

視覚障がい者支援を目的とした車体制御

Seeing-eye Dog Emulation by Android Terminal

松永 倫幸[†], 江口 一彦^{††}

Noriyuki MATSUNAGA, Kazuhiko EGUCHI

Abstract: Strongly impaired people in vision need seeing-eye dogs, however, it takes long time and costs a lot of money to train them. In addition, not all candidate dogs may be grown as seeing-eye dogs. Therefore, available number of seeing-eye dogs is very few and far from to satisfy actual demand.

Some efforts are made to develop a robotic seeing-eye dog. In this paper, emulation of seeing-eye dog using Android terminal and an autonomous run are performed.

1. はじめに

平成 18 年身体障がい者実態調査結果¹⁾によると視覚障がいの総数は 31 万人に上る。視覚障がい者は道路を通行する際に白杖もしくは盲導犬の利用が義務付けられている。盲導犬は障害物を検知し、道路情報を取得する。それらの情報を得ることにより、利用者は目的地へ向かうことができる。しかしながら、盲導犬は必要とする視覚障がい者に対し、供給数が限られているのが現状である。盲導犬育成には 4 か月から 12 か月にわたる訓練期間、300 万円と高額な育成費用が必要となる。また訓練した候補犬全てが盲導犬になれるわけではないため、平成 25 年 1 月 1 日現在の盲導犬実働数は 1043 頭²⁾となっている。そのため盲導犬を必要としているが利用できない障がい者やアレルギーなどにより盲導犬を利用することのできない障がい者がいる。

盲導犬を工学的に再現する試みは日本精工株式会社³⁾などの企業で行われている。再現する機能として、障害物の回避や段差の認識、近くの目標物への誘導などが挙げられる。これらの機能を再現するためにはカメラやセンサなどを用い、障害物の認識、利用者の誘導を行う車体制御システムが必要となる。

上記の研究は高価で大掛かりであるため、本研究においてはカメラや方位センサを備え、外部との通信機能を保有し、安価な Android 端末を用いた視覚障がい者支援を目的とした車体制御システムを検討した。また、進路の認識手法確立に影響しない

ためサイズをスケールダウンして研究を行った。

2. システム概要

本研究において考案した車体制御システムの概要を図 2.1 に示す。

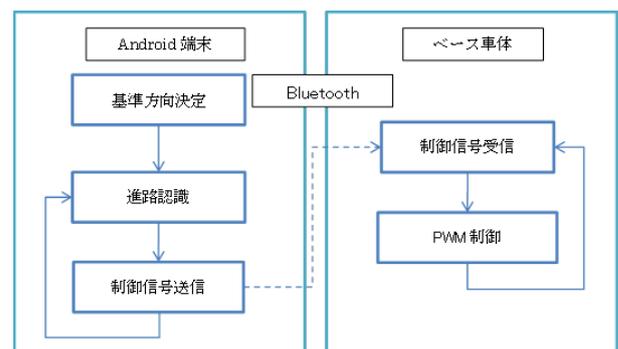


図 2.1 車体制御システム概要

Android 端末付属のカメラで動画を撮影し、画像処理を施し進路を検出する。このとき撮影した画像を 1/2 に縮小し、画像処理後に元のサイズへ拡大する。画像処理を縮小した画像に施すのは処理速度を向上させるためである。進路検出を行い、検出された進路を元に制御信号を作成し、Bluetooth 通信にてベース車体へ送信する。ベース車体側において制御信号を受信し、PWM 制御によって車体を制御する。

上記の処理を繰り返すことにより障害物を回避し、基準方向にベース車体を制御する。本研究において基準方向とはシステ

[†] 愛知工業大学 工学研究科電気電子工学専攻(豊田市)

^{††} 愛知工業大学 工学部電気学科 (豊田市)

ム起動時にベース車体が向いていた方向とする。

作成したベース車体を図 2.2 に示す。

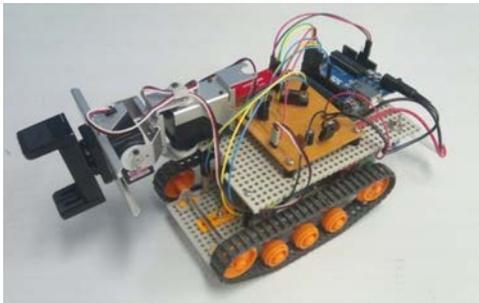


図 2.2 ベース車体

3. 進路認識処理手法

3.1 画像処理

進路認識を行うために画像に処理を加える。画像処理の概要を図 3.1 に示す。

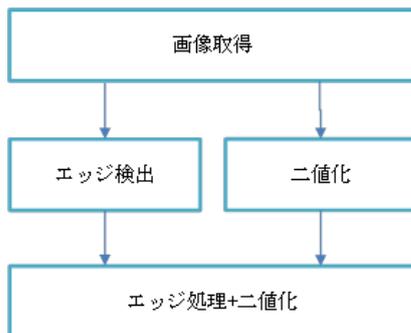


図 3.1 画像処理概要

画像を取得し、エッジ検出および二値化処理を並列して行う。それぞれの処理を行った画像を論理積したのちに進路検出処理を行う。取得した画像を図 3.2 に示す。



図 3.2 取得画像

3.1.1 エッジ検出

エッジ検出とは画素値が急激に変化する点を検出し、画像中の境界を検出する処理である。本研究では Canny 法を用いてエッジ検出を行った。Canny 法の手法は以下の通りである。

- 1: ガウシアンフィルタによる平滑化
- 2: Sobel フィルタによるエッジ強度の計算
- 3: 細線化処理
- 4: ヒステリシス閾処理

ガウシアンフィルタは式 (1) で表される。

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (1)$$

σ の値が小さいほど平滑化の効果は小さくなり、大きいほど効果が大きくなる。

平滑化した後に Sobel フィルタによるエッジ強度の計算を行う。注目画素を中心とし周囲の画素値に対し、以下の垂直方向、水平方向の 2 つの係数行列 (式 (2)) をそれぞれ乗算し、結果を合計する (式 (3))。ここでの注目画素とは中央の画素とする。

$$\begin{matrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\ \text{水平方向 } f_x & \text{垂直方向 } f_y \end{matrix} \quad (2)$$

$$\text{注目画素 } f = \sqrt{f_x^2 + f_y^2} \quad (3)$$

ガウシアンフィルタ, Sobel フィルタによる処理を行った後にエッジと鉛直方向の隣接画素 2 つとエッジ強度を比較し、極大でなければ 0 とする細線化処理を行う。その後、ヒステリシス閾処理を行うことによりエッジ、非エッジの判定を行う。

ヒステリシス閾処理では高低 2 つの閾値を与え、高い閾値より大きいエッジ強度の画素はエッジと判定し、低い閾値未満のエッジ強度の画素は非エッジと判定される。低い閾値以上、高い閾値以下のエッジ強度の画素ではエッジに結合している画素ではエッジと判定される。エッジ検出画像を図 3.3 に示す。

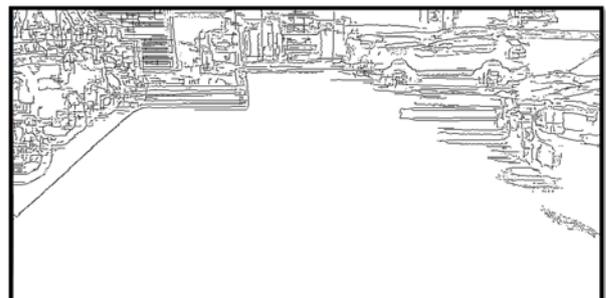


図 3.3 エッジ検出画像

3・1・2 二値化

二値化とは画素の輝度値が閾値以上なら白とし、未満なら黒とする処理である。そのため閾値を変化させることにより処理後の画像が変化する。本研究では判別分析法を用いて閾値を決定した。

判別分析法とは分離度と呼ばれる値が最大となる閾値を求め二値化を行う手法である。分離度はクラス間分散とクラス内分散の比から求める。クラスは輝度値が閾値未満となる黒クラスと閾値以上となる白クラスからなる。黒クラスの画素数を ω_1 、平均を m_1 、分散を σ_1^2 とし、白クラスの画素数を ω_2 、平均を m_2 、分散を σ_2^2 とする。同様に全体の素数を ω_t 、平均を m_t 、分散を σ_t^2 とする。このときのクラス内分散 σ_w^2 とクラス間分散 σ_b^2 をそれぞれ式 (4)、式 (5) に示す。

$$\sigma_w^2 = \frac{\omega_1\sigma_1^2 + \omega_2\sigma_2^2}{\omega_1 + \omega_2} \quad (4) \quad \sigma_b^2 = \frac{\omega_1\omega_2(m_1 - m_2)^2}{(\omega_1 + \omega_2)^2} \quad (5)$$

このときの全分散 σ_t^2 を式 (6) に示す。

$$\sigma_t^2 = \sigma_b^2 + \sigma_w^2 \quad (6)$$

分離度はクラス間分散とクラス内分散の比なので式 (7) となる。

$$\frac{\sigma_b^2}{\sigma_w^2} = \frac{\sigma_b^2}{\sigma_t^2 - \sigma_b^2} \quad (7)$$

分離度が最大となる閾値 t を求めるが、全分散 σ_t^2 は閾値に依存しないのでクラス間分散 σ_b^2 が最大となる閾値を求める。このときクラス間分散の分母も一定であるため分子である $\omega_1\omega_2(m_1 - m_2)^2$ が最大となる閾値を用いる。判別分析法を用いた二値化処理画像を図 3.4 に示す。



図 3.4 二値化処理画像

図 3.4 二値化処理画像と図 3.3 エッジ検出画像を論理積することにより画像内の境界や障害物を検出する (図 3.5)。

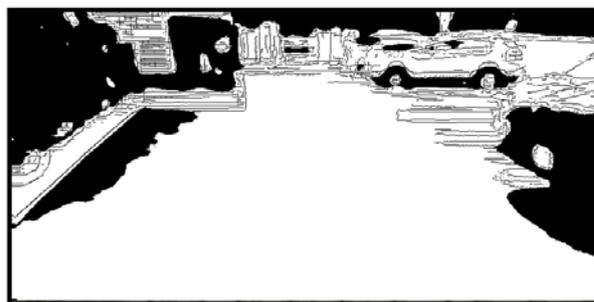


図 3.5 エッジ検出, 二値化処理合成画像

3・2 進路認識

進路認識は画像中の連続した白の画素数によって行う。画像の X 軸に相当する幅を X 、Y 軸に相当する高さを Y と定義し、 $(0, Y)$ から (X, Y) まで走査する。白い画素が連続しラインを構成している領域を白領域ラインとし、両端の座標を記録する。 (X, Y) まで達したら $(0, Y+1)$ から再び走査を開始する。この処理を $(X, 0)$ に達するまで繰り返し、画面中の白領域ラインを記録する (図 3.6)。

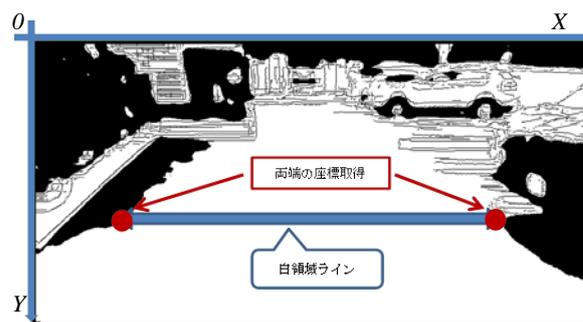


図 3.6 座標軸関係図

進路認識によって得られた白領域ラインから進路を選出する。選出法は白領域ラインの Y 座標 y によって 2 つに分けられる。

記録されている白領域ラインのうち、Y 座標が等しいラインで比較を行う。Y 座標が $0.95Y$ より大きい、すなわち画像下 5% の領域においては最も長い白領域ラインを構成するライン両端座標を選出し、中点座標を求める。画像下の領域で進路の方向性を決定し、残りの領域では選出された進路の中点の延長線上に最も近い中点を持つ白領域ラインを選出する。

認識された進路幅が狭く車体が進行できない場合、進路ではないと判断し進路認識処理を中断する。判断の基準値 β は撮影範囲における車幅の割合である。進路認識中断の判定処理を行う範囲を図 3.7 に示す。

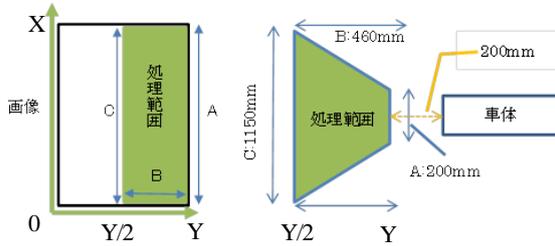


図 3.7 中断判定処理範囲

下底: $Y = 200\text{mm}$, 上底: $Y/2 = 1150\text{mm}$ の台形であるため、 $Y/2 \sim Y$ の任意の高さ y の撮影範囲 L は (8) で示される。

$$L = -1900 * \frac{y}{Y} + 2100 \quad (8)$$

ベース車体の車幅は 200mm なので基準値 β は式 (9) で表される。

$$\beta = \frac{200}{L} \quad (9)$$

基準値が過少であると進路を誤検出する可能性が高まるので画像中央: $Y/2$ における車幅が占める割合を最小基準値とした。進路認識の結果を図 3.8 に示す。進路両端を赤色、進路中央を青色で示す。

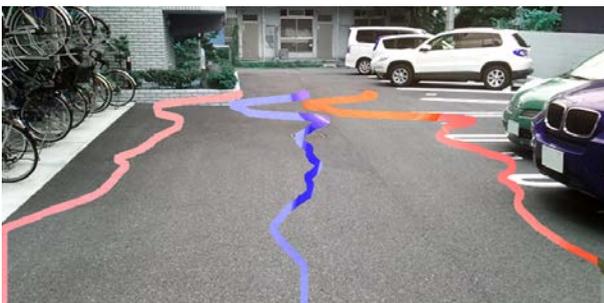


図 3.8 進路認識結果

3.3 車体制御ライン

進路認識の結果から車体の制御を行う車体制御ラインの生成を行う。生成法を図 3.9 に示す。

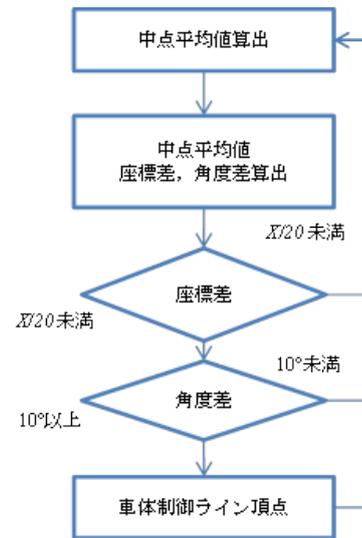


図 3.9 車体制御ライン生成法

隣接する進路中点の平均値である中点平均値を算出する。算出後、隣接する中点平均値の X 座標差: Δave と 2 点が成す角度: α を求める。 Δave と α を画像下部から順に比較していき、差分 Δave , $\Delta \alpha$ を求める。 Δave の差異が $X/20$ 以下であれば頂点を省略し、車体制御ラインを簡素化する。角度については 10° 以下の差異ならば直線であると判断し、頂点を省略する。車体制御ラインの生成結果を図 3.10 に示す。

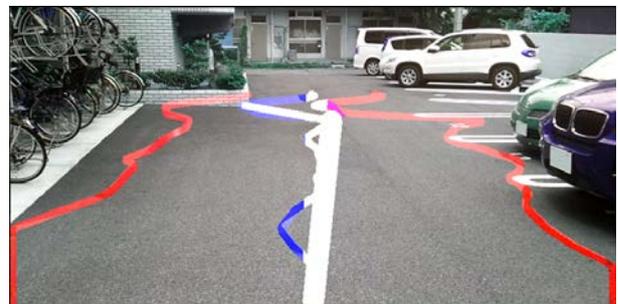


図 3.10 車体制御ライン生成結果

進路両端を赤色、進路中央を青色、車体制御ラインを白色で示す。ベース車体の制御には図 3.10 の白太線を用い、垂直に対する角度がベース車体の制御角となる。

4. 車体制御

車体の制御方法は制御角度によって異なる。分類方法を表 4.1 に示す。

表 4.1 制御角に対する制御法の分類

制御角	制御法
10° 未満	基準方向へ軌道修正
10° 以上 60° 未満	エリア制御
60° 以上	進路探索

制御角が10°未満である場合、ベース車体はシステム起動時に設定された基準方向へ軌道修正を行う。10°以上60°未満の場合のエリア制御、60°以上の場合における進路探索については後述する。

4・1 エリア制御

車体を中心に15*7マスの仮想エリアを設置し、障害物や目標地点の管理をエリア単位で行う。1マスの大きさは車体サイズから46mm*200mmとした。これは車幅2*車体長に相当する。仮想エリアと撮影範囲の関係を図4.1に示す。

緑色が撮影範囲であり、赤色が障害物、黄色が目標エリア、青が車体を示す。車体はAndroid端末の位置を車体位置として扱う。

障害物、目標エリア、車体の位置関係をエリア単位で管理する。

図4.1では目標エリアを(1,2)、障害物エリアを(-3,3)~(0,2)、車体の現在位置を(0,0)としている。各エリアが設定されると車体は現在位置から目標エリアへ移動を開始する。

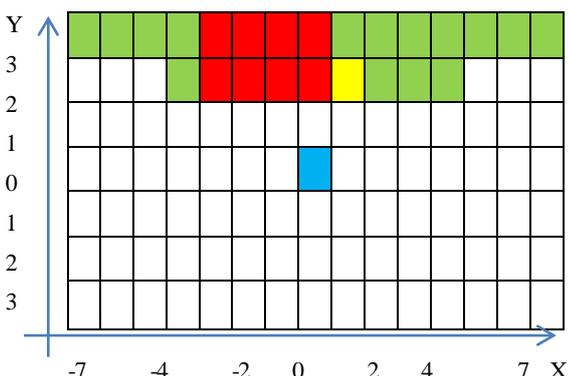


図4.1 仮想エリア

目標エリアに到達すると車体の位置を原点とし、位置関係を再び記録する。この処理を行うことにより、カメラの範囲外となった障害物の位置関係を把握することが可能となる。

4・2 障害物検知

図4.2の薄い赤色に着色しているエリアは障害物の検知エリアである。検知エリアのマスにポイントを割り当て障害物が存在する場合、ポイントを付与する。そしてポイントが基準値以上である場合、車体を停止させる。今回、基準値は6ポイントとした。

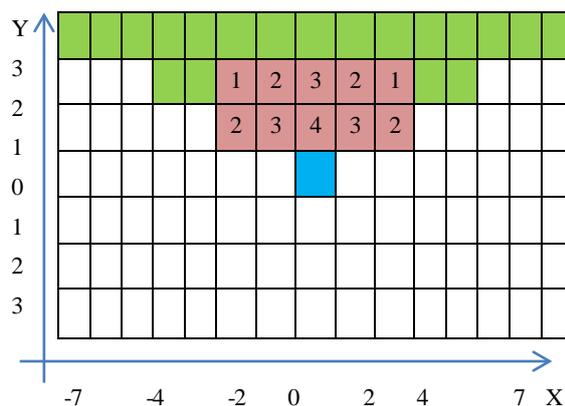


図4.2 仮想エリアによる障害物検知

4・3 進路探索

制御角度が60°以上であった場合や障害物により進路が構築されない場合、車体を停止させ車体前方に設置されたサーボモータにて進路の探索を行う。車体停止後、サーボモータを用いて左右15°、30°および正面0°の方位で進路検出の処理を行う。0°の進路検出は誤検出による障害物の認識に対応するためである。すべての方位で進路が検出されなかった場合においては停止し、車体制御を終了する。複数の進路が検出された場合、修正方向が小さく、長さが長く検出されている進路を選出することが望ましい。検出された進路の長さ、方位、平均ライン幅からリストを作成する(表4.2)。

表4.2 検出進路リスト

グループ	0°	左15°	左30°	右15°	右30°
ポイント	0	0	0	0	0
長さ	16	98	39	15	6
方位	90	30	50	2	12
ライン幅	45	62	4	97	86

表4.2のリストをグループごとに長さに対して降順ソートを行い、ポイントを付与する。同様に方位に対して昇順ソート、ライン幅に対して降順ソートを行い、ポイントを付与する。最後にポイントについて昇順ソートを行い、最もポイントの低い方位へ車体を軌道修正する。

4・4 軌道修正

車体の進行方向が決定されるとBluetooth通信によってベース車体の軌道修正が行われる。車体制御は8bitの信号で行われ、上位4bitが左輪、下位4bitが右輪の制御を行う。図4.3に制御信号の概要を示す。

7	6	5	4	3	2	1	0
左輪		左輪速度制御		左輪		左輪速度制御	

図4.3 制御信号

7bit が左輪進行方向を表し、4~6bit によって左輪進行速度を表す。下位 4bit においても同様である。進行速度は式 (10) で求める。

$$V = V_{MAX} * \frac{6 - \theta}{6} \quad (10)$$

V : 進行速度
 V_{MAX} : 最高速度
 θ : 制御角度

5. 実験結果

Android 端末をベース車体に設置し、撮影した画像から進路認識を行った (図 5.1)。

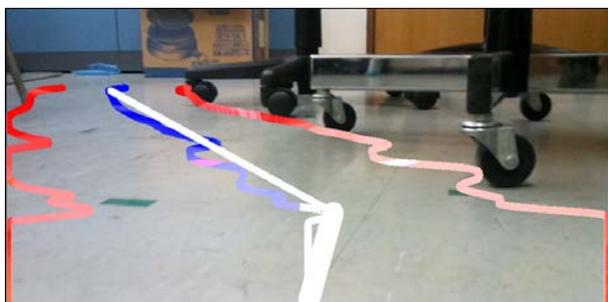


図 5.1 実験結果

障害物を回避する進路を生成している。また、凹凸のある進路中央ラインから直線の車体制御ラインを構成している。

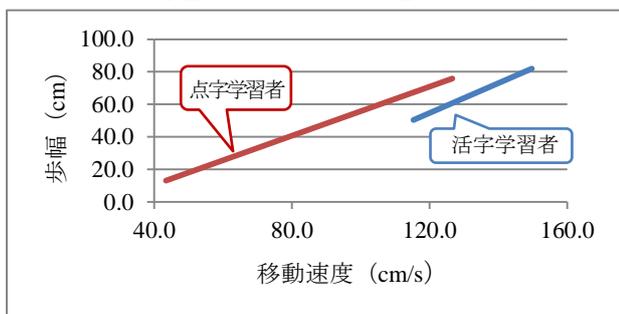


図 5.2 移動速度と歩幅平均値⁴⁾

鈴木等が行った身長に対する歩行速度と歩幅の測定結果⁴⁾を 図 5.2 に示す。測定の被験者は活字学習者と点字学習者のグループに分けられ、徳島県立盲学校の小・中・高等部 (職業訓練課程含む) の児童、生徒、教員を対象とし、盲学校内の屋外歩行者用通路と四国旅客鉄道管内の二軒屋駅プラットフォームにて測定した。

日常空間である校内と不安を感じる場所である駅⁵⁶⁾との平均値が日常生活における歩行速度、歩幅の平均値であると考え、歩幅を算出した。

今回のシステムにおける平均 FPS は 3.539 となり 282ms 毎に

制御信号が更新される。制御信号の更新間隔と歩幅の関係を図 5.3 に示す。

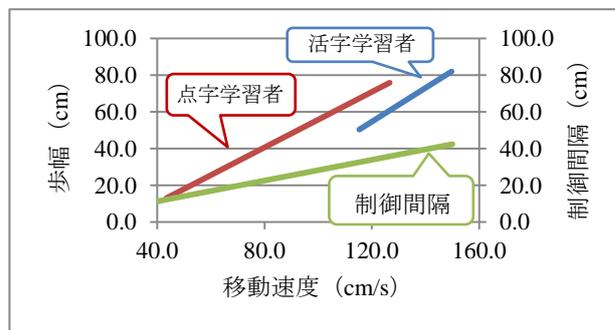


図 5.3 移動速度と歩幅、制御間隔平均値

点字学習者、活字学習者共に歩幅に対して制御信号の更新間隔が短くなる。そのため一歩以内に制御信号が更新されるため誘導を行うことができると考える。しかし 40cm/s 付近では歩幅と制御間隔ほぼ等しくなるため処理速度の向上は必要である。

6. 結論

Android 端末による障害物の認識、進路探索、車体制御情報の生成に成功した。また Bluetooth 通信によりベース車体の制御を行い、障害物を回避することに成功した。

処理速度は視覚障がい者の誘導を行うことのできる速度であった。しかし、確実に誘導を行うためには更なる処理速度の向上が必要である。

7. 今後の課題

現段階ではベース車体が小さく実際に人物を誘導することは困難である。そのためベース車体の大型化が必要である。一般的盲導犬の体高は 60cm 程度であり、ベース車体についても同程度のサイズにする必要がある。

車体が小さく移動範囲が狭いため利用環境を室内と想定した。そのため本研究では GPS を利用しなかった。ベース車体を大型化し屋外での利用を考慮した場合においては GPS を用いた地図との連動が考えられる。地図上から目的地を取得しナビゲーションにより視覚障がい者に対する最適なルートで安全に誘導を行うことが今後の課題となる。

視覚障がい者の誘導法についても考慮が必要である。誘導法としては盲導犬と同じようにハーネスによる誘導、Bluetooth などの無線ヘッドセットを用いた音声による誘導が考えられる。

参考文献

- 1) 厚生労働省 平成18年身体障児・者実態調査結果
<http://www.mhlw.go.jp/toukei/saikin/hw/shintai/06/index.html>
- 2) 厚生労働省 補助犬の実働頭数
<http://www.mhlw.go.jp/topics/bukyoku/syakai/hoiyoken/html/b04.html>
- 3) 日本精工株式会社 ホームページ
<http://www.jp.nsk.com/>
- 4) 鈴木浩明, 藤波浩平, 大野央人, 水上直樹, 末田統, 井出将文: 「視覚障害者が日常空間で誘導用ブロック上を歩行する時の歩幅・歩行速度」人間工学 Vol.37, No4, pp192~196, (2001)
- 5) 大倉元宏, 村上琢磨, 清水学, 田内雅規: 「視覚障害者の歩行特性と駅プラットホームからの転落事故」人間工学, 31巻, pp1~8, (1995)
- 6) 田内雅規, 村上琢磨, 大倉元宏, 清水学: 「視覚障害者による鉄道単独利用の困難な実態」リハビリテーション研究, 第70号, pp33~37, (1992)
- 7) 公益財団法人 東日本盲導犬協会
<http://www.guide-dog.jp/guidedog/suitable-kind-of-dogs/>
- 8) 岩田直樹, 杉浦登, 高木基成, 原田明憲, 吉田研一:
「Android ADK プログラミング&電子工作バイブル」
ソシム株式会社, (2012年6月1日発行)
- 9) 田原淳一郎: 「Arduino で始める電子工作 -8bit マイコンを活用するオープンプロジェクト Arduino の世界-」株式会社カットシステム, (2010年4月20日発行)
- 10) 画像処理ソリューション
<http://imaging-solution.blog107.fc2.com/>

(受理 平成25年3月19日)