

信号の周期を利用した最速のマップ案内

篠原 一志 ストリャロフ ティモフェーイ*1

担当教員：菅野一喜

Email: Kazuki_Kanno@education.metro.tokyo.jp

*1: 東京都立科学技術高等学校

◎Key Words

信号機、マップ、プログラム

1. はじめに

現在、多くの地図アプリや経路案内サービスでは、目的地までの距離や推定移動時間をもとに経路が提示される。徒歩移動においても、経路の距離や道路形状をもとに所要時間が推定されることが多い。

しかし、歩行者の場合、実際の移動時間には交差点での信号待ち時間が影響する。特に、交差点が多い地域や大通りを横断する経路では、距離が短い経路であっても、信号待ちによって到着時間が長くなることがある。交通信号制御は交通流や待ち時間に関わる要素として扱われており^[12]、歩行者向けの経路案内においても信号待ちを考慮する意義がある。

そこで本研究では、Web 地図上に歩行者信号や横断歩道の情報を表示し、取得した徒歩ルートに対して、ルート付近に存在する信号数と推定待ち時間を加算表示する試作システムを開発する。

2. 目的

本研究の目的は、歩行者信号の待ち時間を考慮した経路案内マップを開発することである。具体的には、出発地と目的地を指定すると徒歩ルートを取得し、そのルート付近の信号情報を抽出して、通常の徒歩時間に推定信号待ち時間を加算表示する。

現段階では、実際の歩行者信号の秒数はまだ取得できていない。また、実際の経路比較や、信号待ちを考慮した複数経路の比較もまだ実施していない。そのため、本稿では、現在の実装で行っている信号情報の取得、可視化、および仮の信号周期にもとづく待ち時間加算表示について述べる。

3. 用語と前提

本システムは、Web 技術、地図データ、経路探索サービス、信号情報取得を組み合わせて動作する。初見でも構成を追えるように、本稿で使用する主な用語を表 1 に示す。

表 1 本稿で使用する主な用語

用語	説明
Web ブラウザ	Web ページや Web アプリを表示・操作するソフト。Chrome や Edge など。
Web アプリ	インストール専用アプリではなく、ブラウザ上で動作するアプリ。

用語	説明
フロントエンド	利用者が画面上で操作する部分。入力欄、ボタン、地図表示などを含む。
React	画面部品を組み合わせて Web アプリを作る JavaScript ライブラリ ^[3] 。
JavaScript	ブラウザ上で動くプログラミング言語。地図操作やボタン処理に使う。
ライブラリ	一から作らずに利用できる機能のまとまり。
Leaflet	Web 地図を表示・操作するための JavaScript ライブラリ ^[2] 。
React Leaflet	React 上で Leaflet を扱うためのライブラリ ^[4] 。
OpenStreetMap	道路、建物、信号機などの地物情報を持つ地図データベース ^[1] 。
OSM Wiki	OpenStreetMap で使うタグやデータの意味を説明する文書群。
地物	地図上の対象物。道路、横断歩道、信号機、建物など。
API	外部サービスからデータを取得するための接続口。
Nominatim	地名や住所を緯度経度に変換する OpenStreetMap 系の検索サービス ^[5] 。
ジオコーディング	「西大島」などの地名を緯度経度に変換する処理。
緯度経度	地球上の位置を数値で表す座標。地図表示や経路取得に用いる。
Geolocation API	利用者の許可を得て端末の現在地を取得する Web API ^[11] 。
OpenRouteService	徒歩などの経路を取得する外部経路探索サービス ^[9] 。
OSRM	経路探索用のサービス。OpenRouteService が使えない場合の代替として用いる ^[10] 。
Overpass API	OpenStreetMap から条件に合う地物を取得するための API ^[6] 。
タグ	OSM の地物に付く属性。例として highway=traffic_signals などがある。
highway=traffic_signals	OSM で信号機を表すタグ ^[7] 。
crossing=traffic_signals	信号制御された横断箇所を表すタグ ^[8] 。
highway=crossing	横断歩道などの横断箇所を表すタグ。
徒歩ルート	出発地から目的地まで歩いて移動する経路。
ポリライン	地図上で経路を線として表示するための折れ線。
信号群	同じ交差点にある可能性が高い近接信号をまとめた単位。
信号周期	赤、青、黄が一巡する時間。現段階では仮の秒数を使う。
推定待ち時間	信号状態から計算した仮の待ち時間。実測値ではない。
信号込み表示時間	通常の徒歩時間に推定待ち時間を加えた表示用の時間。
重み付きグラフ	距離や待ち時間などの重みを持つ点と線の構造。将来の経路選択で利用候補となる。

4. システム構成

本システムは、利用者が操作するフロントエンドと、外部サービスから取得する地図・経路・信号データによって構成される。利用者はブラウザ上で出発地と目的地を入力し、システムは地名検索、徒歩ルート取得、信号情報取得、推定待ち時間計算を順に行う。

地図表示には Leaflet および React Leaflet を用いる。地図データには OpenStreetMap を利用する。OpenStreetMap のデー

タ利用時には、ライセンスおよびクレジット表示に注意する必要がある⁽¹⁾。

出発地と目的地の検索には Nominatim の Search API を用いる⁽⁵⁾。徒歩ルートの取得には OpenRouteService を用い、取得できない場合には OSRM public server を用いる。信号情報については、Overpass API を用いて出発地周辺の highway=traffic_signals , crossing=traffic_signals , highway=crossing などの地物を取得する。

表2 システムを構成する要素

構成要素	役割
入力部	出発地と目的地を指定する。現在地取得または地名検索を利用する。
地図表示部	OpenStreetMap の地図上にピン、徒歩ルート、信号マーカを表示する。
経路取得部	OpenRouteService または OSRM から徒歩ルート、距離、通常時間を取得する。
信号取得部	Overpass API を用いて周辺の信号機・横断歩道情報を取得する。
信号処理部	信号の分類、信号群の作成、ルート付近判定、推定待ち時間の計算を行う。
表示部	通常時間、信号数、推定信号待ち時間、信号込み表示時間を表示する。

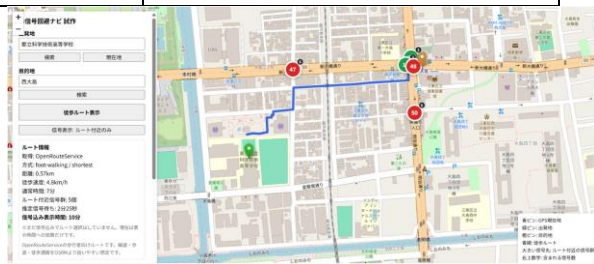


図1 試作システムの表示画面

5. 実装方法

本システムの処理は、入力、経路取得、信号取得、信号処理、表示の順に進む。まず利用者が出発地と目的地を指定する。出発地はブラウザの位置情報取得機能による現在地、または地名検索によって指定できる。目的地は Nominatim による地名検索によって指定する。

次に、指定された出発地と目的地をもとに徒歩ルートを取得し、取得したルートを地図上にポリラインとして表示する。さらに、取得したルートの座標列をもとに、ルート付近に存在する信号を抽出する。近接する複数の信号は、同じ交差点や横断箇所に対応する可能性があるため、一定距離以内の信号をまとめて信号群として扱う。

現段階では、実際の信号秒数を取得できていないため、赤信号、青信号、黄信号に仮の秒数を設定している。推定待ち時間は、赤信号の場合は残り赤信号時間、青信号の場合は0秒、黄信号の場合は赤信号時間の半分として計算する。算出した推定信号待ち時間を通常の徒歩時間に加算し、信号

込み表示時間として表示する。

表3 システムの処理手順

手順	処理内容
1	出発地と目的地を指定する。
2	地名検索または現在地取得によって緯度経度を得る。
3	徒歩ルートを取得し、地図上に表示する。
4	出発地周辺の信号機・横断歩道データを取得する。
5	ルート付近の信号群を抽出する。
6	仮の信号周期から推定待ち時間を計算する。
7	通常時間に推定待ち時間を加え、信号込み表示時間として示す。

6. 現在の実装状況

現在のシステムでは、出発地と目的地を指定し、徒歩ルートを地図上に表示できる。また、出発地周辺の信号機や横断歩道の情報を OpenStreetMap から取得し、地図上に表示できる。

さらに、取得した徒歩ルートの付近に存在する信号群を抽出し、仮の信号周期にもとづいて推定待ち時間を算出する機能を実装した。これにより、通常の徒歩時間に加えて、信号待ちを含めた表示時間を確認できる。今後の実測に向けて、信号計測用機材も準備している。図2に機材の構成と設置例を示す。



図2 信号計測用機材 (a) 構成機材, (b) 設置例

一方で、現在表示される信号待ち時間は実測値ではなく、仮の信号周期にもとづく推定値である。また、複数の経路候補を比較し、信号待ちが少ない経路を自動的に選択する機能は未実装である。したがって、現段階で本システムが行っていることは、信号待ちを考慮した最適経路の提示ではなく、取得した徒歩ルートに対する信号情報の可視化と、推定信号待ち時間の加算表示である。

7. 今後の課題

第一に、実際の歩行者信号の赤信号時間と青信号時間を現地調査によって測定し、現在用いている仮の信号周期を実測データに置き換える必要がある。これにより、推定待ち時間の精度を高めることができる。

第二に、複数の候補経路を取得し、それぞれの距離、信号数、推定待ち時間を比較できるようにする必要がある。現段階では、取得済みの徒歩ルートに対して信号待ち時間を加算しているだけであり、信号待ちを考慮した経路選択までは行っていない。

第三に、実際に歩行して測定した所要時間と、システムが算出した推定所要時間を比較し、推定の妥当性を検証する必要がある。将来的に信号待ち時間を重みとして経路選択を行う場合には、重み付きグラフの最短経路探索手法の利用も検討できる⁽¹³⁾。

8. おわりに

本研究では、歩行者信号の待ち時間を考慮した経路案内マップの試作システムを開発した。現在の実装では、Web 地図上に徒歩ルートと信号情報を表示し、ルート付近の信号に対して仮の信号周期にもとづく推定待ち時間を算出できる。

現段階では、実際の信号周期データの取得、複数経路の比較、信号待ちを考慮した経路選択は未実装である。今後は、実測した信号データを反映し、複数経路の比較と実歩行による検証を行うことで、歩行者にとってより実際の移動時間に近い経路案内の実現を目指す。

参考文献

- (1) OpenStreetMap contributors : "Copyright and License", OpenStreetMap, <https://www.openstreetmap.org/copyright> (参照 2026-06-30) .
- (2) Leaflet : "Leaflet API reference", Leaflet Documentation, <https://leafletjs.com/reference.html> (参照 2026-06-30) .
- (3) Meta Open Source : "React", React Documentation, <https://react.dev/> (参照 2026-06-30) .
- (4) Paul Le Cam and contributors : "React Leaflet: Introduction", React Leaflet Documentation, <https://react-leaflet.js.org/docs/start-introduction/> (参照 2026-06-30) .
- (5) Nominatim : "Search", Nominatim Manual, <https://nominatim.org/release-docs/latest/api/Search/> (参照 2026-06-30) .
- (6) OpenStreetMap Wiki contributors : "Overpass API", OpenStreetMap Wiki, https://wiki.openstreetmap.org/wiki/Overpass_API (参照 2026-06-30) .
- (7) OpenStreetMap Wiki contributors : "Tag: highway=traffic_signals", OpenStreetMap Wiki, https://wiki.openstreetmap.org/wiki/Tag:highway=traffic_signals (参照 2026-06-30) .
- (8) OpenStreetMap Wiki contributors : "Tag: crossing=traffic_signals", OpenStreetMap Wiki, https://wiki.openstreetmap.org/wiki/Tag:crossing=traffic_signals (参照 2026-06-30) .
- (9) openrouteservice : "Directions Service", openrouteservice backend documentation, [https://gis.science.github.io/openrouteservice/api-](https://gis.science.github.io/openrouteservice/api-reference/endpoints/directions/)

[reference/endpoints/directions/](https://gis.science.github.io/openrouteservice/api-reference/endpoints/directions/) (参照 2026-06-30) .

(10) Project OSRM : "OSRM API Documentation", Project OSRM, <https://project-osrm.org/docs/v5.24.0/api/> (参照 2026-06-30) .

(11) MDN Web Docs : "Geolocation API", Mozilla Developer Network, https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API (参照 2026-06-30) .

(12) Wei, H., Zheng, G., Gayah, V., Li, Z. : "A Survey on Traffic Signal Control Methods", arXiv:1904.08117 (2019) .

(13) Dijkstra, E. W. : "A note on two problems in connexion with graphs", Numerische Mathematik, Vol.1, pp.269-271 (1959) .