法, 第 6 回情報科学技術フォーラム一般講演論文集, J-010, pp.409-410, 2007/8.
- [17] 加藤 史也, 高橋 直久, 山本 大介, Focus+Glue+Context 型マップのための凹型多角形の Focus 生成手法, 全国大会講演論文集, Vol.2012, No.1, pp.677-679, 2012/3.
- [18] 加藤 史也, 山本 大介, 高橋 直久, 任意形状 Focus 生成機能を有する Focus+Glue+Context マップシステムの実現, 電子情報通信学会技術研究報告. DE, データ工学, Vol.112, No.346, pp.119-124, 2012/12.
- [19] H. Mizutani, D. Yamamoto, and N. Takahashi, "A Fusion of Multiple Focuses on a Focus+Glue+Context Map", Proceedings of the International Conference on Intelligent Interactive Multimedia Systems and Services (IIMSS 2012), pp.11-21, 2012/5.
- [20] 細井純太, 山本大介, 高橋直久, "フリック入力型マップ操作機能を有するモバイル EMMA," 第 6 回データ工学と情報マネジメントに関するフォーラム, E3-2, Mar.2014.