

一人称ビジョンを用いた 視覚障害者道路横断支援システムの検討

川口 智士^{1,a)} 榎並 直子^{2,b)} 有木 康雄^{2,c)}

概要: 視覚障害者の単独歩行を支援するブラインドナビゲーションの実現のため、道路横断支援システムの検討と、周辺環境認知の第一歩として、歩行者の交差点上の位置と進行方向を推定する手法を提案する。歩行者の一人称画像と Google Street View 画像のマッチングにより交差点上の自己位置と進行方向推定する。検討した撮影設定により撮影した画像データセットの構築とデータセットを用いた評価により提案手法の有効性を示す。

1. はじめに

約 32 万人の視覚障害者の多くは単独歩行のために、日常的に白杖や盲導犬といった補助具を利用している。しかし、盲導犬は育成のコストと時間の問題から需要を満たすには至っていない。また段差、障害物、コーナーの検知といった機能に限定される。視覚障害者の歩行に必要な視覚情報は環境の認知と自己位置の推定である。これまで、情報技術を用いた歩行支援システムが提案されてきた。点字ブロックに IC タグを埋め込み杖についたセンサーで読み込むことで自己位置や段差を検知するシステムや、赤外線センサーなどを用いた障害物検知などであるが十分ではなく、何らかの環境整備や高価な装置が必要な場合が多くコストの問題からも実用・普及には至っていない。また、目的地までの道程における自己位置はいずれの装置でも認知できない。そこで、携帯端末とカメラといった安価な装置を利用して、周囲環境の認識しながらユーザーを安全に目的地までナビゲーションするブラインドナビゲーションシステムの開発が取り組まれている。既存の歩行ナビゲーションに加えて、周辺環境認識を行い歩行にかかわる環境認知のひとつに道路横断の際の安全確認がある。現状では、音響信号のない道路では車両のアイドリング音等の環境音を頼りに視覚障害者は横断の可否を判断しているが、騒音環境下など環境音による判断が困難な場合がある。既存の

センサーでは信号機までの距離が遠く信号灯火の色を認識することが困難であり、また、必ずしも横断歩道に歩行者が存在しない場合がある。そこで本稿ではブラインドナビゲーションでの利用を想定した、道路横断の状況認知の第一歩として、交差点上の歩行者位置・進行方向を推定する手法を提案する。また、実際のブラインドナビゲーションシステム構成の検討のために、道路横断支援システムの検討を行う。これにより、視覚障害者へのナビゲーションだけではなく、歩行者の進路上に存在する車両への注意喚起や、進行方向の信号機の情報の取得などが可能となる。位置推定については、GPS の位置情報と地図情報のマッチングを用いた手法が一般的である。しかし GPS では測位誤差 (約 10 m) のため、歩行者が存在する交差点は特定できても、交差点上の位置や進行方向までは特定できない [1]。そこで地図情報と対応付けられた街並画像と入力された街並画像をマッチングすることで、自己位置を推定手法が提案されている。このような画像ベースの自己位置推定手法は多数提案されている。その際、都市環境の 3 次元モデル [2], [3], [4] や、Flickr のようなジオタグ付き Web 画像データベース [5] を利用することが多い。しかし、密に観測されたデータから作られた高精度なモデルを必要とする手法は、頻繁なモデルの更新が必要となる。また Web 画像データベースの画像の大半は主要な場所や観光地で撮影されており、一般の住宅地の街並画像は非常に少ない。[1] では衛星画像を用いた、交差点上の歩行者位置推定手法を提案している。しかし、遮蔽により横断歩道を検出できない場合は推定が困難となる。そこで [6], [7], [8], [9], [10] では、街並画像データベースとして Google Street View (以下、GSV とする) を用いている。Google Street View では

¹ 神戸大学システム情報学研究科情報科学専攻 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

² 神戸大学自然科学系先端融合研究環 〒 657-8501 兵庫県神戸市灘区六甲台町 1-1

a) kawaguchi@me.cs.scitec.kobe-u.ac.jp

b) naoko.enami@port.kobe-u.ac.jp

c) ariki@kobe-u.ac.jp

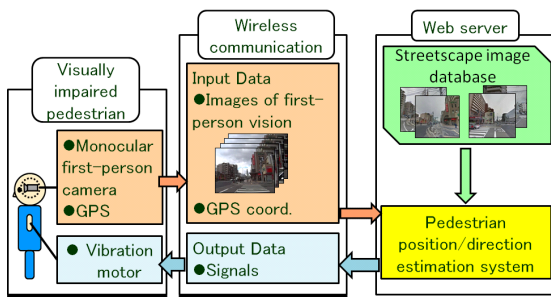


図 1 ブラインドナビゲーションシステムにおける道路横断補助の概要。

Fig. 1 Overview of our crossing support system.

地図情報に対応付けられた車載全方位画像が提供されている。多くの手法では携帯デバイスなどで撮影された画像とのマッチングにより、自己位置推定や画像検索を行っている。本研究でも街並画像データベースとして GSV を利用する。一方で、ブラインドナビゲーションでは対象物にカメラを向けることは出来ない。そのため、入力画像を撮影する機器を本研究ではウェアラブルデバイスとする。また、歩行者の進行方向向きは顔向きに依存するため、一人称カメラを用いる。一人称カメラは歩行者の目の位置に取り付けられたウェアラブルカメラであり、歩行者の視界の相似した一人称カメラ画像（以下、FPV 画像とする）が得られる（図 1）。つまり歩行者が進行方向を見ていると仮定すれば、歩行者の一人称カメラ画像と、街並画像データベース、GPS の位置情報のマッチングにより、歩行者位置、進行方向の推定を行う。本研究では、これらの装置を用いた道路横断支援システムの検討を行う。

GSV は広範囲をの地域を撮影している一方、撮影地点間の間隔が広く更新頻度が低い。そのため、撮影位置の変化、照明条件の変化、季節の違い、部分的遮蔽、建物の変化が生じる。さらに、マッチングの対象となる街並が道路の反対側に存在するといった問題が発生するため、単純な画像マッチングは困難である。安定した画像間のマッチングのために、本稿では FPV 画像列からパノラマ画像を生成し、FPV パノラマ画像と GSV 画像とのマッチングを行う。これにより、FPV 画像の撮影中に発生する移動物体による遮蔽を抑えることができる。次に、各画像から建物が写る領域（以下、建物領域とする）の各画素から SIFT 特徴 [11] を抽出し、SIFT flow [12] による建物領域間のマッチングを行う。SIFT flow マッチングでは類似した構造を持つ画像をマッチさせるため、部分的遮蔽や季節変化、建物の変化に頑健なマッチングが可能となる。さらに建物領域の画素間でのみマッチングを行うことで、建物の構造が似ている画像との類似度がより高くなり、高精度なマッチングを実現できる。実環境下で撮影された FPV 画像を用いた評価実験により、提案手法の有効性を示す。

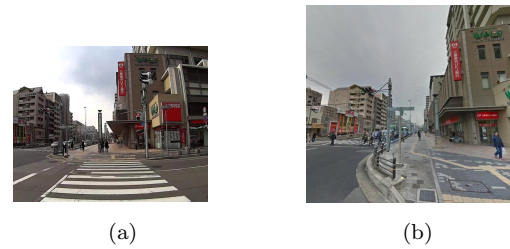


図 2 (a)FPV 画像の例。(b)(a) の撮影地点の近くで撮影された GSV 画像。

Fig. 2 (a)An example of the FPV image. (b)A GSV image captured near the position (a) was captured.

2. 道路横断支援システム

本研究で道路横断支援システムは、視覚障害者が日常的に利用することが想定される。そのため、システムには (1) 用いるデバイスが視覚障害者にとって利用しやすく、煩わしい操作を必要としないこと、(2) 高価で特別な機器を必要とせず、低コストで導入できること、といった要件が求められる。これらを踏まえて想定されるシステム構成の概要を図 1 に示す。以降本章ではシステムの構成や利用方法について述べる。

一人称カメラ：

入力画像を撮影するカメラは一人称カメラとする。従来の画像ベースの自己位置推定手法では、手持ち式カメラを用いて画像を収集する [1], [9]。しかし視覚障害者は杖などを扱うため、手にデバイスを持つことはできない。また、対象物に対してカメラを向けるのも困難である。そこで、Google Glass [13] や OrCam [14] のような一人称カメラであれば、歩行者の進行方向と関連のある顔向き方向の画像を得られる。一人称カメラにより撮影された画像 (FPV 画像) の例を図 2(a) に示す。一人称カメラにもステレオカメラを用いれば奥行き情報などが得られるが、ヘッドマウントデバイスの重量が増加する、また、高解像度のカメラを利用すれば高精度な推定が期待できるが、データ通信量やバッテリーの消費量が增大する。そのため長時間の撮影が可能な低解像度の単眼カメラの利用が望ましい。

GPS：

スマートフォン内蔵の GPS の位置情報の測位誤差は水平方向に 10m 程度とし、更新頻度は 1 秒間に 1 度とする。位置情報には緯度、経度、測位時刻、速度が含まれており、これらを用いて移動時の軌跡を取得する。交差点進入時の緯度と経度から視覚障害者がどの交差点に存在するかを特定し、進入直前までの軌跡からどの通りから進入したかを推定する。

無線通信機能：

ヘッドマウントデバイスは収集した情報を、スマート

フォンの無線通信を用いて Web サーバーに送信する。撮影画像のファイルサイズ (JPEG 圧縮のかかった VGA サイズ) をおよそ 100-150KB, フレームレートを 5fps と仮定すると 1 秒間に最大 750KB のデータ送信が発生する。一方, 現在のスマートフォンの送信時の通信速度は, 実測値でおよそ 5.5-12.56Mbps[15] である。そのため上記の圧縮のかかった画像を用いれば, サーバーに情報をオンラインで送信し, 推定を行うことは十分可能だと考えられる。

街並画像データベース :

提案システムでは街並画像データベースとして Google Street View(GSV) の画像を用いる。先述した通り GSV は撮影地点間の間隔が広い上に, FPV 画像と異なり車道から撮影されている。このため FPV 画像との撮影位置の違いによる見えの変化が大きいという問題点がある。一方, FPV 画像と同様に横断歩道手前で撮影した画像群からデータベースを構築すれば, より高精度なマッチングが期待できる。しかしそのようなデータベースを一から構築するのは, 人的コストの観点から現実的ではない。その点 GSV は更新頻度が低いという問題点はあるものの, 日本国内に関しては既にほとんどの地域の画像が提供されている。そのため, 低コストで導入でき, かつ様々な地域で利用できるというメリットがある。そこで GSV の画像を利用するが, 横断歩道付近の街並を写した画像のみを使用する。図 2(a) の FPV 画像の撮影地点付近で取得された GSV 画像を図 2(b) に示す。

歩行者位置・進行方向推定システム :

入力データと街並画像データベースから歩行者位置・進行方向を推定する。まず GPS の位置情報を基に視覚障害者がどの交差点にいるか, どの通りから交差点に進入したかを特定し, データベースからマッチングに用いる候補 GSV 画像を選択する。そして FPV 画像と GSV 画像のマッチングにより, 歩行者位置・進行方向を推定する。この処理の詳細については 3 章で述べる。

提案システムの利用方法 :

提案システムを利用する視覚障害者は, 前述した一人称カメラ, GPS, ネットワークが付随するヘッドマウントデバイスを装着する。視覚障害者は道路横断の際, 横断歩道の手前で立ち止まり安全確認を行うことが想定される。そこで提案システムでは横断歩道手前で立ち止まった時に, 視覚障害者の判断, もしくは位置情報の変化から, 撮影を開始すると仮定する。撮影中は進行方向を向き, 進行方向の視界と同様の画像を取得する。撮影は一定時間 (本稿では 10 秒と仮定) 経過すると自動的に終了し, サーバーにデータを送信する。

3. 一人称カメラ画像からの歩行者位置・進行方向推定

歩行者の GPS 位置情報と横断歩道手前で撮影された

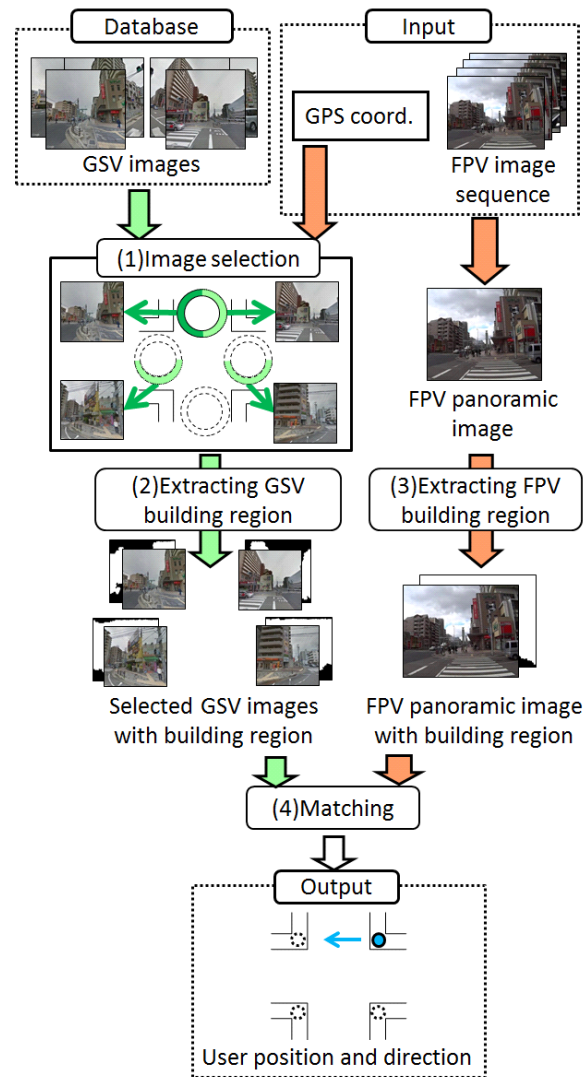


図 3 提案手法の流れ
 Fig. 3 Pipeline of our work.

FPV 画像と GSV 画像の対応付けにより交差点上の歩行者位置・進行方向を推定する手法を提案する。提案手法の流れを図 3 に示す。まず取得した GPS 位置情報によって, 歩行者がどの道路から交差点に進入したかを特定する。候補となる歩行者位置は 4 箇所となる (図 4)。候補歩行者位置に対応する GSV 画像をデータベースより選択する。選択された GSV 画像から建物領域を抽出する。次に FPV 画像列から Microsoft Image Composite Editor[16] を用い, パノラマ画像を生成する。時系列間で隣り合う画像でマッチング率が低い画像は除去されるため, 車両などによる一時的な遮蔽の影響を低減させることが可能となる。GSV 画像と同様に建物領域を抽出する。最後に, GSV 候補画像と FPV パノラマ画像のマッチングを行い, 交差点上の歩行者位置・進行方向を推定する。3.1 節では, 撮影位置の違いから生じる見えの違いに対応するための建物領域抽出方法について述べる。3.2 節では, 抽出された建物領域間

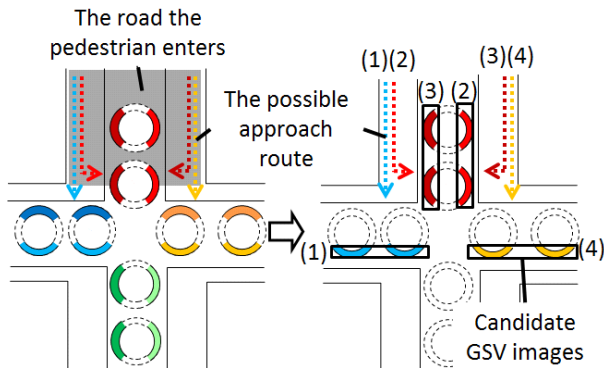


図 4 GPS による歩行者候補位置の選択. 番号付きの矢印は候補となる進入経路を示しており, 同じ番号の候補 GSV 画像に対応する.

Fig. 4 We narrow down candidate images according to the entering direction of the pedestrian and then use the remaining images for matching. The arrow with the number indicates the candidate entering route and corresponds to the candidate GSV images with the same number.

のマッチングについて

3.1 画像からの建物領域の抽出

GSV 画像は車道から車載カメラにより撮影され, FPV 画像は歩道から人の視線位置より撮影される. そのため, 同一地点を撮影された画像であっても見え方が著しく異なる (図 2). また, 画像中に多数存在する道路や空といった領域はどの画像にも存在し, マッチングによる位置推定を困難にする要因となる.

しかしながら, 画像中にどのような建物が存在するかは不明であり, 建物領域を抽出することは困難である. GSV 画像の更新頻度の低さから, 必ずしも同じ建物が存在するとは限らない. そこで, 本稿では道路領域と空領域を除去することにより, 建物領域を抽出する. まずはじめに GSV 画像からの建物領域抽出について述べる. GSV 画像をスーパーピクセルを抽出する. スーパーピクセルの抽出には SLIC[17] を用いる. スーパーピクセルの最大個数は 3200 個とした. 次に, GSV 画像の上端のスーパーピクセルを空領域の初期領域とする. 同様に下端のスーパーピクセルを道路領域の初期領域とする. (図 5(c)). この各初期領域と候補スーパーピクセルとの相関度により, 候補スーパーピクセルに対して初期領域と同じラベルを与える. 初期領域と候補スーパーピクセルとの相関度の算出の際には, GSV 画像は広域を撮影した画像であるため GSV 画像を短冊状の画像 (短冊画像) T_j に分割し, 短冊画像ごとに行う. 分割数は $4(j = 1, 2, 3, 4)$ とした. 各短冊画像の初期空領域とした全スーパーピクセルの CIE Lab 色データ集合から, 平均ベクトル μ_j^{sky} ($j = 1, 2, 3, 4$) と分散共分散行

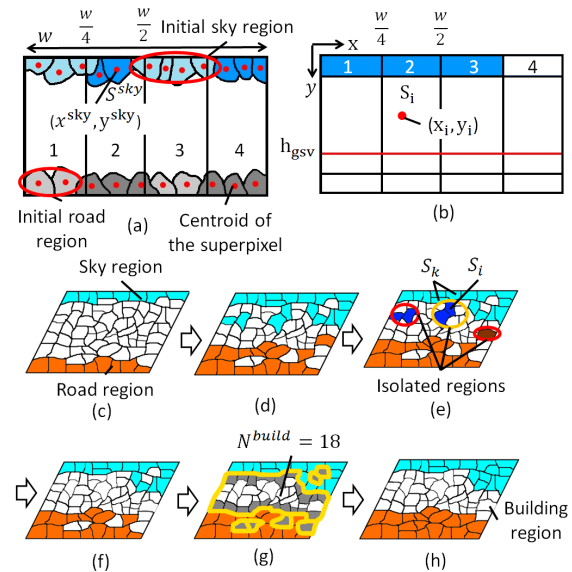


図 5 GSV 画像からの建物領域の抽出. : (a) 初期領域分割の例. 赤い点はスーパーピクセルの重心を表す. (b) 青い矩形は注目スーパーピクセル S_i と初期空領域を表す. (c) 道路と空の初期領域. (d) マハラノビス距離によるラベル割り当て結果. (e) 候補領域から孤立したスーパーピクセルの除去. (f) 孤立スーパーピクセルへのラベル再割り当ての結果. (g) 候補建物領域が孤立点であれば除去する. (h) 最終的な建物領域抽出結果.

Fig. 5 Extracting building regions from the GSV image. : (a) An example of splitting initial regions. Red dots indicate the centroids of the superpixels. (b) Blue rectangles indicate initial sky regions compared with the superpixel S_i . (c) Initial regions of the road and the sky. (d) The result of label assignment after comparison by the Mahalanobis distance. (e) Superpixels surrounded by ellipses are regarded as superpixels isolated from initial regions. (f) The result of label re-assignment of the labels of isolated superpixels. (g) Regarding each building region indicated by the yellow frame, we count the number of the superpixels N^{build} which are not adjacent to the road or sky regions (white superpixels). (h) Final building regions.

列 Σ_j^{sky} を算出する. 初期道路領域についても同様に, 色データ集合の平均ベクトル μ_j^{road} と分散共分散行列 Σ_j^{road} を算出する. j は短冊画像のインデックスとする. 次に, 候補スーパーピクセルの Lab の平均値集合と初期領域の色データ集合との相関をマハラノビス距離により算出し, 候補スーパーピクセルに対してラベルを割り当てる. i 番目の候補スーパーピクセル S_i を属する短冊画像および隣接する短冊画像の初期領域と比較する. 例えば, 候補スーパーピクセル S_i の重心が短冊画像 T_2 に属する場合, T_2 の初期領域, および隣接する T_1 と T_3 の初期空領域と比較する (図 5(b)). 次式により, 候補スーパーピクセル S_i の CIE Lab 色集合の平均ベクトル μ_i と, 初期空領域の μ_j^{sky} とのマハラノビス距離を算出し, 閾値 c^{sky} 以下であれば候補

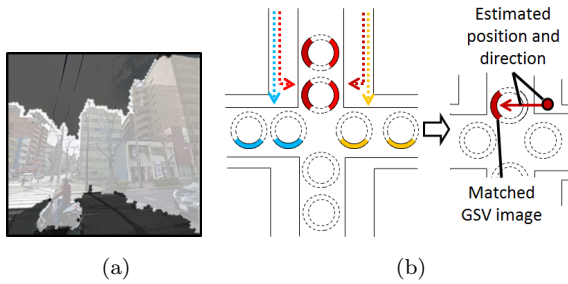


図 6 (a) 建物領域の抽出結果の例. (b) マッチング結果からの FPV 画像の撮影位置と撮影方向の推定.

Fig. 6 (a) An example of the result of extracting building regions. (b) Since the capturing position and direction can be identified from the matching result, we can estimate the pedestrian position and direction at the intersection.

スーパーピクセルを空領域とする.

$$\sqrt{(\mu_i - \mu_j^{sky})^T (\Sigma_j^{sky})^{-1} (\mu_i - \mu_j^{sky})} < c^{sky} \quad (1)$$

c^{sky} は実験的に $c^{sky} = 0.1$ と定めた. 同様に, 次式により初期道路領域とのマハラノビス距離を算出し, 閾値 c^{road} 以下であれば候補スーパーピクセルを道路領域とする.

$$\sqrt{(\mu_i - \mu_j^{road})^T (\Sigma_j^{road})^{-1} (\mu_i - \mu_j^{road})} < c^{road} \quad (2)$$

c^{road} は実験的に $c^{road} = 0.04$ と定めた. 初期領域との相似度によるラベル割り当て結果を図 5(d) に示す. さらに, 空・道路領域において孤立したスーパーピクセルは誤対応である可能性が大きいので除去をする (図 5(e)). まず, 空領域候補から孤立点を探索する. 次に孤立点 S_i の隣接スーパーピクセル S_k との輝度値の相関度を算出する. S_i の輝度の平均値を a_i , S_k の輝度の平均値を a_k とする. $|a_i - a_k|$ が閾値 α より大きければ, S_i のラベルを削除する. 本稿では実験的に $\alpha = 1.0$ と定めた. 結果の例を図 5(f) に示す. また, 孤立した建物領域においても同様のラベルの割り当てを行う (図 5(g)). 道路領域と空領域以外の領域を建物領域とする. は最終的な建物領域抽出結果を図 5(h) に示す.

次に FPV 画像からの建物領域の抽出について述べる. FPV パノラマ画像を生成後, 同様に建物領域を抽出するが, FPV 画像では撮影位置から建物が遠いため, 空領域が少ない. そのため, 道路領域のみ抽出し, その他の領域を建物領域とする.

3.2 領域マッチングによる撮影位置と進行方向の推定

抽出された建物領域間のマッチングに SIFT flow[12] を用いる. SIFT flow では画素ごと密なマッチング GSV 画像を絞り込み, 図 6(a) のように建物領域を抽出した後, SIFT flow[12] を用いてマッチングを行う. まず, FPV パノラマ画像の建物領域に属する各画素と GSV 画像の建物領域の画素間のマッチングを行う. SIFTflow では全画素

間のマッチング結果の総エネルギーが小さい画像間が相関度が高いとみなせる. そこで, マッチング候補の全 GSV 画像についてマッチングコストを算出し, コストが最小となる画像をマッチング結果とする. マッチした GSV 画像の撮影位置から, FPV 画像が交差点上の位置, どの向きから撮影されたかを推定し, 交差点上の歩行者位置・進行方向が得られる.

4. 実験

4.1 データセット

5 本章では 2 章のシステム構成を元に構築したデータセットを構築した. 兵庫県神戸市灘区の 3 か所の交差点で, 単眼の一人称カメラを用いて FPV 画像を収集した. 撮影時期は 2015 年 2 月の複数日で, 撮影時刻は午前 (10:30 から 12:00), または午後 (13:30 から 15:00) である. 天候は晴天, 曇天または小雨の 3 つとする. 撮影者は横断歩道の前で立ち止まり, 進行方向を向いて約 10 秒間停止し, その間の FPV 画像例を用いる. FPV 画像の解像度は 640x480 ピクセルで, フレームレートは 5fps だった. 位置情報により歩行者が停止したと思われるフレームから 50 フレームを抽出し, 1 シーケンスとした. 各交差点につき 16 通りの経路で各 6 回撮影したため, 1 交差点で計 96 シーケンス, 全体で 288 シーケンスを取得した. 次に, GSV 画像は 1 交差点につき 8 地点の GSV 画像を Google Street View Image API[18] を用いて取得した. GSV 画像は全方位を複数のカメラで撮影した画像で構成されているため, その中から左右の街並の正面を撮影した街並正対画像と街並正対画像と隣接する画像の計 4 画像を取得した. 取得した GSV 画像の解像度は 640x640 ピクセル, GSV 画像に存在するロゴなどを取り除くために下部の 30 ピクセルを削除した. なお, GSV 画像は 2015 年 3 月から 4 月にかけて撮影された. GSV 画像と FPV 画像には, 撮影時刻の違いによる輝度変化, 撮影位置の違い, 建物の変化といった見えの変化が生じた. 加えて, GSV と FPV 画像はともに, 電柱, 街路樹, 車両などによる遮蔽が頻繁に生じた. 取得した GSV 画像に位置番号と向き番号を割り当てた. 位置番号は画像の交差点上での位置を示し, 向き番号は画像の交差点上の向きを示す. FPV パノラマ画像はマッチした画像の重心と向き番号が, 真値に一致した場合に正しくマッチしたとする.

4.2 建物領域抽出の評価

撮影位置の変化に頑健なマッチングを実現するために本手法では建物領域の抽出を行う. その目的を達成できたかを抽出する領域を変えて評価した. GSV 画像については,

- (1) オリジナル画像
- (2) 建物領域と空領域を含む画像
- (3) 建物領域と道路領域を含む画像

表 1 建物領域抽出結果ごとのマッチング結果

Table 1 The matching results.

(a)

GSV	FPV	
	(1) オリジナル画像	(2) 道路・建物領域
(1) オリジナル画像	87.5%	94.4%
(2) 空・建物領域	75.7%	88.9%
(3) 道路・建物領域	72.9%	67.4%
(4) 建物領域	64.9%	63.2%

表 2 マッチング手法ごとの正解率

Table 2 The matching result using other matching method.

手法	FPV	
	(1) オリジナル画像	(2) 道路・建物領域
(1)SIFT	82.3%	85.1%
(2)SURF	67.4%	69.4%
(3)DAISY flow	80.9%	91.3%
(4)SIFT flow	87.5%	94.4%

(4) 建物領域のみの画像

の 4 通りの GSV 画像を使用した。また FPV パノラマ画像については、

(1) オリジナル画像

(2) 建物領域と空領域を含む画像

の 2 通りの FPV 画像を使用した。これらの全組み合わせによる評価実験を行った。正しく対応付けられた FPV パノラマ画像の割合 (正解率) を表 1(a) に示す。FPV 画像については、地面領域を除去すると概ね正解率が上昇した。道路領域は車両による遮蔽が大きいため、除外することでマッチングの頑健性が向上したと考えられる。一方 GSV 画像については、地面領域や空領域を除去すると正解率が低下した。特に空領域を除去した場合、除去しない場合と比べ概ね 14% から 25% 低下した。領域抽出の誤抽出が原因であると考えられる。

4.3 他のマッチング手法との比較

マッチング手法の評価を行うために、他のマッチング手法 (SIFT[11], SURF[19] および DAISY flow マッチング) と SIFT flow マッチングを比較した。各手法を (1) GSV, FPV ともにオリジナル画像 (2) FPV は地面領域除去, GSV は除去なしの画像を用いて比較した。SIFT および SURF の特徴点間マッチングでは、2 枚の画像で検出された局所特徴量をマッチさせ、[11] の距離比率規則を満たす対応だけを残した。残った対応点数を FPV パノラマ画像で検出されたキーポイントの数で割った値をマッチ率として使用した。このマッチ率が最大となる GSV 画像と対応付けられた画像とした。DAISY flow マッチングでは SIFT flow と同様の方法でマッチングを行ったが、SIFT 特徴量の代わりに DAISY 特徴量を用いた。各手法を用いた時の正

解率を表 2 に示す。他手法に比べ、SIFT flow マッチングの正解率が最も高く、SIFT flow マッチングは有効であったといえる。通常の SIFT マッチング、DAISY flow マッチングがこれに続く結果となったが、SURF マッチングは他と比べ大きく正解率が低下した。そのため、密な画素間のマッチングが適しているといえる。

5. まとめ

本論文では視覚障害者ためのブラインドナビゲーションシステムの構築のため、道路横断支援システム検討と一人称カメラで撮影された画像と Google Street View の画像をマッチさせることで、交差点上の歩行者の位置・進行方向を推定する手法の提案を行った。さらに検討した撮影方法やシステム構成に従い、一人称カメラ画像データセットを構築した。撮影位置や照明条件、季節、建物の変化による見えの違いや部分的遮蔽に対処するため、我々は建物領域の抽出と SIFT Flow マッチングを用いた。建物領域抽出の有効性の検証実験では、FPV 画像については道路領域を除去した方が正解率が高く、建物領域抽出の有効性を示すことができた。本稿では今後は視覚障害者にカメラを装着して画像を収集し、カメラのブレといった視覚障害者の歩行特有の問題点の検証を行う。

参考文献

- [1] G. Fusco, H. Shen, J. Coughlan: Self-Localization at Street Intersections, In: CRV, pp. 40–47 (2014)
- [2] C. Arth, M. Klopschitz, G. Reitmayr, D. Schmalstieg: Real-time self-localization from panoramic images on mobile devices, In: ISMAR, pp. 37–46, 2011.
- [3] B. Klingner, D. Martin, J. Roseborough: Street View Motion-from-Structure-from-Motion, In: ICCV, pp. 953–960, 2013.
- [4] Y. Li, N. Snavely, D. Huttenlocher, P. Fua: Worldwide pose estimation using 3D point clouds, In: ECCV, vol. I, pp. 15–29 (2012)
- [5] J. Hays, A. A. Efros: Im2gps:estimating geographic information from a single image, In: CVPR, pp. 1–8 (2008)
- [6] A. Torii, Y. Dong, M. Okutomi, J. Sivic, T. Pajdla: Efficient Localization of Panoramic Images Using Tiled Image Descriptors, IPSJ Trans. on CVA, vol.6, pp.58–62 (2014)
- [7] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla: 24/7 Place Recognition by View Synthesis In CVPR, 2015:
- [8] A. R. Zamir, M. Shah: Accurate image localization based on google maps street view, In: ECCV, vol. IV, pp. 255–268 (2010)
- [9] G. Schindler, M. Brown, R. Szeliski: City-scale location recognition, In: CVPR, pp. 1–7 (2007)
- [10] A. Taneja, L. Ballan, M. Pollefeys: Never Get Lost Again: Vision Based Navigation using StreetView Images, In: ACCV, pp. 1–15 (2014)
- [11] D. G. Lowe: Distinctive image features from scale-invariant keypoints. International journal of computer vision. 60, pp. 91–110, 2004.
- [12] C. Liu, J. Yuen, A. Torralba, J. Sivic, W. T. Freeman:

Sift flow: Dense correspondence across different scenes.
In: ECCV, pp. 28–42 (2008)

- [13] Google Glass, <https://www.google.com/glass/start/>
- [14] OrCam - See for Yourself, <http://www.orcam.com/>
- [15] ICT 総研—市場調査・マーケティングカンパニー, <http://ictr.co.jp/report/20150904.html>
- [16] Image Composite Editor, <http://www.ncbi.nlm.nih.gov>
- [17] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk,.: SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. on PAMI, vol.11, no. 34, pp. 2274–2282 (2012)
- [18] Google Street View Image API, <https://developers.google.com/maps/documentation/streetview/>
- [19] H. Bay, T. Tuytelaars, L. Van Gool,.: SURF: Speeded up robust features, In: ECCV, pp. 404–417 (2006)