愛知工業大学研究報告
第 48 号　平成 25 年

Android 端末を用いた路上障害物認識装置

Device to Recognize the Obstacles for Vision Guide Using the Android Terminal

荒川 貴則†，江口 一彦††
Takanori ARAKAWA, Kazuhiko EGUCHI

Abstract: Severely 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 dogs. Therefore, available number of seeing-eye dog 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 ultrasonic sensor is discussed. An obstacle is recognized by introduction of image processing and an ultrasonic sensor. Obstacle avoidance by the voice guidance from an Android terminal is performed after that. As a result, it succeeded in a wearable evasion guidance.

1. はじめに

内閣府刊行の「平成 24 年版障害者白書」[1]によると，日本の身体障がい者の数は，視覚障がい者 31.5 万人，聴覚・言語障がい 36 万人，肢体不自由 181 万人，内部障がい 109.1 万人となっている．その中で一定以上の視覚障害を持つ視覚障がい者は道路交通法により外を歩く際には白杖又は盲導犬の使用が義務付けられている．しかし後天的に視覚障がいを負った人の中には白杖を持つことに抵抗感を抱く人も多く，盲導犬に関しては育成のための費用や時間によって制限を受けているため，十分な頭数が確保できない状態である．そのため近年では盲導犬の機能を工学的に再現する盲導犬ロボット等の補助装置の研究が進められている．白杖の役割としては障害物の認識と回避方向の認識，段差や階段の認識などがあり，盲導犬ではさらに目標物への誘導や道の端を歩くための誘導などの役割がある．

その中で本研究では障害物の認識及び回避について，カメラや方位センサ等の各種センサ機器，スピーカが搭載され，画像処理や音声誘導が行える Android 端末を用いて障害物認識装置の開発を検討した．

2. 装置について

本装置は前方道路上の障害物の認識には Android 端末のカメラを使い端末自身で画像処理を行うことによって障害物の認識を行う，そして前方の壁などの障害物を認識するために超音波センサを用いる．また装着者の体型などによってカメラの角度が変わることがあるため，角度を一定に保つためにサーボモータを用いている．超音波センサとサーボモータの制御には Arduino[2]を用いており，Bluetooth 通信モジュールを通し Bluetooth で Android 端末と通信を行っている．本研究で制作した装置の外観を図 2.1 と図 2.2 に示す．

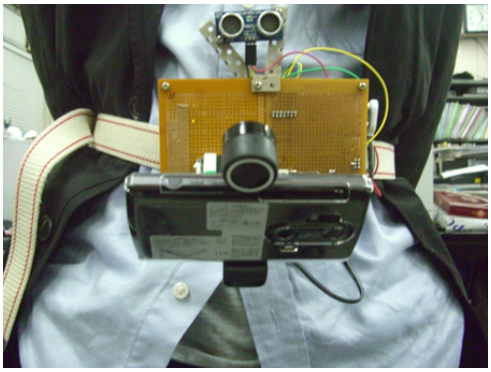

図 2.1　装置外観(正面)

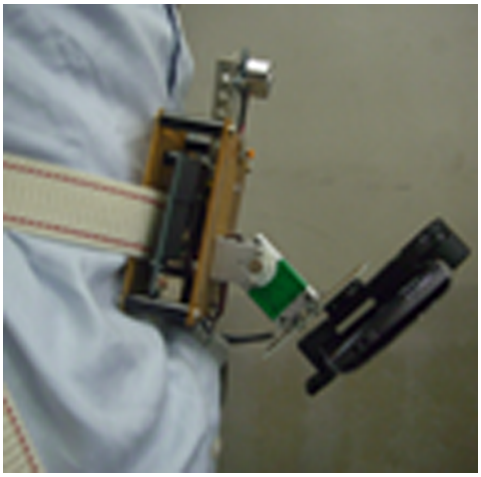

図 2.2　装置外観(側面)

†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)

愛知工業大学研究報告,第 48 号, 平成 25 年, Vol.48, Mar, 2013

3. 障害物認識及び回避動作の手順

3・1 画像処理

本研究では図 3.1 に示す手順で障害物の回避を行う.

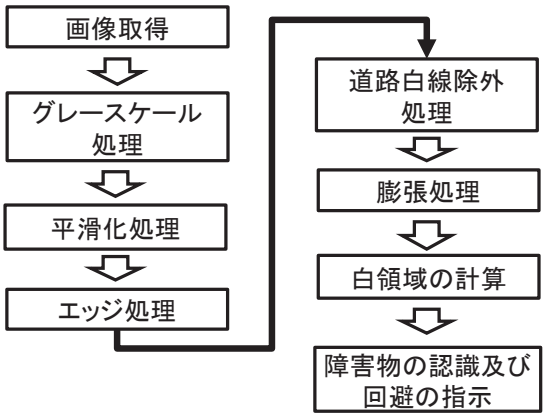

図 3.1 画像処理フローチャート

3・1・1 画像取得

研究では装着者の進行方向道路上の障害物を検出することを目的としているため, 歩行中常に進行方向を向いている腰に装置をつけることを想定している. そのため画像を取得するカメラの位置は地面から約 1m の位置, 角度を水平方向から約 45 度と設定する. また, 画像処理の負荷を抑えるために画像を取得後, 縦横のサイズを半分まで縮小し, 一連の処理後元のサイズに復元している.

3・1・2 グレースケール処理

初めに取得画像を R(Red), G(Green), B(Blue)の 3 原色を用いた加法混色である RGB カラーモデルから, 白から黒までの明暗で表されるグレースケール画像に変換する. これは画像処理を行う際に色を単純化し処理を行いやすくするために行う. 変換の式を式(3.1)に示す.

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (3.1)$$

3・1・3 平滑化処理

取得した画像はそのままではエッジ処理などを行った際にノイズが残り誤検出の原因となる, そのためメディアンフィルタを用いフィルタリングを行う.

メディアンフィルタは注目画素とその近傍 8 点の画素の輝度値を比較し, 小さい順に並べた時の中央の値を出力画素とする処理方法である.

例として, 図 3.2(a)のように注目画素(値 129)の周囲の輝度値が与えられている場合, これを(33, 37, 39, 54, 57, 78, 76, 88, 129)の順に並べ, この中央の値 57 を注目画素である 129 と置き換え, 図 3.2(b)のように変化させる.

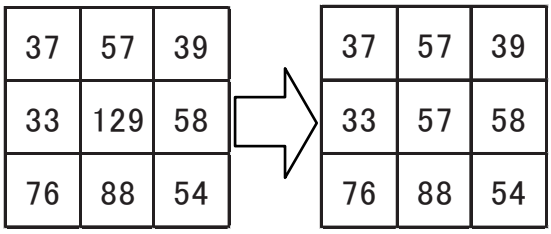

図 3.2 メディアンフィルタの処理

3・1・4 エッジ処理

エッジ処理は平滑化された画像を元に画像の輪郭を抽出する為に行う. エッジ処理には Sobel フィルタ, Laplace フィルタ, Canny フィルタ等の種類がある, 本研究では Canny フィルタを用いて行い, これによって検出された輪郭を障害物として認識する.

Canny フィルタは Canny アルゴリズム[3]を用いて処理される. Canny アルゴリズムの手順は以下のとおりである.

まず, Canny アルゴリズム内でガウシアンフィルタを用いて平滑化を行う. ガウシアンフィルタは注目画素とその周辺の輝度値にガウス分布の関数を用いて計算した係数を掛け合わせ輝度値を求めることによって画像を平滑化する処理である. ガウス分布の関数は式(3.2)のとおりである. また例となる図を図 3.3 に示す

$$f(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (3.2)$$

$$\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$$

図 3.3 ガウシアンフィルタ係数

次に Sobel フィルタを用いてのエッジ強度と勾配方向の計算を行う. Sobel フィルタは 1 次の微分フィルタであり, 中央の注目画像の周囲に重み付けを行い, 割り当てられた係数と画素の乗算を求める, この後それぞれの値を式(3.3)に当てはめ変化の大きい箇所を強調する. そしてこの処理を水平方向と垂直方向で行いエッジを検出する.

図 3.4 にそれぞれの係数を示す.

$$g = \sqrt{{g_{HS}}^2 + {g_{VS}}^2} \quad (3.3)$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

水平方向g_{HS} 垂直方向g_{VS}

図 3.4 水平方向・垂直方向の Sobel フィルタ係数

検出されたエッジは平滑化された画像のものなのでエッジを細くするための細線化処理を行う．この処理はエッジと鉛直方向の隣接画素二つを比較し最大でなければ 0 とする．

最後に閾値を二つ定めその間でエッジに結合しているものはエッジと認識するように処理を行う．

3・1・5 白線除外処理

道路上を歩く際に白線は頻繁に取得画像に入るがこの白線をそのままにしておくとエッジ処理の際に検出されたものが残り，障害物として判断されてしまう．これを避けるためにエッジ処理まで終わった画像と図 3.5 の手順を用いて作成したネガポジ画像を論理積することによって画像内の道路白線の除外処理を行う．

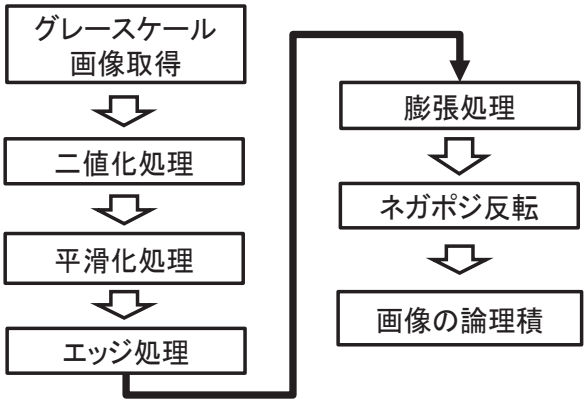

図 3.5　道路白線除外処理

この処理では取得画像の二値化処理を行い白線の検出を行なっている．二値化処理は元画像の輝度値が閾値より大きい画素を黒に，それ以下の画素を白にする処理を行なっている．この閾値は白色のものが検出できる輝度値の大きさを設定した．その後にノイズを消すためにメディアンフィルタによる平滑化を行う．図 3.6 にグレースケール化画像を図 3.7 に二値化後平滑化処理を行った画像を示す．

図 3.6　グレースケール画像

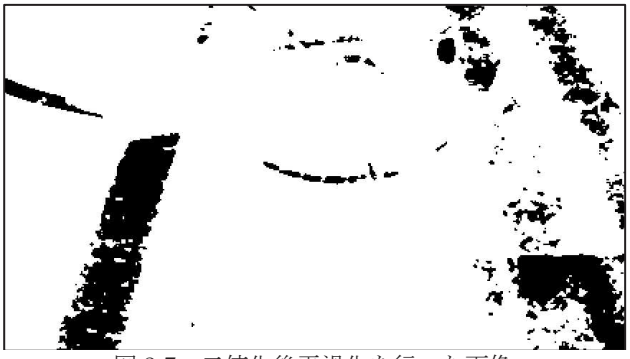

図 3.7　二値化後平滑化を行った画像

白線のみを検出した二値化画像から，エッジ処理を行い白線の輪郭を検出する．この二値化処理で得られた白線の輪郭はこのままでは元の画像の白線の輪郭と完全には重ならないため膨張処理を行い白線の輪郭を太くする．膨張処理の説明は§3・1・5 にて行う．エッジ処理後膨張処理を行った画像を図 3.8 に示す．

図 3.8　膨張処理画像

この太くした輪郭にネガポジ反転を行う．この処理は色の階調を反転させるもので今回の処理では白と黒の色を反転させている．これを図 3.9 に示す．

図 3.9　ネガポジ反転

最後にネガポジ反転を行った白線の輪郭の画像である図 3.9 と白線除外処理までに生成した元画像のエッジ画像である図 3.10 を論理積することによって白線部の輪郭を除外することができる．除外後の画像を図 3.11 に示す．

愛知工業大学研究報告,第 48 号, 平成 25 年, Vol.48, Mar, 2013

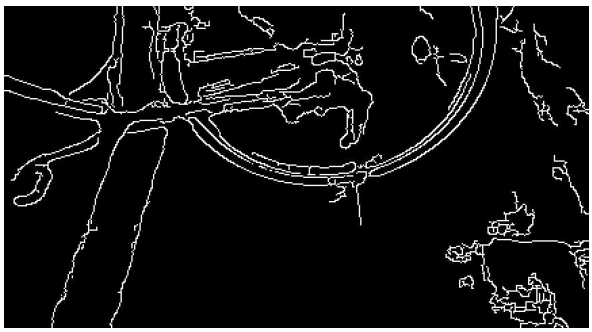

図 3.10　元画像のエッジ処理画像

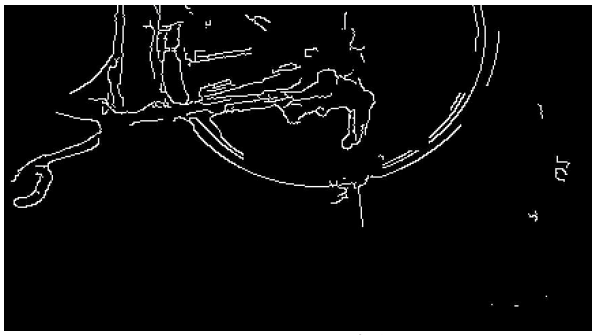

図 3.11　論理積後

3・1・6　膨張処理

膨張(Dilation)処理とは主に二値化された白黒画像に対して行う処理である，注目画素の周辺に 1 画素でも白い画素があれば白い画素に置き換える処理である．また，逆に周辺に 1 画素でも黒い画素があれば黒い画素に置き換える処理で収縮(Erosion)処理というものがある．この二つの処理を組み合わせることによって色々なノイズの除去に役立つ．本研究においては膨張のサイズを 3×3 とし，実行回数を 4 回とした．この処理はエッジ処理において検出した輪郭を拡大し，障害物の面積の大きさに近づけるために用いている．

3・1・7　白領域の処理

白領域の計算は，膨張処理まで行った画像に対して白色のピクセル数を計算する処理である．計算の際は図 3.12 に示す領域の白色ピクセル数を各領域それぞれで計算している．この領域は，人が歩く際に肩幅よりも大きく左右に足を伸ばして歩くことはまず無いことから，中央列の下端の幅が男性の肩幅の平均[4]である 40cm よりも大きい 50cm 以上となるように左，中央，右の画面比率を 2:3:2 と設定している．これは後の障害物認識及び回避の際に使用する．

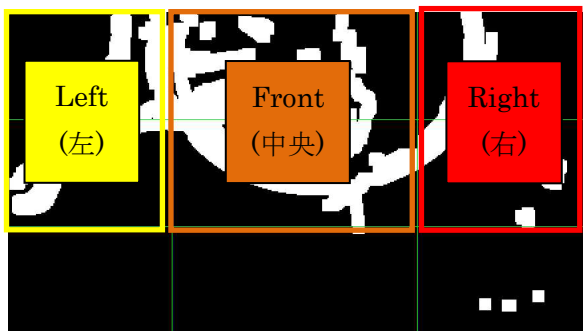

図 3.12　各領域

3・1・8　障害物の認識及び回避の指示

障害物の認識及び回避は，各領域の白ピクセル数を用いて行う. 図 4.22 で示した様な中央列を装着者の進行方向であり歩行部分として考える．そのため中央の領域に設定した閾値以上の白色ピクセル数が検出された場合障害物と判断し静止するように Android 端末から音声で指示を出す．その後，図に示す左右の領域の白色ピクセル数を比較し回避の指示を出す．音声には音声合成フリーソフト Softolk[5]を用いる．

障害物の認識に使われる閾値は 3000 としている，これは中央領域の総ピクセル数が 29280 であるため約 10%の値を基準とした．また，それぞれの画像内に表示されているピクセル数を図の左に表記してある．

図 3.13 の場合, 右のピクセル数の方が左のピクセル数より少ないので右を向くように指示を出す.

図 3.13　ピクセル数の比較

3・2　超音波センサでの検出

超音波センサを用いて壁の検出を行う．これは本研究の画像処理では単一色の壁に対してエッジを検出することができない事から前方の壁を障害物であると認識できない，そのために超音波センサを用い前方の壁の認識を行う. 画像処理では認識できない壁の例を図 3.14 に示す. 壁の認識では超音波センサの出力信号が cm 単位で Arduino に入力されている．この値から壁との距離が 140cm より小さい値になった場合に障害物があると認識する．

また，超音波センサは特性上，角度に弱く本研究に用いた超音波センサでは対象物との角度が 20 度以上では距離を測定することができない[6].

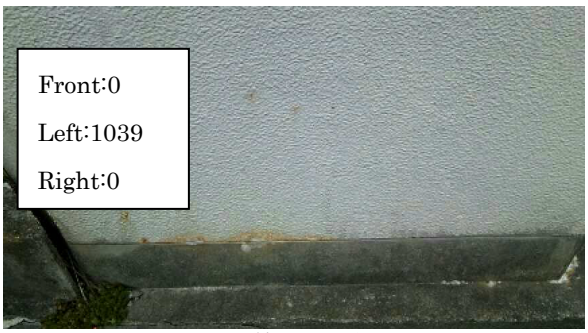

図 3.14　検出できない壁

3・3　検出時の一連の動作

本装置が障害物を検出した場合の処理の流れを図 3.15 に示す.

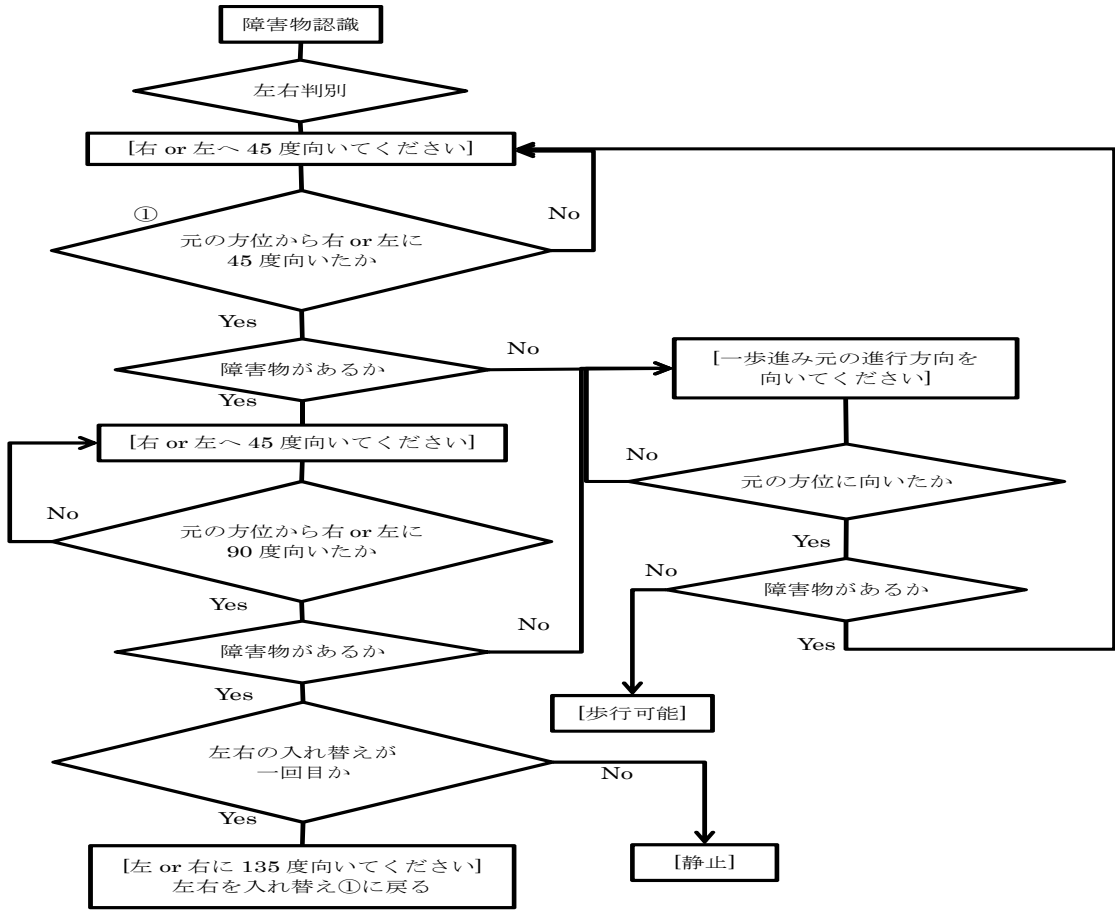

図 3.15 回避動作フローチャート

前方の障害物を認識後, 左右の領域の白色ピクセル数を比較し, 値の低い方を障害物が無い可能性の高い方向と考え，そちらに体ごと向くように音声で指示を行う.

Android 端末に搭載されている方位センサを用いて画像処理を 45 度ずつ行い障害物があるかどうかを確かめる. この際, 画像処理では検出されなくとも超音波センサで検出された場合は障害物が存在すると考える.

障害物が検出されなければ一歩進み元の方向に体を向けるように指示を出す. 元の方向に戻り障害物がなければ音声の案内を終了し, 前方に進めることが分かるようになっている，逆に障害物が存在した場合，前回と同じ方向に体の向きを変えて行くように設定している.

元の方位から 90 度まで体を向けても障害物が存在した場合，逆方向に向いて行くように指示を出す.
図 3.15 の流れが左右両方で行われた場合，袋小路に入ったと考え，静止するように指示をする.

4. 実験結果

4・1 画像処理による障害物認識

障害物の認識のみの結果を以下に示す.

4・1・1 対象:障害物

道路上の障害物を撮影し画像処理を行う. 今回は道路上で障害物となりやすい自転車，植木鉢を対象とした.
図 4.1 では，障害物の検出に成功した．しかし，図 4.2 の障害物ではピクセル数は閾値よりも遥かに低くなった, これは白線除去処理により輪郭の一部が消されたためである.

図 4.3 は検証のためそれぞれ白い植木鉢のエッジ処理を行った画像である, この画像に白線除去処理を行った画像が図 4.4 である. このことから白い植木鉢の画像が白線除去処理によって輪郭が消えてしまい検出が行えなかったことがわかる．自転車の画像については§3・1・5 白線除外処理の際に提示しているため省略する.

図 4.1 自転車

愛知工業大学研究報告,第 48 号, 平成 25 年, Vol.48, Mar, 2013

図 4.2　白色の植木鉢

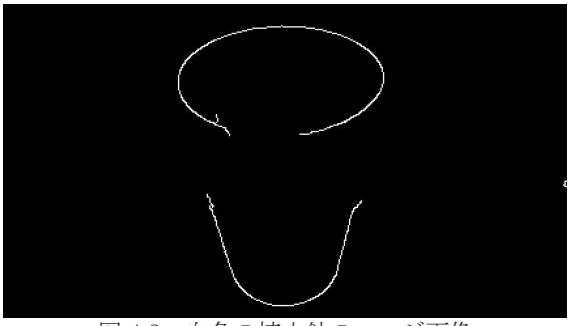

図 4.3　白色の植木鉢のエッジ画像

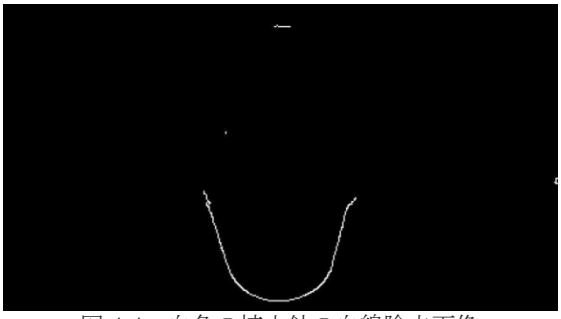

図 4.4　白色の植木鉢の白線除去画像

4・1・2　対象:路面

障害物のない路面そのものの画像処理を行う.

路面の検出では障害物となる輪郭が検出されないことが望ましい. 図 4.5 では問題なく処理が行われている. しかし図 4.6 では輪郭が検出され, 閾値よりも高い値になっている. これはインターロッキングブロックの溝の輪郭を検出してしまったためである.

検証のため各図のエッジ画像を以下に示す. 図 4.7 ではエッジを検出せず障害物が無いと判別できている. しかし図 4.8 ではインターロッキングブロックの溝を検出していまい障害物有りと誤検出していることがわかる.

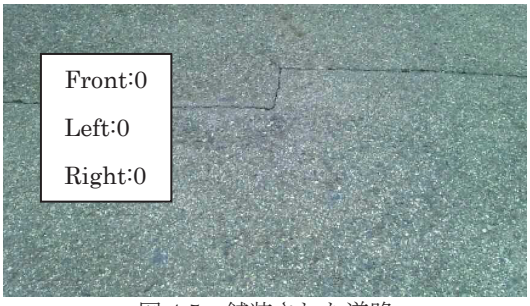

図 4.5　舗装された道路

図 4.6　インターロッキングブロックの歩道

図 4.7　舗装された道路のエッジ画像

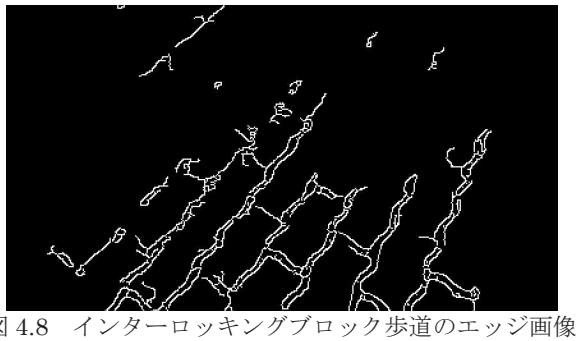

図 4.8　インターロッキングブロック歩道のエッジ画像

4・2　障害物回避動作

画像処理による障害物認識及び音声による回避動作の誘導の結果を以下に示す.

それぞれ図 3.15 のフローチャートにおいて,[歩行可能],[静止]となる状況下を作成し実験を行った.　また, 誤動作となった結果についても示す.

4・2・1　状況[歩行可能]:前方に障害物

状況を略式化した図を図 4.9 に示す.

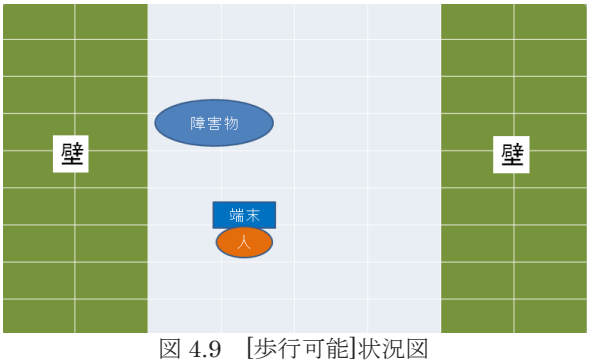

図 4.9　[歩行可能]状況図

状況[歩行可能]では障害物の数 m 手前から歩行して行き「障害物です，静止してください」という音声による指示に従って止まった．その時の画像が図 4.10 で，障害物との距離は約 1m である．この時，表示されている左のピクセル数は 4931，右のピクセル数は 2798 であるので右 45 度を向くように音声で「右に 45 度向いてください」と指示が出された．

図 4.10　障害物認識(正面)

指示に従い右に向き 45 度になった際にもう一度画像処理を行うその時の画像が図 4.11 である．この時前方のピクセル数が 0 となっているため障害物は無いと判断され「一歩進み左 45 度を向いてください」と音声で指示を出された．

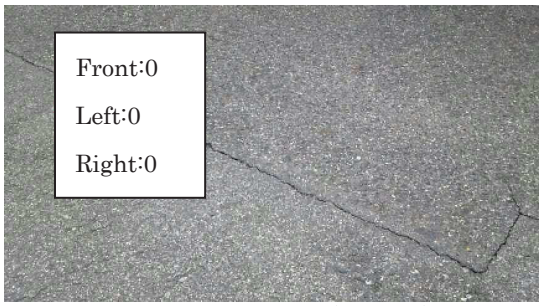

図 4.11　障害物認識(右 45 度)

一歩進み元の進行方向に向き直した後再び画像処理を行う，その時の画像が図 4.12 である．前方のピクセル数が 0 であるため障害物が無いと判断し，音声での誘導が止まった．

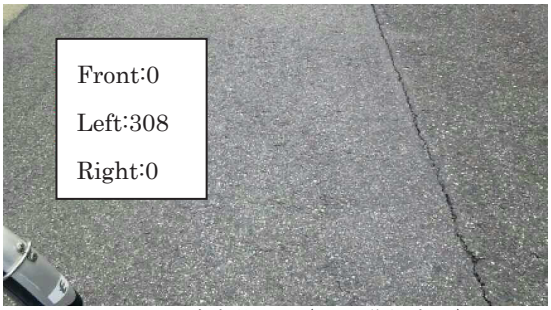

図 4.12　障害物認識(元の進行方向)

4・2・2　状況[静止]:前方及び左右に障害物

状況を略式化した図を図 4.13 に示す．

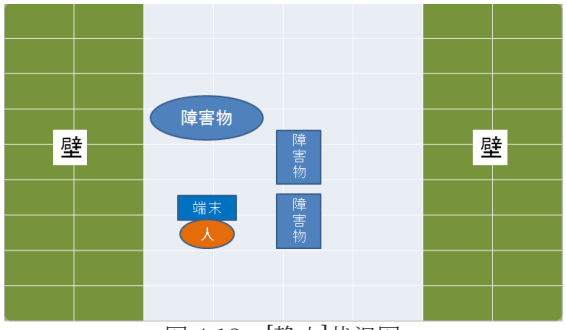

図 4.13　[静止]状況図

状況[静止]においても前方障害物から数 m 離れたところから歩いて行き音声による警告が出たところで静止した，その時の画像が図 4.14 である．障害物との距離は約 80cm であり，状況[歩行可能]よりも短くなった，これは状況[静止]の中央領域がサドルの部分であり障害物検出に用いる輪郭が少ない為であると考えられる．

左右のピクセル数の比較においては左:8019，右:10906 となったため，「左 45 度を向いてください」と指示が出された．

図 4.14　障害物認識(正面)

指示に従い左を向いて行き左に 45 度向いた時に画像処理を行い障害物の有無を調べる，状況[静止]では図 4.15 の様に前方に障害物があったのでさらに左に 45 度向くように音声で指示が出された．

図 4.15　障害物認識(左 45 度)

左 90 度まで向いた時に再度画像処理を行い障害物の有無を調べる，状況[静止]では図 4.16 の様になり前方に障害物があるので今度は元の進行方向から右に 45 度向いた場所を調べるために「右に 135 度を向いてください」という指示が出された．

愛知工業大学研究報告,第 48 号, 平成 25 年, Vol.48, Mar, 2013

図 4.16　障害物認識(左 90 度)

その後，元の進行方向から 45 度右に向いた時に画像処理を行い図 4.17 の通り前方に障害物がるため，さらに右に 45 度に向くように指示が出された．
そして元の進行方向から 90 度向き画像処理を行うと図 4.18 となった．既に左 45 度，90 度の検出が終わっているため現在の位置が行き止まりであると考えられるので「静止してください」と言う指示が出され，その場で静止した．

図 4.17　障害物認識(右 45 度)

図 4.18　障害物認識(右 90 度)

4・2・3　状況[誤動作]:前方に障害物，左に単色の壁

この状況では図 4.19 の様な前方の障害物を回避するため左に 45 度向いた，図 4.20 の画像では壁が有り進むことができないが，処理では一歩進み右 45 度を向くように指示が出された．これは単色の壁であるため輪郭が検出できず中央領域の検出ピクセル数が 0 となった事と超音波センサは 45 度の角度では障害物を検出することができない事ために誤動作をした．

図 4.19　障害物認識(正面)

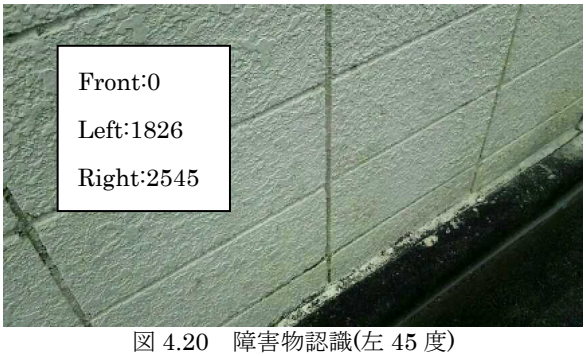

図 4.20　障害物認識(左 45 度)

4・3　画像処理の FPS

本研究で用いた画像処理の FPS は表 4.1 の様になった．FPS とは 1 秒あたりに処理されるフレーム数を表す単位である．本研究の画像処理おいて誤検出防止のために 5 フレーム連続して中央ピクセル数が閾値を超えた時のみ障害物と認識する．図 4.21 にグラフで示す．

表 4.1　障害物認識処理においての FPS

	FPS	
経過時間(秒)	障害物無し	障害物有り
0	5.8	5.45
5	6.12	5.32
10	6.18	5.45
15	6.37	5.35
20	6.25	4.91
25	6.01	5.36
30	6.02	5.49
35	6.01	5.52
40	6.12	5.45
45	5.93	5.37
50	5.94	5.4
55	6.31	5.45
60	6.17	5.39
平均	6.09	5.38

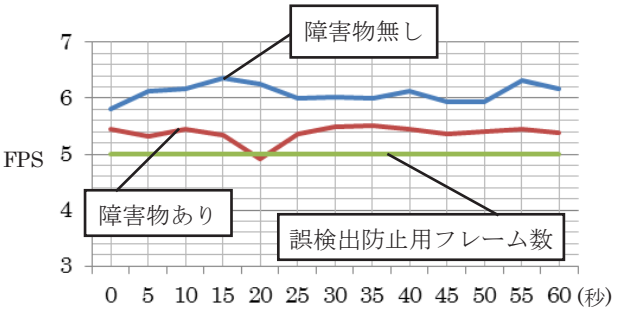

図 4.21　障害物認識処理においての FPS グラフ

5. 結論

本研究において成功したことは以下のとおりである.

- Android 端末を用いた道路上障害物の認識
- 音声を使った障害物回避誘導
- 超音波センサを用いた壁の認識

このことから得られた結論を以下に述べる.

Android 端末による障害物の認識及び回避において道路上の障害物の認識を行い音声によって障害物を回避する一連の動作には一部の状況を除いて成功した. この際, FPS は障害物未検出時で 6.09, 検出時には 5.38 となった, 誤検出防止のため 5 フレームを用いて障害物の判定を行うため十分ではないが, 障害物認識と同時に静止を行い, 方向転換時にのみ画像処理を行うため, 回避動作において問題はないと考える.

しかし画像処理においてタイルや土がむき出しになっている道では誤認識も多い. また, 現在の白線除去処理では白色の障害物も除去してしまい障害物の認識に失敗してしまうため, 画像処理及び障害物検知の工夫が必要だと考えられる.

6. 今後の課題

現在の装置では目的地までの道は装着者本人が知っていなければならないが, しかし視覚障がい者は地図を見ることが難しいため, 初めて向かう目的地までの経路を自身で決定するのは難しい. そのため Android 端末に搭載されている方位センサや GPS と google マップ等の地図アプリを用いて視覚障がい者を誘導するナビ機能の搭載が考えられる.

また, 歩行者が歩行する際には道路の端となる道路境界と平行して歩くことが多い, そのため道路境界を認識することによって安全に歩行し, 障害物回避の際には実験結果 4・2・3[誤動作]を防ぐ形で回避方向を決定することができると考えられる. この道路境界認識の研究は本研究室 2011 年度修了の上條[7]が研究しており, 本研究にも応用できると考えられる.

同時に装置の小型化も必要であると考えられる. これは, 現在はカメラとなっている Android 端末が飛び出している形になっているため, 装着者の位置よりも取得画像の位置が若干であるがずれてしまっていることの改善や装着者の負担を軽減に繋がるからである.

参考文献

[1] 内閣府「平成 24 年度版　障害者白書」
http://www8.cao.go.jp/shougai/whitepaper/h24hakusho/zenbun/pdf/index.html
[2] Arduino　HomePage
http://www.arduino.cc/
[3] しぼりたてブログ
http://d8yd.blog105.fc2.com/blog-entry-84.html
[4] 身体別　男女肩幅の平均 :「男性の体のサイズ」と「女性の体のサイズ」の身長別の平均値がわかるサイト
http://homepage3.nifty.com/orangejuice/body1.html
[5] SofTalk
http://www35.atwiki.jp/softalk/
[6] 荒海宇宙/荒川貴則/井尻健嗣「視覚障がい者の歩行補助機器の開発」愛知工業大学　卒業論文(2010)
[7] 上條善治「 Android 端末を用いた視覚障がい者歩行補助システムの検討」愛知工業大学　修士論文(2011)
[8] インターネットコム
http://japan.internet.com/allnet/20120822/3.html
[9] 寝屋川市ホームページ「駅周辺道路のバリアフリーカルテ」
http://www.city.neyagawa.osaka.jp/var/rev0/0004/9118/03.pdf
[10] 野田宏治/松本幸正/荻野弘/栗本譲「視覚障害者のための歩行案内システムの評価に関する研究」土木学会論文集　No.548/IV-33, 45-54, 1996.10
[11] 溝口早苗「Smartphone World Volume.1 」CQ 出版社(2011 年 5 月 1 日発行)
[12] 小澤拓治「Smartphone World Volume.2」 CQ 出版社(2011 年 9 月 10 日発行)
[13] 公益財団法人　日本盲導犬協会
http://www.moudouken.net/knowledge/article.php?id=33
[14] OpenCV.jp　OpenCV 逆引きリファレンス
http://opencv.jp/cookbook/
[15] 吉井博史「作例で覚える Android プログラミング入門」株式会社　ソーテック社(2011 年 9 月 20 日発行)
[16] 大木敦夫　「色・エッジ情報を用いた汎用性のある障害物回避」　早稲田大学大学院　修士論文(2004)
[17] 光学堂ロービジョンルーム
http://www.kougakudo.jp/index.html
[18] 厚生労働省　ほじょ犬情報
http://www.mhlw.go.jp/topics/bukyoku/syakai/hojyoken/html/b04.html
[19] 日本精工ホームページ
http://www.jp.nsk.com/

（受理　平成 25 年 3 月 19 日）