

# ユーザの体感と歩行履歴情報を用いた移動時間推定システム

宮本健太<sup>†</sup> 梅津高朗<sup>†</sup>

**概要:** 近年, Google Mapsをはじめとした地図アプリケーションなどの目的地までの経路と移動時間を検索するシステムが登場し, さらに一般にスマートフォンが普及したことでこれらのシステムを利用する機会が増えた. しかし, 徒歩移動に関わる時間は, 過去に歩いたことのある経路であっても, 体調や感情, 体感気温などといったユーザの身体状況と感覚や勾配などの歩行経路の環境によって大きく変化する可能性がある. ユーザが実際に移動する際の環境に合わせた移動時間を推定するためには, これらの原因を考慮する移動時間推定手法が必要である. 本研究の目的は, 実際の歩行者であるユーザが, 自身の体調や気分などを考慮した上で, 目的地までの歩行移動時間を精度高く推定できるようにすることである. 本稿では, 一般的な歩行者向けナビゲーションシステムで平均的な到着時間が予測できているとみなし, ユーザの体感を選択肢中から選んで申告する UI を介して得た情報から, 個々人の体力やその時々で発生する移動時間のずれを推定するアプローチを採った. 著者自身を被験者として収集した 41 件の歩行履歴情報を用いてモデルを作成したところ, 決定係数 0.973 の説明力が高いモデル式を算出できた. 作成したモデルを用いて, 2 種類の経路における徒歩移動時間の推定を行った結果, モデル作成時に使用した経路では推定誤差が 9 秒に収まり, Google Maps の推定結果よりも推定誤差が 7 秒小さかった. また, モデル作成時に使用しなかった経路では, 推定誤差が 12 秒となり, Google Maps よりも推定誤差が 20 秒小さかった. 最終的に提案手法を用いることによってユーザによる歩行速度のずれをモデル式で表現し, 有用な推定徒歩移動時間を算出できる可能性を示せた.

**キーワード:** 歩行履歴情報, GPS

## Estimating Walking Travel Time Using User's Physical Feeling and Walking History

KENTA MIYAMOTO<sup>†</sup> TAKA AKI UMEDU<sup>†</sup>

**Keywords:** Walking History, GPS

### 1. はじめに

近年, 歩行者の移動を補助するため, 歩行者向けのナビゲーションシステムの研究が多く行われてきた[1]. これらの研究により, Google マップ[2]をはじめとした地図アプリなど, 目的地までの経路を検索するシステムが登場し, 特に最近ではスマートフォンが普及したことで, これらのシステムを利用する機会が増えた.

例えば, 自宅から大学までの経路を検索した場合, 地図アプリからは目的地までの公共交通機関の乗車時間と徒歩移動にかかる時間を合わせた情報が得られる. ここから得られる公共交通機関の乗車時間は, 様々な原因を考慮しても, 情報通りになる場合が多い.

しかし, 徒歩移動にかかる時間は, 過去に歩いたことのある経路であっても, ユーザの身体状況と感覚(体調や感情, 体感気温など)や歩行経路の環境(天候, 気温, 勾配など)によって変化する可能性があるため, 経路検索によって算出された徒歩移動の予想移動時間よりも早く到着したり, 遅く到着したりすることが考えられる. ユーザが実際に移動する際の環境に合わせた移動時間を推定するためには, これらの原因を考慮する移動時間推定手法が必要である.

本稿では, 上記に示したような体調や感情, 体感気温などのユーザ申告情報と実際の移動で得られた経路の情報や移動時間を記録し, モデル化することによって, それぞれのユーザが普段歩く経路に対して, ユーザに合わせた移動時間を推定するシステムを提案し, そのシステムによる実験結果を述べる.

### 2. 関連研究

本章では, 関連した研究を紹介する.

#### 2.1 歩行速度を調節するナビゲーション

藤沢ら[3]は, 歩行のペース, すなわち歩行速度を調節する点に着目し, ユーザの指定した時間に目的地へ到着できるように支援するシステムを提案している. 通常, 目的地までの残りの距離と, そこまでの移動時間を歩行中に見積もることは困難であり, 見当をつけたとしても, 時間通りに到着できるよう適切に歩行速度を調節することは困難である. そこで, この研究では StepNavi と呼ばれる歩行速度ナビゲーションシステムを開発している. このシステムでは, 端末上にユーザの歩行速度と適切な歩行速度が相対的に表示されるグラデーション UI によって, 視覚的にユーザが適切な歩行速度を理解できるようにしている. しかし, 実際に歩行する経路の環境やユーザの身体状況と感覚によっては, システムが指示する歩行速度を維持すること

<sup>†</sup> 滋賀大学  
Shiga University

が難しいと考えられる。本稿では、それらの要因を考慮することによって、ユーザに歩行速度ではなく、出発時間の調整や徒歩以外の適切な移動手段が選択できるようになることを期待している。

## 2.2 特定の要因による歩行速度の変化に着目した研究

特定の要因と歩行速度の変化に着目した研究に、松本ら[4]の文献がある。歩行速度が変化する要因を考える際に、地域性や歩行者ごとの特徴以外に「街路空間の“魅力”」が影響しているのではないかとという視点からアプローチしている。特徴別に街路の魅力を得点化して分析し、歩行速度が減少するに連れて街路空間の魅力が大きくなるという結果が示されている。この研究では、歩行速度の変化に関わる要因を数値化し、歩行速度とともに単回帰分析を行っている。本稿では、後述のユーザから得た自己申告情報を点数化することにより、歩行移動時間推定モデルの作成時に用いる重回帰分析の説明変数として利用できるようにする。

## 2.3 歩行履歴情報から歩行ナビゲーションに必要な要素を分析する手法

歩行履歴情報を活用した研究に、白川らの研究[5]がある。ユーザから収集した歩行履歴情報によって、歩行ナビゲーションに必要な情報を自動的に分析する手法を提案している。道路の斜度や建物の入口の位置、歩行者用信号機の所在や横断歩道の位置は、一般に地図には記載されていない。そこで、位置情報や高度、それを記録した時間をまとめた歩行履歴情報を分析して、それらの位置を推定している。この研究では、歩行履歴情報の取得に、GARMIN[6]の eTrex Summit が使用されているため、一部のスマートフォンに搭載されていない気圧高度計の値が使用されている。本稿では、機種を問わずに搭載されている GPS のみで利用できることを重視しているため、外部サービスによって歩行履歴情報に高度情報を付加している。

## 2.4 歩行履歴情報から移動時間を推定するシステム

夏堀ら[7]は、本稿と同様にユーザの歩行履歴情報を用いた移動時間を推定するシステムを提案している。ここでは、位置情報と心拍数を計測するために、腕時計型の専用端末である、Pianta[8]の GPS-22HRW+ II を使用し、ここから得られるデータと、そのデータを取得した時刻を蓄積し、そこから歩行速度と特徴量を算出している。そして、歩行予定経路の周辺環境と類似した歩行者ログデータを使用して、その移動速度と距離から移動時間を推定する。この研究では、専用端末と道路情報データベースから得られた情報から推定しているが、本稿では、ユーザによる身体状況や感覚の自己申告情報を移動時間推定時に利用する。

歩行者の歩行移動時間を推定するためには、位置情報履歴から歩行速度と斜度を算出することと、歩行に関わる情報を管理し、関連付けたうえで回帰分析することが必要だと考えられる。

## 3. 研究目的とアプローチ

### 3.1 研究目的

本研究の目的は、実際の歩行者であるユーザが、自身の体調や気分などを考慮した上で、目的地までの歩行移動時間を精度高く推定できるようにすることである。本稿では、一般的な歩行者向けナビゲーションシステムで平均的な到着時間が予測できているとみなし、その予想から、個々人の体力やその時々状況によるずれを推定するアプローチを採る。そのため、それぞれのユーザが普段から何度も歩くような道を対象に、主にその道を歩いた本人のデータを用いて、その人に特化した推定を行うことを目指す。移動時間の増減に直接関わる歩行速度は、体調や身体感覚、気温や湿度、服装や所持品の重さなど、様々な要因によって変化することが考えられる。しかし、それらの要因をすべて記録することは難しいため、ここではユーザの体感がそれらの環境要因をある程度反映していると考え、ユーザが選択肢中から選んで申告するユーザーインターフェースを介して情報を集めることとした。

提案手法の応用としては、歩行者ナビゲーションシステムに適用できるほか、信号の切り替えタイミングなども推定し、それらも考慮した経路の最適化、そしてユーザの歩行速度に合わせた観光順路マップの提案などが考えられる。

### 3.2 アプローチ

3.1 節で示した目的を実現するために、1) ユーザの歩行履歴と自己申告情報を取得し、2) 歩行履歴・自己申告情報記録から経路の周辺環境情報を取得した後、3) 取得した情報を管理し、モデルを作成することが必要だと考えられる。

1) は、スマートフォンを用いて、ユーザが普段歩く道を対象に搭載されている GPS から得られる位置情報と、位置情報を取得した時間、ユーザが自己申告した身体状況と感覚(体調や感情、体感気温)を取得する。なお、機種によっては搭載されているセンサにより気温や湿度、気圧などを取得できる場合もあるが、すべての機種に搭載されている機能ではないため、ここでは使用しない。

2) は、位置情報を取得した位置の高度と 1) で記録した経路区間の歩行時間を別に推定した場合の情報を外部から取得し、目的実現のための材料とする。

3) は、1)、2) で取得したデータを歩行履歴情報として定義し、管理を行う。ここで管理しているデータを用いて歩行距離の算出や位置座標間の高低差・歩行時間の算出を行い、モデル作成をする。

## 4. 提案手法

### 4.1 提案手法の概要

このシステムは、図1のような、歩行履歴情報の収集と図2のような歩行移動時間の推定の2つの機能からなる。本節では、それぞれの機能について紹介する。ここでは、到着予想時間の一般的な歩行者向けナビゲーションシステム

ムの推定値からのずれは、到着時間に影響のある要因の線形結合である程度表現できると考え、重回帰分析を用いて行う。

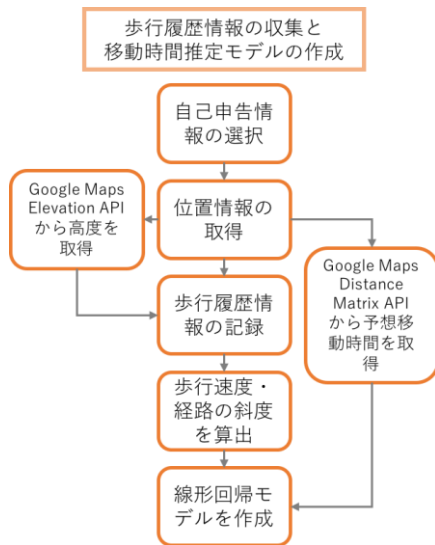


図 1 歩行履歴情報の収集と歩行移動時間推定モデルの作成

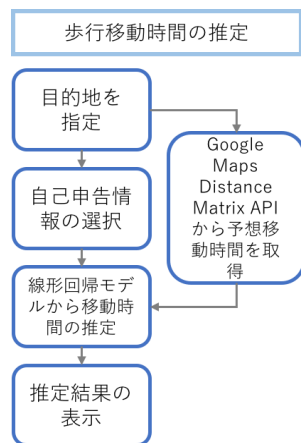


図 2 歩行移動時間の推定

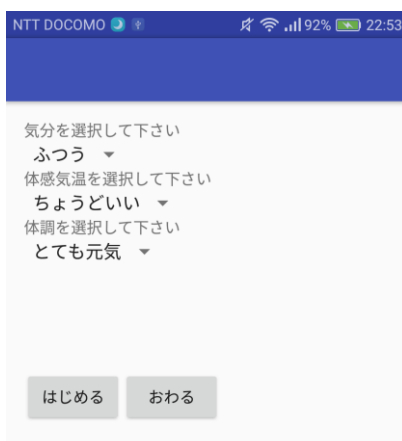


図 3 ユーザ自己申告情報入力画面例

表 1 ユーザ申告情報選択肢一覧

体調	気分	体感気温
とても元気(2)	良い(2)	暑い(-1)
元気(1)	楽しい(1)	快適(1)
普通(0)	普通(0)	寒い(-1)
疲れている(-1)	暗い(-1)	
崩している(-2)		

表 2 位置情報データテーブルのデータ構造

属性名	説明
体調	表 1 からユーザが選択した項目
気分	同上
体感気温	同上
時刻	位置情報データを取得した時刻 (UNIX 時間)
位置情報	GPS センサが取得した位置情報 (緯度・経度)
高度	Google Maps Elevation API から取得した高度

#### 4.1.1 歩行履歴情報の収集と移動時間推定モデルの作成

ユーザの歩行移動時間を推定する際に必要な情報を蓄積し、歩行移動時間推定モデルを作成する歩行履歴学習部について説明する。

歩行履歴情報を取得開始する前に、図 3 のような画面で自己申告情報をユーザに登録してもらう。表 1 は、この画面で表示する自己申告情報の選択肢を示しており、各項目に付している数字は、後ほど線形回帰を行う際の説明変数になる。体調と気分の説明変数は、「普通」を中心に、歩行速度の増加に係ると考えられるものは正の数を、減少に係ると考えられるものには負の数を付している。体感気温は、「快適」の場合に歩行速度が増加、その他の場合には減少すると考えて、それぞれの数字を付している。なお、「寒い」ときに歩行速度が速くなるとも考えられるが、気温が低い場合には歩行経路上が凍っていたり、積雪していたりすることが考えられるため、ここでは歩行速度が減少する状態として設定した。他にも、歩行距離が伸びるに連れて体が温まり、体感気温が「快適」に変化するなど、歩行履歴の記録中に自己申告情報が変化する可能性が考えられるが、本稿では考慮しない。

自己申告情報の選択後、「はじめる」をタップすると、自己申告情報と関連付けた歩行履歴情報の取得を始める。歩行終了後、「おわる」をタップすると、歩行履歴情報の取得が終了し、Google Maps Elevation API から各記録地点の高度情報を取得する。同時に、Google Maps Distance Matrix API から、歩行履歴情報を取得した経路をナビゲーションした場合の到着予想時間を取得する。このとき Google Maps Distance Matrix API には歩行履歴情報の記録を開始した位置座標と終了した位置座標を送信する。

これらの収集した情報の蓄積や管理は、表 2 の位置情報

データテーブルのデータ構造で示した形で行う。なお、個人ごとにパラメータが異なると思われることとプライバシー保護の観点により、スマートフォンから得た位置情報は、基本的に端末内部で記録・管理し、外部との通信は Google Maps の各種 API から情報を取得する場合のみに利用することを想定している。

次に、移動時間推定モデルを作成するため、位置情報データテーブルのデータを表 3 の歩行距離・時間計算データテーブルに変換する。表 3 の「距離」の項目の算出には緯度・経度の情報が記録されている表 2 の「位置情報」を用いている。計算方法は国土地理院のウェブサイトにある情報[9]を参考にした。また、位置情報データテーブルには位置情報の記録時の時刻を UNIX 時間で記録しているため、移動時間を算出する際連続したレコードの「時刻」項目の差をとることによって移動時間を算出している。

最後に、表 4 の変数一覧に沿った形式に変換し、目的変数に実際の移動時間、説明変数に各歩行履歴情報をあてはめて重回帰分析を行う。例えば線形回帰を行う場合、式 1 のモデルに表 4 の各変数をあてはめる。それぞれの目的変数の偏回帰係数を調整することで、ユーザに合った徒歩移動時間推定モデルを作成する。歩行履歴情報を蓄積するたびに各係数は更新される。

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_6 \quad (1)$$

$a$  : 定数

なお、表 4 の「上り坂の割合」「下り坂の割合」の各変数は、位置情報データテーブルで連続している 2 点間の位置情報と高度から求めている。式 2 のように、2 点間の高度の差を求め、その値を正の場合と負の場合で分ける。そしてそれぞれの総和を全経路の距離で除する。

$$d_n = h_{n+1} - h_n$$

$$d_n > 0 \text{ の場合 } \quad a_n = p_n$$

$$d_n < 0 \text{ の場合 } \quad b_n = p_n$$

$$\text{その他の場合} \quad a_n = b_n = 0$$

$$x_5 = \frac{1}{dist} \sum_{k=1}^n a_k \quad (2)$$

$$x_6 = \frac{1}{dist} \sum_{k=1}^n b_k$$

$h_n$  : 高度     $p_n$  : 2点間の移動距離  
 $dist$  : 全経路の距離

表 3 歩行距離・時間計算データテーブル

項目	説明
距離 (m)	位置情報テーブルの連続したレコードのうち「位置情報」を用いて算出したもの
高低差 (m)	位置情報テーブルの連続したレコードのうち「高度」の差をとったもの
移動時間 (秒)	位置情報テーブルの連続したレコードのうち「時刻」の差をとったもの

表 4 変数一覧

変数	説明
$y$ (実移動時間)	歩行履歴情報取得時に計測した、実際の移動時間
$x_1$ (予想移動時間)	Google Maps Distance Matrix API から取得した予想移動時間
$x_2$ (体調)	表 2 に記録されている「体調」の選択を表 1 の値で示したもの
$x_3$ (気分)	表 2 に記録されている「気分」の選択を表 1 の値で示したもの
$x_4$ (体感気温)	表 2 に記録されている「体感気温」の選択を表 1 の値で示したもの
$x_5$ (上り坂の割合)	経路上で勾配が正の値をとった 2 点間の距離の総和を全経路の距離で割ったもの
$x_6$ (下り坂の割合)	経路上で勾配が負の値をとった 2 点間の距離の総和を全経路の距離で割ったもの

#### 4.1.2 歩行移動時間の推定

前節で作成したモデルを用いて、歩行移動時間を推定する。はじめに、ユーザに目的地を指定してもらう。その後、Google Distance Matrix API から目的地までの経路の到着予想時間を取得し、同時にユーザには経路探索時点での自己申告情報を登録してもらう。最後に、作成してあるモデルの説明変数にそれぞれ自己申告情報の選択肢に応じた点数を当てはめ、実際の移動時間を推定する。推定した経路においても位置情報の記録を行い、次回以降の歩行移動時間推定に使用する。

### 5. 評価実験

本章では、4 章で示した提案手法の有用性を検討するために行った実験とその実験結果について述べる。

歩行履歴の記録には、iPhone 向けアプリケーションの ZweiteGPS を使用し、3 秒毎に 1m 進んだ場合に位置情報を記録する設定にした。そしてそれらの記録を JSON 形式で出力した。なお、iPhone6 以降では気圧計によって、相対高度を算出することができ、ZweiteGPS においても位置情

報の記録時にその結果を記録するが、3.2 節で述べたとおり、本稿は気圧計を搭載していない旧型の iPhone などでも利用できるようにすることを目標としているため、今回は使用せずに Google Maps Elevation API から取得した高度を使用している。

次に、本実験では Python を用いて ZweiteGPS から取得した JSON ファイルを処理する次の機能を実装した。データベース管理システム (DBMS) には、SQLite を使用した。

- I. ZweiteGPS から取得した JSON ファイルから緯度と経度、そして UNIX 時間を抜き出し、データベースに記録する。記録する緯度と経度の情報を用いて Google Maps Elevation API からその位置座標の高度を取得する。
- II. I. で記録したデータベースにある連続したレコードを用いて、それらの緯度と経度から距離を算出する。同時に、UNIX 時間の差を求めて移動時間を算出し、高度の差を取って地点ごとの高低差を算出する。これらの情報を新たにデータベースに記録する。
- III. II. で記録したデータベースから式 2 を元に上り坂・下り坂の割合を算出する。なお、式 2 の  $d_n$  は II. で算出と記録をしているため、この情報を使用する。また、総移動時間と総移動距離の算出も行う。I. で記録されている位置情報のうち、最初と最後のレコードから緯度と経度の情報を抜き出し、それを Google Maps Distance Matrix API に送信する。ここから、Google Maps が推定した歩行移動時間が取得できる。最終的にデータベースに新たなテーブルを作成し、これらの結果を記録する。その際、点数に変換したユーザ自己申告情報と紐付ける。

I. から III. までの処理を ZweiteGPS から取得した JSON ファイルすべてに対して適用する。最終的に、III. で作成したデータベースのテーブルを CSV ファイルとしてエクスポートし、統計解析ソフトの R を用いて重回帰分析を行った。出力された結果を元に、次は提案手法による推定部を Python で実装した。実装した機能は以下のとおりである。

- i. 出発地点と到着地点の緯度・経度、そしてユーザ自己申告情報を数字で入力する。なお緯度・経度は、Google Maps に表示されている地図をクリックした際に表示される緯度・経度の情報を用いている。
- ii. 出発地点と到着地点の緯度・経度情報から、Google Maps Distance Matrix API より Google Maps が推定した移動時間を取得する。
- iii. i. と ii. で取得した情報を元に、重回帰分析によって作成されたモデル式に代入する。

なお、推定後に行う歩行移動時間の測定は、ZweiteGPS を

用いた。

### 5.1 実験環境

被験者は著者自身 (20 代男性) とし、2016 年 5 月 11 日から 12 月 14 日までの期間に収集した 41 件の歩行履歴と自己申告情報の記録を歩行履歴情報の一部とした。収集した 41 件のうち、37 件は著者の自宅から最寄りのバス停まで住宅街を通る約 600m の経路を歩行した記録であり、3 件は在籍している大学と駅との間の約 1900m の経路、1 件は上り坂を中心としている約 1750m の経路である。なお、自宅から最寄りのバス停までの経路は住宅街を通っているため、信号や横断歩道などの歩行を制限する要素は含まれていない。

### 5.2 移動時間推定モデル作成

章のはじめに述べたとおり、前節で示した歩行履歴を含む JSON ファイルすべてを処理し、それを CSV ファイルとして R にインポートしてから重回帰分析を行った。この実験では式 1 で示した形式でモデルを作成することとし、式 3 に実際に作成したモデル式を示す。また、表 5 には分析結果を示している。なお、決定係数は 0.973 であり、自由度調整済み決定係数は 0.968 であった。

$$y = 2340.746 + 1.172x_1 - 11.813x_2 - 38.271x_3 + 6.866x_4 - 2253.404x_5 - 2515.398x_6 \quad (3)$$

表 5 重回帰分析結果

	係数	標準誤差	t 値	P-値
切片	2340.74 6304	1394.52	1.678532	0.102412912
予想移動時間 ( $x_1$ )	1.17208 2913	0.041757	28.06914	4.21785E-25
体調 ( $x_2$ )	-11.812 52771	10.58542	-1.11592	0.272279687
気分 ( $x_3$ )	-38.270 94647	13.96338	-2.74081	0.009696785
体感気温 ( $x_4$ )	6.86590 7517	9.855922	0.696628	0.490771504
上り坂の割合 ( $x_5$ )	-2253.4 04257	1480.873	-1.52167	0.137337957
下り坂の割合 ( $x_6$ )	-2515.3 98151	1321.336	-1.90368	0.065444746

### 5.3 移動時間推定モデルを用いた移動時間推定結果

前節で作成した移動時間推定モデルを用いて、実際に移動時間を推定した。歩行経路として、5.1 節で紹介した 37 件の歩行履歴情報を取得した著者の自宅から最寄りバス停

までの経路と、この経路を全く通らない自宅から最寄り鉄道駅までの経路の2種類を選択した。自宅から最寄りバス停までの経路には信号はないが、自宅から最寄り鉄道駅までの経路には信号と横断歩道が1か所ある。なお、どちらの歩行経路も実際に歩いたあとに、その結果を用いてモデル式の各係数の更新を行っていない。

### 5.3.1 自宅から最寄りバス停までの経路の推定結果

まず、自宅から最寄りバス停までの経路において歩行移動時間の推定を行った。表6に推定結果と推定時に用いた各説明変数に代入した値を示している。なお、移動距離は約530mである。

表6 自宅から最寄りバス停までの推定結果

項目	値	推定誤差
実際の歩行時間	214 秒	—
提案手法による推定時間(y)	205 秒	9 秒
Google Maps Distance Matrix API による推定時間(x <sub>1</sub> )	230 秒	16 秒
体調(x <sub>2</sub> )	0 (普通)	
気分(x <sub>3</sub> )	2 (良い)	
体感気温(x <sub>4</sub> )	1 (快適)	
上り坂の割合(x <sub>5</sub> )	0.685273	
下り坂の割合(x <sub>6</sub> )	0.314727	

上記の表より提案手法による推定誤差は9秒となった。これはGoogle Maps Distance Matrix APIから取得した推定結果よりも誤差が7秒短い結果となった。

### 5.3.2 自宅から最寄り鉄道駅までの経路の推定結果

別の経路として、自宅から最寄り鉄道駅までの歩行移動時間の推定を行った。前項と同様、表7に推定結果と各説明変数に代入した値を示した。なお、移動距離は約910mである。

表7 自宅から最寄り鉄道駅までの推定結果

項目	値	推定誤差
実際の歩行時間	559 秒	—
提案手法による推定時間(y)	571 秒	12 秒
Google Maps Distance Matrix API による推定時間(x <sub>1</sub> )	527 秒	32 秒
体調(x <sub>2</sub> )	0 (普通)	
気分(x <sub>3</sub> )	0 (普通)	
体感気温(x <sub>4</sub> )	1 (快適)	
上り坂の割合(x <sub>5</sub> )	0.461017	
下り坂の割合(x <sub>6</sub> )	0.538983	

上記の表より提案手法による推定誤差は12秒となった。Google Maps Distance Matrix APIから取得した推定結果と

比べて、20秒短い結果となった。

## 5.4 考察

### 5.4.1 自宅から最寄りバス停までの経路

本経路は、5.2節で紹介した移動時間推定モデルを重回帰分析によって算出した際に使用した歩行履歴情報群の一部と同じ経路を通っている。本稿では、Google Mapsから取得した推定移動時間で平均的な到着時間が予測できるとみなしているため、この推定移動時間と提案手法による推定移動時間、そして移動開始時点でユーザから収集した自己申告情報に対応した実移動時間とのそれぞれのずれを測った。

結果として、提案手法による推定時間205秒に対して実際の歩行時間は214秒と、9秒の推定誤差が発生した。また、Google Mapsの推定時間230秒と実際の歩行時間を比べると、16秒の推定誤差が発生し、提案手法に比べて7秒大きく誤差が発生していた。提案手法による移動推定時間は、Google Mapsから取得した推定移動時間に、個々人の体調や気分、体感気温の身体感覚と歩行経路の勾配の割合によって発生する実移動時間とのずれを考慮に入れて推定できたと考えられる。

本結果により、ユーザが普段から何度も歩くような道においては、本システムを用いて有用な推定徒歩移動時間を算出できると考えられる。

### 5.4.2 自宅から最寄り鉄道駅までの経路

本経路は、5.2節で紹介した移動時間推定モデルを算出した際に使用した歩行履歴情報群には含まれない経路である。本稿では、主にそれぞれのユーザが普段から何度も歩くような道を対象に、その道を歩いた本人のデータを取得してそのデータを推定に利用することを想定しているため、モデル作成の際に使用したことのないような経路、すなわちユーザが普段歩かない経路でも正確に推定できるのか評価する必要がある。そのため、この経路を選択した。結果、実際の歩行時間が559秒だったのに対して提案手法による推定時間は571秒と推定誤差が12秒となった。また、Google Mapsによる推定時間527秒とは推定誤差が32秒となり、提案手法よりも20秒大きく誤差が発生していた。

初めて訪れるような経路でも比較的正確に移動時間を推定できた理由として、Google Mapsから得られる推定移動時間と、ユーザ自身の身体感覚や経路中の上り坂・下り坂の割合によって影響された実移動時間とのずれがモデル式によって表現できているからだと考えられる。

本結果により、ユーザが普段歩かない経路においても、本システムを用いることにより有用な推定徒歩移動時間を算出できると考えられる。

## 6. まとめと今後の課題

本研究では、既存の歩行者向けナビゲーションシステムであるGoogle Mapsが推定する歩行移動時間に、ユーザに

よって異なる, 歩行する時点の環境を加味してより精度の高い歩行移動時間を推定するシステムを提案した. ユーザ自身の感覚である「体調」「気分」「体感気温」と歩行する経路の上り坂・下り坂の割合に着目し, これらが要因となって Google Maps が推定する歩行移動時間からずれが発生すると考えた.

これらの要因を加味した場合に, 精度の高い移動時間推定が行えるかを評価するために, GPS から取得した位置情報とそれを取った時間, ユーザに自己申告してもらった体調や感情(気分), 体感気温を収集し, ユーザ自己申告情報とともに重回帰分析を用いて歩行移動時間推定モデルを作成した. 結果, 決定係数 0.973 の説明力が高いモデル式を算出できた.

作成したモデルを用いて, 2 種類の経路における徒歩移動時間の推定は, どちらも Google Maps が提示する推定移動時間よりも誤差が小さい結果となった. 最終的に提案手法を用いることによってユーザによる歩行速度のずれをモデル式で表現し, 有用な推定徒歩移動時間を算出できることがわかった.

しかし, 「あせり」の状態など歩行速度を特別に加速させる条件や地域差による体感気温と歩行速度の不一致, 経路上にある信号機や横断歩道によって発生する待ち時間の考慮などをモデル式に反映できていない. 評価実験時の環境ではこれらの項目を加味する必要がなかったが, 他の環境において使用することを考える場合, 上記のような条件を考慮する必要がある. 今後の課題として, これらの条件をモデル式に反映させる手法を模索することが挙げられる.

以上のことより, 本研究での提案手法は環境条件によって精度の高い歩行移動時間を推定することができると言える.

## 参考文献

- [1] 間邊哲也, 長谷川孝明, 歩行者ナビゲーションコンセプトリファレンスモデルの提案, 電気情報通信学会論文誌, Vol.J95-A, No.3, pp.283-302 (2012).
- [2] Google マップ <https://www.google.co.jp/maps/>
- [3] 藤沢和哉, 安村通晃, StepNavi: 歩行速度ナビゲーションシステムの開発, 情報処理学会インタラクティブシンポジウム論文集, Vol.2012, No.3, pp.307-312 (2012).
- [4] 松本直司, 櫻木耕史, 東美緒, 伊藤美穂: 街路の魅力と歩行速度の関係. 日本建築学会計画系論文集, Vol.77, No.678, pp.1831-1836(2012).
- [5] 白川洋, 歌川由香, 福井良太郎, 重野寛, 岡田謙一, 歩行者ナビゲーションのための歩行履歴情報の分析手法, 情報処理学会研究報告, Vol.2003-MBL-25, pp.69-76 (2003).
- [6] GARMIN <http://www.garmin.com/>
- [7] 夏堀友樹, 白石陽, 歩行者ログを用いた移動所要時間推定システムの提案, マルチメディア, 分散協調とモバイルシンポジウム 2013 論文集, pp.1051-1056 (2013).
- [8] Pianta <http://pianta.ne.jp>
- [9] “経緯度を用いた 2 地点間の測地線長, 方位角を求める計算”, <http://vldb.gsi.go.jp/sokuchi/surveycalc/surveycalc/algorithm/bl2st/bl2st.htm>, (参照 2017-01-13).