

航空写真と地図を使った 歩行者道路ネットワークの自動作成

前田 篤彦^{1,†1,a)}

受付日 2019年4月19日, 採録日 2019年11月7日

概要: 車椅子利用者等の身体が不自由な人を電子的にナビゲートするには、「歩行空間ネットワークデータ」の整備が必要である。歩行空間ネットワークデータとは、歩道や横断歩道をノードとリンクでネットワークとして表現し、各々のリンクに幅員や傾斜等の様々なバリアに関する情報を属性として持たせたものである。しかし、このようなデータは今のところ十分整備されておらず、網羅的な整備には通常の工程をふむと莫大な費用がかかる。そこで本研究では、既存の地理データを活用し、上記データの要となる、歩道と横断歩道のネットワークデータを効率的に自動作成する方法を提案する。加えて、車椅子利用者が通れるかの判断に有用な歩道の幅員も同時に抽出する。提案手法はだまかに3つの工程からなる。(1) 歩道情報の抽出に関しては、専門家が高精細航空写真をトレースする等して作成した電子地図に一般的に歩道の領域が描かれているため、このデータから歩道中心線と幅員を求める。(2) 横断歩道に関しては、それらの位置と向きを航空写真から抽出する。(3) 最後に、抽出した歩道および横断歩道の情報を適切に接続して歩行者道路ネットワークデータを構築する。提案手法を用いて、JR 山手線をカバーする幅員情報付き歩行者道路ネットワークデータを作成し、評価した結果も報告する。

キーワード: 歩道, 横断歩道, 幅員, 歩道ネットワーク, 歩行者道路ネットワーク, 歩行空間ネットワーク, バリアフリーマップ

Automatic Generation of a Pedestrian Road Network Using Aerial Photographs and Maps

ATSUHIKO MAEDA^{1,†1,a)}

Received: April 19, 2019, Accepted: November 7, 2019

Abstract: Electronic navigation is possible for impaired people, i.e., wheelchair users, through a certain “pedestrian space network data,” where sidewalks and crosswalks are presented with nodes and links, and the various attributes related to barriers, such as road widths and slopes, are embedded on each link. However, the creation of such data on a large scale remains ideal at present because the realization of such data would require significant amounts of time and costs. This study introduces an alternative solution, which is an automatic technique for generating the pedestrian road network data. Additionally, this technique is capable of extracting the sidewalk width information, which can be used to determine whether the wheelchair users can pass through sidewalks. In principle, the technique is approximately divided into the following three processes: (1) extraction of the sidewalk centerlines and widths from electronic maps that generally contain the sidewalk polygon data traced from the high-resolution aerial photographs; (2) detection of the crosswalk positions and directions from aerial photographs because they are not traced in the maps; and (3) construction of the road network data through an interconnection of the extracted sidewalk centerlines and the detected crosswalk positions. The proposed technique can be validated with a generated pedestrian road network covering the JR Yamanote Line.

Keywords: sidewalk, crosswalk, road width, pedestrian road network, pedestrian space network, barrier-free map

¹ NTT 未来ねっと研究所
NTT Network Innovation Laboratories, Musashino, Tokyo
180-8585, Japan

^{†1} 現在, NTT サービスエボリューション研究所
Presently with NTT Service Evolution Laboratories

1. はじめに

視覚障がい者や車椅子利用者等の身体が不自由な人を電

^{a)} atsuhiro.maeda.va@hco.ntt.co.jp



- 幅員3m以上歩道
- 幅員2m以上
- 幅員1m以上
- 横断歩道ありリンク
- 横断歩道なしリンク

図 1 作成した幅員情報付き歩行者道路ネットワークの一部

Fig. 1 A part of a generated pedestrian road network data having the sidewalk width information.

子的にナビゲートするには、「歩行空間ネットワークデータ」[1]の整備が必要である。歩行空間ネットワークデータとは、歩道および横断歩道の位置と接続関係をノードとリンクでネットワークとして表現し、各々のリンクに対して、幅員や傾斜等の様々なバリアに関する情報を属性として付与したデータセットを意味する。このようなデータが整備されていれば、たとえば、車椅子利用者をナビゲートする際には、車椅子が通過できる幅があり、かつ車椅子では通過困難な段差や斜面が存在しない経路を抽出して示したり、視覚障がい者には、点字ブロックが充実した経路を探索して音声で案内したりする等、多様な人々に向けた様々なインタラクティブサービスの開発が可能になる。

しかし、今のところ十分な整備がされておらず、網羅的な整備には通常の工程をふむと莫大な費用がかかる。また、この問題に対する取り組みとして、従来研究では、目視でも比較的判別しやすい点字ブロックの有無や路面の大きな凹凸等の情報を複数人のボランティアが協力して収集するものや、歩行者が持つスマートフォンのセンサ等を活用して歩道上の段差の情報を取得することに焦点を当てたものが多く、ナビゲーションの要となる歩行者道路ネットワークデータの効率的な作成方法に関するものは少なかった。また、歩道の幅員情報に関しても目視による方法では正確に判別しにくいという問題があった。

そこで本論文では、幅員情報を備えた歩行者道路ネットワークデータを効率的に自動作成する手法を提案する。歩行者道路とは歩道および横断歩道のことである。提案手法は大まかに3つの工程からなり、歩道に関しては、専門家が高精度航空写真を正確にトレースする等して作成した電子地図データに一般的に歩道の領域も描かれているため、この地図から歩道中心線と幅員を求める。横断歩道に関しては、一般的に地図制作会社は地図に描いておらず情報も

持っていないため、航空写真からそれらの位置と向きを検出する。最後に、抽出した歩道と横断歩道の情報を適切に相互接続して歩行者道路ネットワークデータを構築する。提案手法を実装し、JR 山手線内ほぼ全域をカバーする約90平方kmの幅員情報付き歩行者道路ネットワークを作成し(図1)、評価した結果も報告する。本研究の貢献は、従来研究で焦点が当てられることが少なかったが、障がい者等のナビゲーションで要となる歩行者道路ネットワークデータを効率的に作成する方法を提案し、これまでにない規模のデータを実際に作成して評価したこと、また、本手法を用いて作成した道路ネットワークデータには、幅員情報に限らず様々なバリア情報を属性情報として埋め込むことができ、従来からよく収集されてきた路面の凹凸や傾斜等のバリア情報をバリアフリー経路探索に活用できる基盤を整えたことである。

2. 関連研究

障がい者等をナビゲーションするのに有用な歩行空間ネットワークデータを整備するためには、歩道および横断歩道のネットワーク、歩道の幅員や縦断勾配、段差の大きさ、階段・点字ブロック・手すり・エレベータ等の有無等を収集し、これらの接続関係を整理する必要がある。国土交通省によりガイドラインが作成されている[1]。これまで歩行空間に関するバリア情報を効率的に収集するために様々な手法が研究されてきた。

最も一般的な方法は、人が目視によりバリア情報を収集するというものである。効率化のために複数のボランティアが効率的に情報共有し、正しい情報を反映できる仕組み[2],[3],[4]や、人が現地に出向く労力をなくすためにリモートセンシング画像で判断する仕組み等が考えられている[5],[6],[7]。目視によりバリア情報を検出する手法の長

所は、機械的で自動化された方法では検出困難な多様な種類のバリア情報の収集に対応しやすいことである。一方、短所としては、歩道の幅員や縦断勾配等の定量化に必要なデータの収集が通常困難なことである。加えて、データ収集者としてボランティアを想定する場合には、参加へのインセンティブも考慮する必要がある。

一方、特定のバリア情報にフォーカスし、センサ等で自動収集する方法も提案されている。文献 [8], [9] は、車両移動時のセンサデータから車道上のくぼみを検出する試みである。車椅子を使って、歩道上の凹凸等を評価する試みも数多く行われている [10], [11], [12], [13], [14], [15]。さらに、車椅子だけでなく歩行時のセンサデータから歩道上の傾斜や階段の検出を試みた研究もある [16], [17], [18]。歩行者のデータからでも検出できれば、より多くのボランティアがバリア検出に参加できる可能性が高まる。しかし、センサでバリアを自動検出する手法は、現状、路面上の段差や傾斜の検出に限られている。

上記はいずれも、歩行空間ネットワークデータを構成するリンク情報に付与すべき属性情報の収集・整備に関するものであるが、ナビゲーションの要となる歩行者道路ネットワークデータを効率的に作成する方法の提案は少ない。現在のところ、地図制作会社でも大規模な歩行者道路ネットワークデータは整備できていない。車道ネットワークに関しては、通常、現地測量の結果や正斜投影により歪みが補正されたオルソ航空写真のトレース等から地図を作成するとともに、道路ネットワークデータも作成する方法がとられる。最近では、高解像度航空写真から自動作成する方法 [19] や車の移動軌跡データから自動作成する方法 [20], [21]、あるいはその両方を併用する方法 [22] が試みられている。移動軌跡データから自動作成する方法は情報の更新頻度を向上させられる可能性が期待できるものの、対象となるすべての道路に関する複数回の移動軌跡データを人々のプライバシー等も考慮してどのようにして集めるかが問題となる。

3. 提案手法

3.1 全体構成

提案手法の特徴は、従来から安定して提供・更新され続けており、すでに全国を網羅している既存の地理データのみを活用し、これらに対して多様な処理を積み重ねることで、新たな地理データの取得コストをいっさいかけずに実用的な効率と精度でデータ生成できるようにしていることである。具体的には、既存の航空写真と電子地図データのみを入力データとする。電子地図データは専門家が高精細航空写真をトレースする等して作成されており、図 2 に示すとおり、一般的には歩道の領域が緯度・経度座標で表されるポリゴンデータとして含まれている。一方、横断歩道は、航空写真には写っているものの地図に描かれていることは少ない。



図 2 歩道領域を含む地図 (左) と航空写真 (右)
 Fig. 2 Map containing sidewalk areas (left) and aerial photograph (right).

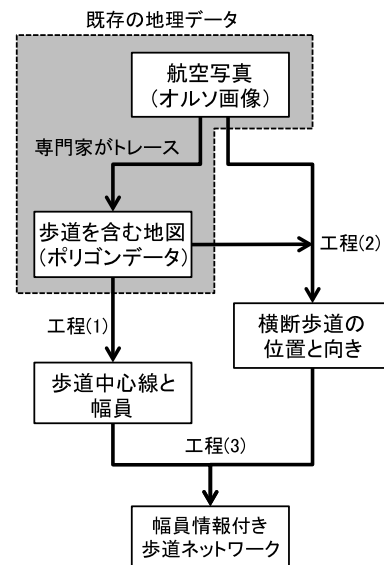


図 3 処理全体の流れと出力データ
 Fig. 3 Overall workflow and output data.

図 3 に提案手法の処理全体の流れと出力データを示す。提案手法は大まかに 3 つの工程から構成される。1 つ目の工程では、地図に含まれている歩道ポリゴンデータから歩道中心線と幅員を抽出する。本研究で抽出したい歩道中心線は人が進行する方向を表すものである。2 つ目の工程では、航空写真から横断歩道の位置と向きを検出する (ただし地図データに含まれている車道のポリゴンも処理領域のマスクに使用)。航空写真は正斜投影により高層建築物で路面が隠れていることが少ないオルソ画像が望ましい。最後の工程では、1 つ目および 2 つ目の工程で抽出した歩道中心線と横断歩道の情報を適切に接続し、歩行者道路ネットワークデータを構築する。

3.2 工程 (1) : 歩道中心線と幅員の抽出

歩道ポリゴンデータから歩道中心線を抽出する工程で

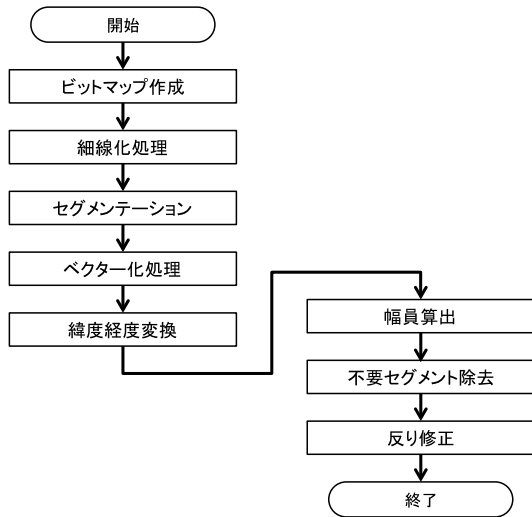


図 4 歩道中心線と幅員の抽出フロー

Fig. 4 Sidewalk centerline and width extraction workflow.

ベースとなる手法は、指紋画像認識の前処理等で使われる細線化アルゴリズムである。ただし、通常の細線化処理を適用するだけでは、幅員が広めの歩道において、求めた歩道中心線に余計な分岐が発生してしまう。また、図 2 にあるように、歩道の端は、アール（カーブ半径）を持たせてある、いわゆる巻き込み部分となっているため、歩道の端に近づくとつれ、歩道中心線が反った形状になってしまい、ナビゲーションで進行方向を示すには好ましくなく、幅員の抽出にも悪影響を与える。当然、人手で作成した歩道中心線には余計な反りはない。そこで提案手法では、細線化によって得られる歩道中心線に対して余計な分岐を除去し、反りを修正する処理を導入する。図 4 は、歩道中心線と幅員の抽出フローである。以下で処理の詳細を述べる。

ビットマップ作成

電子地図に含まれている歩道ポリゴンデータは、緯度および経度のリストとなっている。それらを順に接続して形成されるポリゴンで歩道の領域が表される。最初に、このデータを 2 値のビットマップ画像に変換する。緯度・経度は小数点で表されるため、歩道ポリゴンデータをビットマップに変換する前に、所望の精度に応じて、歩道ポリゴンデータの緯度・経度に十分大きな値をかけて整数化する。そして、緯度の最大値と最小値との差分を縦幅とし、経度の最大値と最小値との差分を横幅とするビットマップを作成し、このビットマップの全体に歩道ポリゴンデータをモノクロで描画する。本論文の評価では、緯度「35.692694242」、経度「139.702499333」といった値に 10000000 をかけて、「356926942」「1397024993」という値に整数化し、1 画素が緯度方向の 10 cm に相当するよう変換した。

細線化処理

次に、歩道ビットマップを細線化する処理を行い、歩道の領域から、当該領域の幅方向の中心線を抽出する。細線化については様々なアルゴリズムが提案されているが、こ

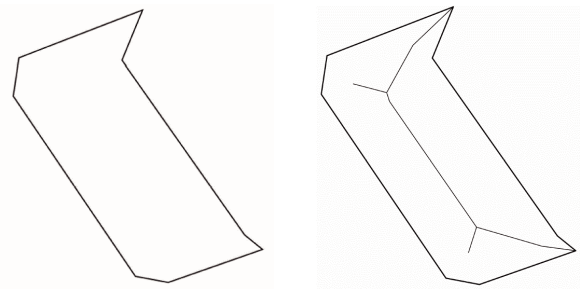


図 5 歩道ポリゴンデータ (左) と細線化した結果 (右)

Fig. 5 A sidewalk polygon (left) and the result of thinning the shape (right).

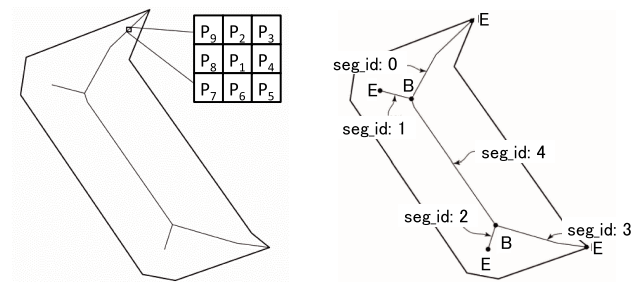


図 6 画素へのナンバリング (左) とセグメンテーションの例 (右)

Fig. 6 Numbering to pixels (left) and segmentation of the extracted centerline (right).

こでは Zhang と Suen による手法 [23] を適用する。この手法の概要は、生成された歩道ビットマップをラスタスキャンし、注目画素を中心に 3×3 の画像値のパターンを観測し、注目画素が以下の 3 条件を満たすとき、この注目画素の値を白画素の値にするというものである [24]。

- (1) 境界上にある黒画素であること。
- (2) 白画素に変更しても連結性が保存されること。
- (3) 端点でないこと。

図 5 の左は歩道ポリゴンデータの境界の 1 例を示し、右はそれに対して細線化処理の結果を重ねた例である。

セグメンテーション

次に、後のベクター化処理のために、細線化画像のセグメンテーションを行う。セグメンテーションとは、細線化画像における中心線を、当該中心線の端点、分岐点をもとに分割して複数の線を抽出する処理である。細線化画像をラスタスキャンし、注目する画素を P_1 とし、この P_1 の値が黒であるとき、 P_1 に隣接する 8 つの画素を図 6 の左のように P_2 から P_9 にナンバリングする。そして、これら P_2 から P_9 の画素における値が黒である画素の数 C_1 をカウントする。この結果が $C_1 = 1$ であれば、画素 P_1 を端点に設定する。また、 $C_1 = 1$ 以外であれば、ナンバリングした画素の値を $P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$ の順に並べて計 9 つの値からなる配列を作成し、この配列における、白の次に黒になる箇所の数 C_2 をカウントし、 $C_2 \geq 3$ であれば、 P_1 を分岐点とする。すべての端点

および分岐点を見つけた後は、これらの点間で線が存在する組を輪郭追跡により探し、2点の組のリストを作成する。以後、これら2点の組でつながれた線をセグメントと呼ぶこととする。図6の右は細線化したビットマップにセグメンテーションを施した結果である。Eが端点、Bが分岐点であり、seg_idがセグメントのIDである。

ベクター化、緯度・経度への変換

次に、ビットマップデータである個々のセグメントに対してベクター化処理を実施する。ここでは文献[24]の194ページに記載されている2分割法を用いる。これにより、ビットマップとして表されていた歩道中心線はいくつかの代表点を直線で結んだベクターデータとして表される。最後に、歩道ポリゴンをビットマップに変換したときの情報を用いて、各々の点の座標を緯度・経度に戻す。

幅員算出

歩道の幅員を算出するために、ベクターデータとして表される歩道中心線の各ポイントにおける、歩道ポリゴンの境界までの最小幅を求める。この方法としては、各ポイントにおける中心線の傾きを算出し、それに対する垂線を歩道ポリゴンの境界まで伸ばして幅員を求める方法が第一に考えられる。しかし、ベクター化処理をどの程度の細かさで行うか等の影響によって垂線の方が必ずしも最小距離とはならないケースが発生するため、ここでは歩道中心線の各ポイントから最小距離となる方向を総あたりで確実に求める方法をとる。

図7は、この方法を図式化したものである。具体的には、歩道中心線を形成する各々のポイント P_s に対し、当該ポイントの横方向に、歩道が描かれている画素に対応する配列を作成する。次に、この配列の各要素に対して、上下2方向を走査し、歩道の境界までの距離がより短くなるほ

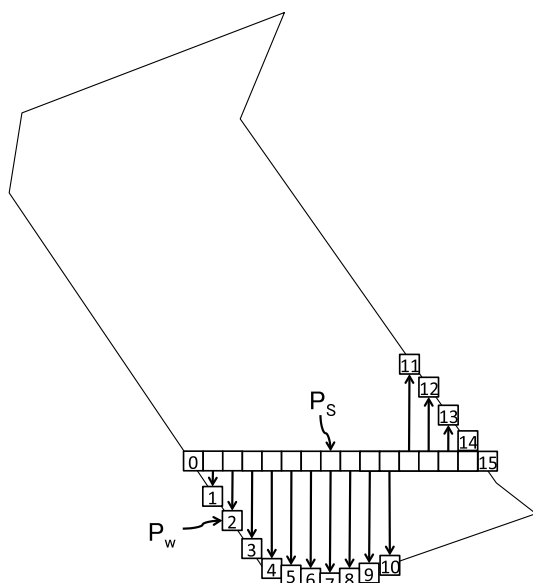


図7 幅員算出過程

Fig. 7 Extraction process of sidewalk width.

うの画素を検出する。図7ではこのようにして0から15までの画素が検出されている。次に、これらの画素をすべて緯度・経度座標に戻した後、 P_s までの距離を求め(本評価で緯度・経度から距離を求める際には、すべてヒュベニの公式を用いた)、最小距離となる画素 P_w を見つける。最後に、この距離を2倍したものを P_s における幅員とする。不要セグメントの削除

全長に対して幅員が広めの歩道では、求めた歩道中心線の端に余計な分岐が生じ、不要なセグメントが作成されてしまう。そこで、これらを除去する処理を行う。ここでは、緯度・経度の座標列として示される線のうち、同一座標の分岐点から延びる複数の線であって、分岐点を一端として他方が端点である線を抽出し、それらの中に所定の長さ(本評価では3m)未満の線があるときのみ、それらの線を分岐がなくなるまで削除する。図8は不要セグメント削除処理の結果の例であり、図6の右からseg_id:1およびseg_id:2が削除されている。さらに不要セグメントを削除した結果、残ったセグメントが1つのパスとして表せる場合には、データ形式を1セグメントに変換する。具体的には、各セグメントの端点のなかに他のセグメントの端点と重複するものがあり、その重複点が1つだけであるかを判定する。図8で、seg_id:0はa, b, cの3点, seg_id:4はc, d, eの3点, seg_id:3はe, f, gの3点からなるとすると、cとeがそれに該当するのでこの部分でセグメントどうしがつながられ、a, b, c, d, e, f, gという7つの連続する点からなる1セグメントのデータに変換される。

反り修正

一般的に歩道の端は、アールを持たせてある巻き込み部分となっているため、歩道中心線のベクターデータは、歩道の端に近づくにつれ、図8のように曲がった形状になる。そこで、この曲がった形状を整形する反り修正処理を行う。図9の左は、反りの判定と修正方法を図式化したものである。具体的には、最初に歩道中心線の端点を見つけ、この端点を図9の左に示すAとし、これが歩道中心線の先頭のポイントだとすると、ポイントを末尾方向にたどっていき、隣接するポイント間の長さが反り修正の基準となる

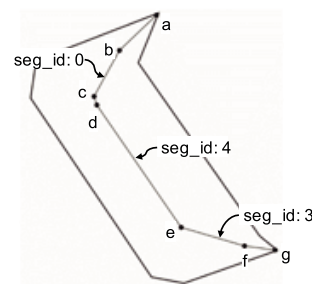


図8 不要セグメントを削除した例

Fig. 8 Deletion of unnecessary segments from the extracted centerline.

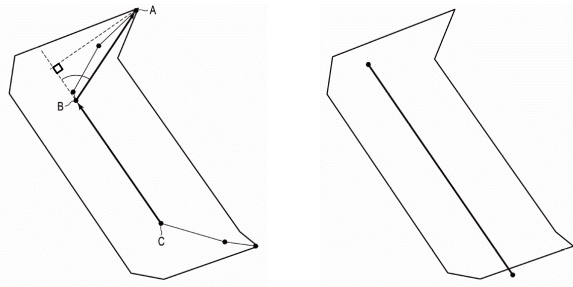


図 9 反りの判定・修正 (左) と反り修正結果 (右)

Fig. 9 Detection and modification of curves in the centerline (left) and the straightened centerline (right).



図 10 反り修正処理で除外される例. 端線の長さが L_{first} 以上のケース (上). 端線の両端の幅員がほぼ等しいケース (下)

Fig. 10 Two cases where an end line segment of the sidewalk centerline is not straightened: The length of the end line segment is L_{first} or more (top), and both widths of the end line segment are almost the same (bottom).

長さ L_{base} (本評価では 4.88m) 以上となる線分を探し出す. この L_{base} 以上となる線分を図 9 の左に示す BC とすると, ベクトル \mathbf{CB} とベクトル \mathbf{BA} の $\cos\theta$ を求め, \mathbf{CB} の B 側を $|\mathbf{BA}| \cos\theta$ の長さ分だけ延長し, 点 B の直前までのポイントをすべて削除する. また, 端点 A が, 歩道中心線の末尾のポイントである場合は, ポイントを逆方向にたどり, 同様の処理を実施する. 以上の処理の結果, 反りが修正された歩道ベクターデータが生成される. 図 9 の右は, 反り修正結果の 1 例である.

ただし, 以下に該当する場合は反り修正処理を実施しない.

- (1) 端点 A から隣接するポイントまで (端線) の長さが反り修正除外基準となる長さ L_{first} (本評価では 9.7m) 以上である場合
- (2) 端線両端の幅員 W_1 と W_2 の比 W_1/W_2 が 1 に近い場合

図 10 は, これらの具体例を示したものである.

以上の処理をすべての歩道ポリゴンデータに対して行う. 図 11 では, 不要セグメント除去および反り修正処理の効



図 11 不要セグメント除去および反り修正していない歩道中心線 (上). 不要セグメント除去および反り修正した歩道中心線 (下). 青は幅員 1m 未満, ピンクは 1m 以上で濃いほど広い

Fig. 11 Unstraightened sidewalk centerlines (top), where unnecessary segments are left, and straightened sidewalk centerlines (bottom), where unnecessary segments were removed. Blue lines are less than 1 m wide, pink lines are 1 m wide or more, and the deeper the pink, the wider the width.

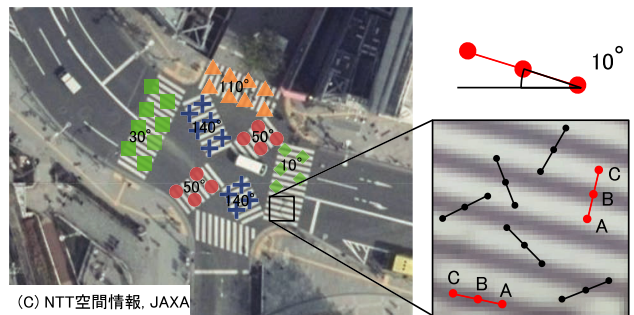


図 12 横断歩道の検出過程. 同一角度の模様を検出された箇所には同一記号を描画

Fig. 12 Crosswalk detection process. The same symbols were assigned to the same angle patterns.

果を複数の歩道で比較している. 上図はこれらの処理を実施しなかった場合であり, 下図は実施した場合である.

3.3 工程 (2) : 横断歩道の位置と方向の検出

電子地図には一般的に描かれていない横断歩道の情報を抽出するためにここでは航空写真を用いるが, そのためには横断歩道の縞模様がおおむね 45 cm 間隔であることを利用する. 具体的には, 図 12 に示すように, 一定サイズのウィンドウ (本評価では 4 m 四方) を使って航空写真を走査する. このウィンドウ内で, 直線上 45 cm 間隔の点 A, B, C をランダムに様々な位置と角度でつくり, 以下の 1

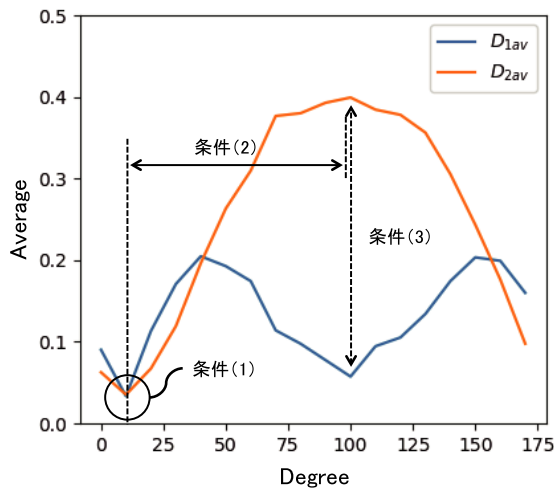


図 13 横断歩道上における評価値

Fig. 13 Evaluation values on a crosswalk.

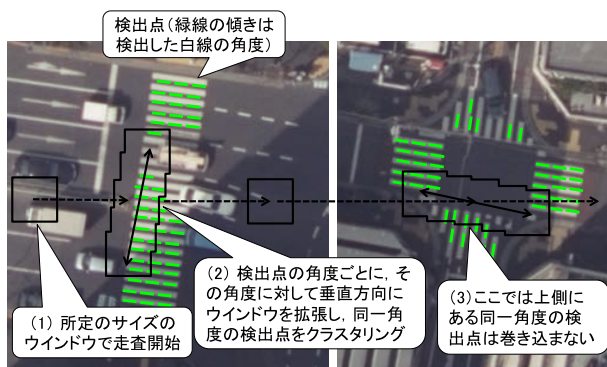


図 14 検出点のクラスタリング

Fig. 14 Clustering of detected points.

階差分絶対値 D_1 と 2 階差分絶対値 D_2 を求める.

$$D_1 = |V_C - V_A|$$

$$D_2 = |(V_C - V_B) - (V_B - V_A)|$$

ここで、 V_A , V_B , V_C は A, B, C それぞれの位置に対応する画像上の明度である. 次に図 13 に示すように、これらの 2 つの値のそれぞれの平均 D_{1av} と D_{2av} を直線の角度別 (本評価では 10 度間隔 18 段階) に求めると、横断歩道上では次の条件を満たすことになる.

条件 (1) D_{1av} の極小値の角度の 1 つと D_{2av} の極小値の角度が一致

条件 (2) D_{2av} の極小値と極大値の角度の差分が 90 度

条件 (3) D_{1av} の極小値の角度の 1 つと D_{2av} の極大値の角度が一致

ウィンドウを走査し、上記 3 条件に該当する場合、その緯度・経度座標を D_{2av} の極小値の角度 (横断歩道の白線の角度に相当) とともに出力する.

次に、これらの出力結果から位置と角度が近いものをクラスタリングする. 図 14 は横断歩道の検出座標をクラスタリングする方法を示したものである. 最初は所定

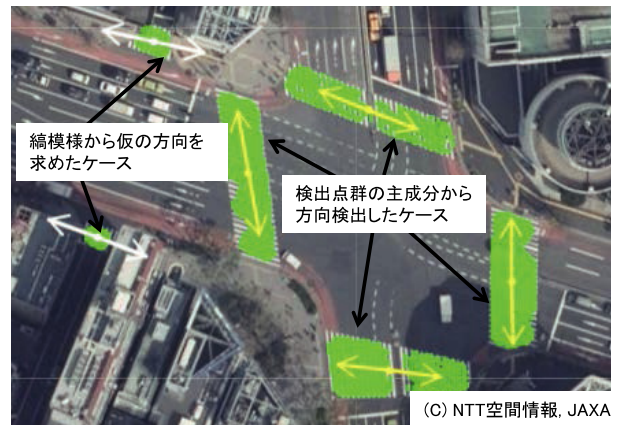


図 15 推定された横断歩道の方向

Fig. 15 Estimated directions of crosswalks.

のサイズ (本評価では 3.5m 四方) のウィンドウで走査を開始する. ウィンドウ内に検出点が入った場合には、当該検出点を持つ角度の種類をすべて列挙する. そして、角度の種類ごとに、当該角度の垂直方向にウィンドウを拡張し、同一角度の検出点がウィンドウ内に入れば、それらを 1 つにクラスタリングする. たとえば、初期サイズのウィンドウ内に 30 度と 50 度の 2 種類の検出点が入ったならば、まず 30 度の検出点に対して垂直な 120 度と -60 度方向に初期サイズのウィンドウを拡張し、30 度の検出点のみをクラスタリングする. 次に同じ走査位置で 50 度の検出点に対して垂直な 140 度と -40 度方向にウィンドウを拡張し、50 度の検出点のみをクラスタリングする. このようにして、図 14 の左の写真にあるように、全長が長い横断歩道において検出点が途中で離れていても 1 つにクラスタリングすることを可能にし、同図の右の写真にあるような狭い交差点においては同一角度の白線を持つ 2 つの横断歩道を別々にクラスタリングすることを多くの場合可能にする. クラスタリングしたあとは、個々のクラスタの中心座標も求めておく.

最後に横断歩道の方向 (人が進行する方向) を求める. そのためには、同一クラスタに分類された横断歩道検出点の 2 次元座標に対して主成分分析を行い、第 1 主成分の分散 E_1 と第 2 主成分の分散 E_2 の比 E_1/E_2 が一定以上であるならば、第 1 主成分の方向を横断歩道の方向とする. しかし、一定未満すなわち横断歩道の長さが短い場合には、この方法で方向を検出することはできない. その場合には、横断歩道の白線の角度 D_{2av} に対する垂直方向を仮の方向とする. ただし、横断歩道の白線の方向は横断する車道方向であるため、それに対する垂線の角度は必ずしも横断歩道を人が進行する方向を表すわけではない. しかし、後述する歩道と横断歩道の接続処理では、長さが短い横断歩道は、歩道中心線の情報を起点として横断歩道の向きの情報に頼らずに歩道と接続できることが多いため、問題が発生することは少ない. 図 15 は 2 種類の方法で求めた横断歩道の方向の例である.

なお、航空写真を走査する際には、電子地図データに含まれている車道ポリゴンデータを使い、航空写真の車道の領域のみを走査対象とする。これにより処理を高速化するとともに、車道ではない領域における横断歩道の誤検出を防ぐことができる。

3.4 工程 (3)：歩道中心線と横断歩道の接続

ここまでの工程で、歩道中心線と横断歩道のデータが準備できる。最後の工程では、複数の歩道中心線と横断歩道の情報を適切に相互接続し、歩行者道路ネットワークデータを構築する。ここでは接続の条件を以下とする。

- 実際に横断歩道によって連結されている歩道どうしはデータ上でも実際の連結位置に近い形で無条件に接続する。
- 横断歩道によって連結されていない場合でも、歩道中心線の端点およびコーナから一定の短い距離以内に別の歩道があれば（障がい者が渡っても危険が少ない場所と距離）接続する。

すなわち、横断歩道のデータを起点として接続箇所を探索するだけでなく、歩道を起点としても接続箇所を探索する必要がある。なお、横断歩道がない箇所での接続条件を端点もしくはコーナに限定している理由は、これらの場所が車道の分岐点に相当し、車の運転手が注意して走行する箇所と考えられるからである。図 16 は本工程のフローである。以下で処理の詳細を述べる。

端点・コーナ検出

最初に、歩道中心線の端点およびコーナを検出する。端点は、個々の歩道中心線を構成するセグメントの先頭と末尾のうち、他のセグメントと重複していない点である。一方、コーナに関しては、まず歩道中心線を形成する個々の線分の長さを求める。次に、この求めた長さが一定長（本評価では 5.5m）以上となる 2 つの線分からみて歩道中心線上においてなるべく中間の距離にある点をコーナとして検出する。具体的には、上記 2 つの線分の間にある歩道中心線上の各点から両線分までの中心線上の距離を計算し、

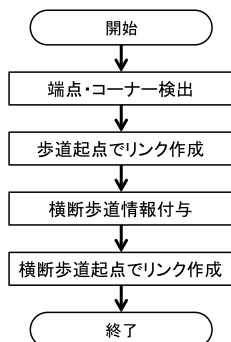


図 16 歩道中心線と横断歩道の接続フロー

Fig. 16 Workflow for interconnecting sidewalk centerlines and crosswalks.

それらの差が最も小さい点をコーナとする。

ポイント補間

次に、歩道中心線上の各点の間に、細かい間隔で新たな点の座標を挿入（補間）するポイント補間処理を行う。具体的には図 17 に示すように、歩道中心線ベクターデータにおける隣接する 2 点からベクトルを生成し、各ベクトルの距離を求めて単位ベクトルを生成し、この単位ベクトルを定数倍することで挿入点の座標を取得する（本評価では 1m ピッチとした）。

歩道起点でリンク生成

次に、端点およびコーナを起点として、リンクを生成する。個々の端点およびコーナに対して、その他の点の中から、歩道中心線ごとに所定の距離以内（本評価では 6m）にある 1 点を抽出し、それらの座標を結んだ線分をリンク情報として保存する。ここでの探索対象は、補間したポイントも含む。

横断歩道情報付与

次に、上記リンク情報に付与すべき横断歩道情報が存在するかを判別し、存在するならば付与する処理を行う。そのためには上記リンク情報を形成する 2 点間の中心座標を求め、この中心座標から一定距離（本評価では 4m）未満にある横断歩道の中心座標を探索し、このような中心座標が存在すれば、リンク情報に「横断歩道あり」という属性情報を付与する。なお、ここで一定距離以内にあるとされた横断歩道の情報に対しては、後述する横断歩道起点でのリンク作成に重複して利用されないよう使用済みとするフラグをつけておく。図 18 は端点・コーナを検出した直後の 1 例で、横断歩道の検出座標も緑の点群で示している。図 19 は歩道の情報を起点としてリンクを作成し、作成したリンクに横断歩道有無の情報を付与した例であり、この段階ではまだ利用されていない横断歩道の検出座標は残されている。

横断歩道起点でリンク作成

最後に、残された横断歩道の情報を起点としてリンク情報を生成する。この処理のために考えられる通常の方法は、横断歩道の方向を表す線分の延長線上で歩道中心線と

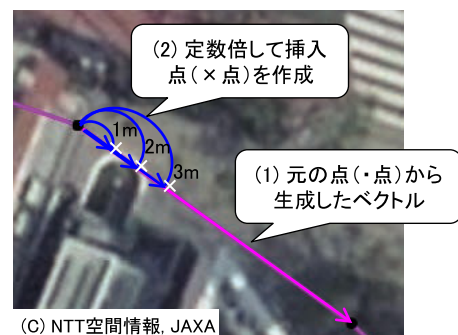


図 17 歩道中心線に対するポイント補間過程

Fig. 17 Point interpolation process in the sidewalk centerline.

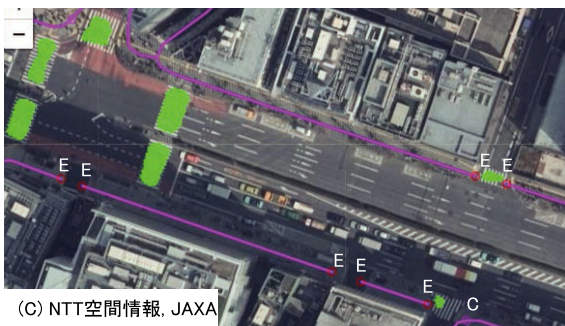


図 18 端点 (E) とコーナ (C) の例. 緑の点群は横断歩道検出点群
Fig. 18 Examples of edges (E) and corners (C). Green points show the detected points on crosswalks.

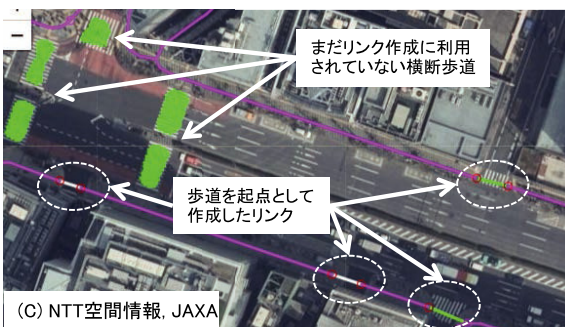


図 19 歩道を起点としてリンク作成後、横断歩道属性を付与した例
 (白線は横断歩道なし、緑線はあり)
Fig. 19 Examples of links emerged from sidewalk data; green/white lines show links with/without crosswalks.

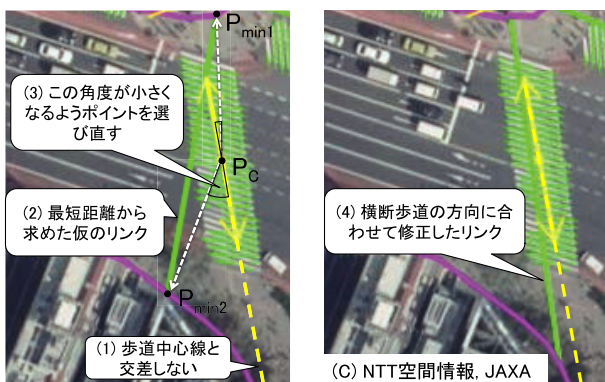


図 20 横断歩道を起点としたリンク作成の例
Fig. 20 An example of link emerged from crosswalk data.

の交点を求めることであろう。しかし、図 20(1)の黄色の点線で示すように、歩道の領域に対して細線化処理を施した後では横断歩道の方向の延長線が歩道中心線と交差しないことがある。そこで、実際の横断歩道の方向になるべく近づける形で横断歩道を起点とするリンク情報を作成する。具体的には以下のように処理する。

個々の横断歩道から最短距離にある歩道中心線のポイントを探し、 P_{min1} とする。ここでの探索対象は、補間したポイントも含む。さらに、上記の歩道中心線を除外したうえで再度同じ処理を行い、検出したポイントを P_{min2} と

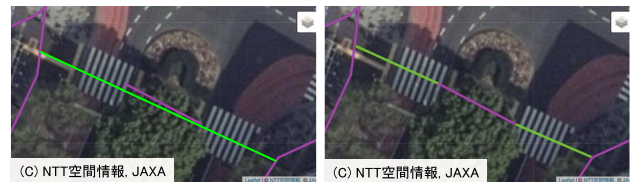


図 21 安全島を経由せずに形成されたリンク (左) とその修正 (右)
Fig. 21 A link passing over the safety island (left) and the modified links (right).

する。図 20(2)の緑の実線で示すように、これら P_{min1} と P_{min2} を結んだ線分が仮のリンクとなるが、このままでは、実際の横断歩道と歩道の連結位置から大きく離れるため、接続位置を修正する処理を行う。

そのためには、先に求めた横断歩道の方向を示すベクトル (図 20 の黄色の実線矢印) と横断歩道の中心 P_c から P_{min1} および P_{min2} それぞれへのベクトルとの角度 (図 20(3)) を算出し、これらの角度がなるべく小さくなるよう歩道中心線上のポイントを選び直す。具体的には、上記の角度が最小となるポイントをまず取得し、その角度との差が一定以下で、かつ P_c との距離が最も短くなるポイントを選び直す。図 20(4)の緑の実線はこのようにして修正した後のリンクである。

なお、先の横断歩道検出点のクラスタリング処理で1つのクラスタとされた全長が長い横断歩道の中間に安全島としての歩道が存在する場合がある。このような場合、そのままでは図 21 の左のように当該横断歩道の両端に存在する歩道どうしのみを接続するリンクが作成され、中間の歩道は無視されてしまう。そこで、横断歩道を起点としてリンクを作成する場合には、リンクの中心点から所定の範囲内 (本評価では 3m) に歩道中心線が存在するかを判定し、存在するならば、当該リンクをその中心点で分割し、分割した点を歩道と接続している点側に一定距離戻した後、そこから最小距離にある歩道中心線上のポイント (安全島を想定) を探索し、接続し直すことで、図 21 の右のように安全島としての歩道を経由したネットワークを構築する。最後に、ポイント補間処理で挿入したポイントのうち、リンクの作成に使われなかったポイントをすべて削除する。

4. 評価

提案手法のうち、歩道情報の抽出に関する不要セグメント削除と反り修正以外の処理と横断歩道検出の処理は Python3 系で実装した。歩道の不要セグメント削除および反り修正処理と、歩道中心線と横断歩道情報の接続処理は Javascript で実装し、ウェブブラウザ上で地図に重畳表示しながらのパラメータ調整と Node.js 上での一括処理のどちらからでも利用できるようにした。また、複数の歩道ポリゴンデータや分割された航空写真を設定値に基づいて並列処理できるようにした。ただし、横断歩道検出点のクラ

表 1 評価結果一覧

Table 1 Evaluation results.

評価項目	結果	備考
歩道情報抽出の処理時間	6.5 時間	対象とした領域には約 1 万の歩道ポリゴンデータが存在
横断歩道検出の処理時間	7 時間	約 90 平方 km の 10 cm 解像度航空写真を走査
歩道中心線と横断歩道の接続処理時間	10 分	
歩道中心線・幅員抽出の精度	99%	無作為に 200 カ所の歩道を抽出し、中心線がとれているかは目視で確認、幅員は航空写真上の 2 点をクリックすると、その間の幅が計測できるツールを作成して誤差 5%未満を正解とした
横断歩道検出の再現率	93.7%	無作為に選んだ領域に存在した 239 カ所の横断歩道で、クラスタリング、方向検出まで正しくできたかを確認
横断歩道検出の適合率	100%	
歩道中心線と横断歩道データの接続精度	95.5%	リンクを作成すべき箇所を無作為に 200 カ所選び確認

スタリング処理については、複数の航空写真にまたがる横断歩道検出点を 1 つにクラスタリングすべき可能性があるため、1 プロセスで処理する実装とした。歩道中心線と横断歩道情報の接続処理も 1 プロセスで処理する実装とした。

これらの実装を用いて、JR 山手線内のほぼ全域を含む緯度 35.6197~35.738062、経度 139.701334~139.778837 の矩形範囲約 90 平方 km における幅員情報付き歩行者道路ネットワークデータを作成し、評価した。入力データとしては NTT 空間情報株式会社から販売されている電子地図（高架下に隠れている箇所以外の歩道ポリゴンデータが含まれている）と 10 cm 解像度のオルソ航空写真を利用した。歩道中心線と幅員の抽出には、歩道ポリゴンデータをどの程度の大きさのビットマップに変換するかで処理時間が大幅に変わるが、ここでは 10 cm 解像度相当（1 画素が緯度 10 cm に相当）で処理した。評価に使用した計算資源は CPU：Intel Core i7-7700HQ 2.8 GHz、メモリ：16 GByte、OS：Windows10 の PC 1 台のみである。並列処理は 8 並列に設定した。評価項目とそれらの結果を表 1 に示す。

5. 考察

処理時間に関しては、すべての工程を足し合わせても、PC 1 台で半日程度あれば東京都の中心のように道路が密集している地域でも約 90 平方 km 分の処理が完了する。幸



図 22 歩道中心線（人が進行する方向）と幅員が正しく抽出できないケース

Fig. 22 A case where the centerline and width of a sidewalk cannot be extracted correctly.

いにも歩道の形状を表すデータは複数のポリゴンに分かれているため、計算資源が準備できるならば、さらに処理を並列化し、短時間で処理することも可能である。分割航空写真を用いた横断歩道の検出処理についても同様である。

歩道中心線の抽出精度は 99%となったが、どのような場合に処理が失敗しているかを確認したところ、歩道の長さより幅員のほうが大きい歩道において、反り修正等での失敗ではなく、細線化処理の段階で失敗していることが判明した。このような形状の歩道は、図 22 に示すように、長い横断歩道の中にしばしば存在する安全島であった（すべての安全島において幅員のほうが大きいわけではない）。このような歩道は、ゆっくりにしかな移動できない者が安全に通過できるかを判定するためには、正確に情報抽出されるべきであろう。しかし、この形状から中心線（人が進行する方向）を正しく抽出するには提案手法のように歩道ポリゴンを個別に処理するだけでは困難で、各ポリゴンの周囲の状況も考慮したうえで中心線を求める必要があり、今後の課題である。現状での対処方法として、このような歩道は中心線抽出の成否にかかわらず全長が著しく短くなるという特徴を利用して、手作業で修正するために存在する箇所を自動で割り出すといったことは可能である。横断歩道検出の再現率は 93.7%であったが、その詳細は、幅が狭く長さも短い横断歩道に対して、検出段階で失敗しており、検出点のクラスタリングや方向検出によるものではなかった。また、歩行者道路のネットワーク構築で失敗している箇所も、横断歩道検出の失敗に起因していた。とはいえ、各種精度はいずれも 95%前後を達成しており、最終的に専門家による確認作業を実施するとしても、その作業は軽微なものになると思われる。

抽出した幅員の評価については、提案手法で抽出した幅員と航空写真から手作業で計測した幅員を比較し、差が 5%未満となる場合を正解としたが、歩道の中心線がとれていると判断できた場合には、すべて正解となった。提案手

法では、歩道ポリゴン上の幅員、すなわち専門家がトレースした形状における幅員は、総あたりで探索しているため確実に取得できていると考えられる。歩道ポリゴン上の幅員と航空写真上の幅員との差に関しては、トレースの専門家ではない著者が計測した航空写真上の幅員のほうが不正確で、両者の本当の差は評価基準とした5%未満よりさらに小さい可能性も考えられる。

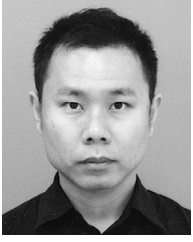
一方、幅員情報の活用という観点からは、本手法で抽出した幅員は実際に通過できる幅員（有効幅員）ではないことに注意する必要がある。実際の歩道上には電柱や植木等が存在している可能性があり、有効幅員は抽出した幅員以下である。そのため、本手法で抽出した幅員をそのまま利用するのであれば、たとえば、電動車椅子の幅がJIS規格で700mm以下であるのに対し、余裕をみて2m以上の幅員を持つ歩道のみを通過できる経路として提示する等すべきである。

6. 結論

本論文では、障がい者等を電子的にナビゲートするために不可欠な幅員情報付き歩行者道路ネットワークデータを自動作成する方法を提案し、評価した。今後は提案手法で作成した歩行者道路ネットワークに、関連研究として述べたような目視で集められた点字ブロック等の情報やスマートフォンのセンサ等で収集された縦断勾配等の情報も組み込み、より有用な歩行空間ネットワークデータとして整備していくとともに、多様な歩行者に柔軟に対応できるインタラクティブな経路案内サービス等へのデータ提供を進めていく予定である。

参考文献

- [1] 国土交通省：歩行空間ネットワークデータ等整備仕様 (2018), 入手先 (<http://www.mlit.go.jp/common/001244374.pdf>) (参照 2019-04-05).
- [2] Sozialhelden, e.V.: Wheelmap, available from (<http://wheelmap.org>) (accessed 2019-04-05).
- [3] Miura, T., Yabu, K., Ikematsu, S., Kano, A., Ueda, M., Suzuki, J., Sakajiri, M. and Ifukube, T.: Barrier-free Walk: A Social Sharing Platform of Barrier-free Information for Sensory/Physically-impaired and Aged People, *Proc. SMC '12*, pp.2927–2932 (2012).
- [4] 山本千尋, 船越 要, 小長井俊介, 小西宏志, 川野辺彰久: 歩行者移動支援のためのバリアフリー関連情報収集手法の提案, 信会技報, Vol.116, No.23, LOIS2016-8, pp.39–44 (2016).
- [5] Hara, K., Le, V. and Froehlich, J.E.: Combining Crowdsourcing and Google Street View to Identify Street Level Accessibility Problems, *Proc. CHI '13*, pp.631–640 (2013).
- [6] Rundle, A.G., Bader, M.D.M., Richards, C.A., Neckerman, K. and Teitler, J.: Using Google Street View to Audit Neighborhood Environments, *American Journal of Preventive Medicine*, Vol.40, No.1, pp.94–100 (2011).
- [7] Badland, H., Opit, S., Witten, K., Kearns, R. and Mavoa, S.: Can Virtual Streetscape Audits Reliably Replace Physical Streetscape Audits?, *Journal of Urban Health*, Vol.87, No.6, pp.1007–1016 (2010).
- [8] Eriksson, J., Giriod, L., Hull, B., Newton, R., Madden, S. and Balakrishnan, H.: The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring, *Proc. MobiSys '08*, pp.29–39 (2008).
- [9] Mohan, P., Padmanabhan, V.N. and Ramjee, R.: Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones, *Proc. SenSys '08*, pp.323–336 (2008).
- [10] 牧 恒雄, 竹内 康, 松田 誠: 歩道の凹凸評価方法に関する研究, 第1回舗装工学講演会論文集, pp.151–158 (1996).
- [11] 石田真二, 亀山修一, 岳本秀人, 姫野賢治, 鹿島 茂: 車椅子の走行負荷に基づいた歩道の路面凹凸評価方法, 土木学会論文集 E, Vol.62, No.2, pp.295–305 (2006).
- [12] 岡田美好: 車いすの乗り心地に着目した歩行者系舗装の性能指標に関する一考察, 土木学会舗装工学論文集, Vol.14, pp.189–194 (2009).
- [13] 岩澤有祐, 矢入郁子: 多次元時系列データ解析によるアクセシビリティ可視化システムの開発, *JSAI '14* (2014).
- [14] 隅田康明, 松永勝也, 合志和晃, 志堂寺和則: 車いす利用者向け経路探索のための路面の傾斜及び段差測定システムの開発, 信学技報, Vol.114, No.357, WIT2014-64, pp.63–68 (2014).
- [15] Kuwahara, N., Nishimura, M., Shiomi, Y., Morimoto, K., Iwawaki, Y. and Nishida, N.: A Study on a Ubiquitous System for Collecting Barrier-free Information of Evacuation Centers for Wheelchair Users, *Proc. CASEMANS '10*, pp.36–39 (2010).
- [16] 佐藤 匠, 廣森聡仁, 山口弘純, 東野輝夫: スマートフォンと靴センサを活用した災害時通行路の状況推定, *DICOMO '14*, pp.258–265 (2014).
- [17] 藤井海斗, 羽田野真由美, 西田京介, 戸田浩之, 澤田 弘, 鹿島久嗣: 歩行者クラウドセンシングによる路面状態の推定, *DEIM '16* (2016).
- [18] 宮田章祐, 荒木伊織, 王 治順, 鈴木天詩: 健常歩行者センサデータを用いたバリア検出の基礎検討, 情報処理学会論文誌, Vol.59, No.1, pp.22–32 (2018).
- [19] Sghaier, M.O. and Lepage, R.: Road Extraction From Very High Resolution Remote Sensing Optical Images Based on Texture Analysis and Beamlet Transform, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol.9, No.5, pp.1946–1958 (2016).
- [20] Li, H., Kulik, L. and Ramamohanarao, K.: Automatic Generation and Validation of Road Maps from GPS Trajectory Data Sets, *Proc. CIKM '16*, pp.1523–1532 (2016).
- [21] Zhang, Y., Liu, J., Qian, X., Qiu, A. and Zhang, F.: An Automatic Road Network Construction Method Using Massive GPS Trajectory Data, *International Journal of Geo-Information*, Vol.6, No.12 (2017).
- [22] Yuan, J. and Cheriadat, A.M.: Image driven GPS trace analysis for road map inference, *Proc. SIGSPATIAL '13*, pp.480–483 (2013).
- [23] Zhang, T.Y. and Suen, C.Y.: A Fast Parallel Algorithm for Thinning Digital Patterns, *Comm. ACM*, Vol.27, No.3, pp.236–239 (1984).
- [24] 画像情報教育振興協会: デジタル画像処理 [改訂新版] (2015).



前田 篤彦 (正会員)

1998年武蔵野美術大学造形学部映像学科卒業。2003年北陸先端科学技術大学院大学知識科学研究科博士後期課程修了。同年日本電信電話株式会社入社。現在、NTT サービスエボリューション研究所で、スマートシティ、データマイニング関連の研究に従事。博士（知識科学）。2010年度情報処理学会論文賞，山下記念研究賞，インタラクシオン2009，2011 ベストペーパー賞。