

屋外環境における自律走行車両の開発

飯倉, 裕樹 / IIKURA, Hiroki

(発行年 / Year)

2005-03-24

(学位授与年月日 / Date of Granted)

2005-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

2004 年度 修士論文

屋外環境における自律走行車両の開発

指導教授 小林一行 助教授

法政大学大学院工学研究科

システム工学専攻修士課程

03R6104

イクラ ヒロキ

飯倉 裕樹

DEVELOPMENT OF AUTONOMOUS GROUND VEHICLE
FOR AUTONOMOUS CHALLENGE COMPETITION

Hiroki Iikura

Abstract

To develop autonomous navigation technology, the Intelligent Ground Vehicle Competition (IGVC) is held every year in United State of America since 1993. Autonomous challenge competition (ACC) is one of the competitions of the IGVC. In the competition, the vehicle must autonomously travel between white lanes avoiding randomly placed obstacles such as construction drums which are often used at actual freeway. And also, in the travel course, several simulated painted potholes which should have to avoid, are placed. In order to travel smoothly, the intelligent vehicle automatically recognizes white lanes, obstacles and potholes, individually.

The developed recognition engine based on the captured image from omni-directional camera and the range profile from laser rangefinder are fused to achieve autonomous navigation task. The image recognition is one of key technology for our developed recognition engine, such as a template matching and Hough transforms are used as real-time image recognition. Based on the developed recognition engine, the vehicle demonstrates successfully recognized and controlled.

The validity of the proposed recognition engine is verified at 2004 IGVC competition at autonomous challenge.

Key Words: IGVC, ACC, image recognition, template matching, Hough transform, vehicle control

目次

目次	3
1.はじめに	5
1.1 概要	5
1.2 Autonomous challenge Competition (ACC)	5
2. 車両システムと問題の記述	7
2.1 仮定	7
2.2 問題	7
2.3 システムの記述	7
2.4 オムニカメラについて	9
3.画像処理による障害物検出	11
3.1 パイロン検出	11
3.1.1 R成分画像によるパイロン検出	11
3.1.2 U成分画像によるパイロン検出	12
3.1.3 比較実験	12
3.1.4 実験結果	12
3.1.5 結論	13
3.2 ポットホール検出	14
3.2.1 円形度と面積を用いた手法	14
3.2.2 円形検出 Hough 変換	15
3.2.3 テンプレートマッチングによる検出	15
3.2.4 実験	16
3.2.5 考察	18
3.2.6 結論	18
3.3 バケツ検出	19
4.白線検出	20
4.1 従来手法	20
4.2 形状の検出	21
4.3 ノイズ除去	22
4.3.1 濃淡値の分布によるノイズ除去	22
4.3.2 時系列画像によるノイズ除去	23
4.4 実験	23
4.5 結果	24
4.6 考察	24
4.7Hough 変換による直線検出	25
4.7.1 提案する手法	25
4.7.2 レーン検出実験	26
4.7.3 実験結果	26
4.7.4 考察	26

5. 走行制御.....	27
5.1 レーザレンジファインダによるパイロンの検出.....	27
5.1.1 探索エリア.....	27
5.1.2 トラップの回避.....	27
5.1.3 坂道により地面を検出した場合の対処法.....	28
5.2 目標走行ラインの選択.....	28
5.3 目標ラインの追従(微分先行型PD制御).....	29
6. 走行実験.....	31
6.1 実験環境.....	31
6.2 結果.....	32
6.3 考察.....	32
7. まとめ.....	33
参考文献.....	34
付録1 テンプレートマッチング.....	35
付録2 ハードウェア仕様.....	36
謝辞.....	38

1.はじめに

1.1 概要

近年、様々な分野で自動化が試みられている中、ロボットの自律走行は大きな課題とされている。自律走行の研究として、路面上に磁気ネイルや誘導ケーブルを埋設し、センサによってそれらの位置を検出する手法が報告されているが¹⁾、この装置を実現するためのインフラの設備にかかる費用と時間はきわめて大きい。一方設備のいない画像処理技術を用いた方法は単に走路形状を認識するだけでなく、同時に周囲の環境を認識してしまうため、高度な環境認識アルゴリズムが要求され、まだ実用的な段階になっていない。

渡辺・小林研究室では屋外自律走行車両の研究の一環として Intelligent Ground Vehicle Competition(IGVC)に参加している。この大会は Association For Unmanned Vehicle Systems International(AUVSI)が自律走行技術の向上のために 1993 年から毎年開催しているものである。その中の1つの競技として Autonomous Challenge Competition (ACC)というものがある。この競技は車両を自律走行させ、屋外の芝生に描かれた二本の白線の間を障害物を避けながらいかに早くゴールまでたどり着けるかというものである。本論文では大会に使用する白線検出と走行制御について報告する。

1.2 Autonomous challenge Competition (ACC)

この競技は芝生の上に描かれた 2 本の白線の間を障害物を避けながらゴールを目指す競技である。コースにおかれている障害物はパイロン・ポットホール・バケツの 3 種類ある。パイロンは半径 0.25[m]、高さ 1.2[m]の赤白にペイントされた円柱プラスチックコーンで、倒すと失格、触れると減点となる。ポットホールは地面に描かれた半径 0.3[m]の白色の円であり踏むと減点となる。バケツは半径 0.15[m]、高さ 0.6[m]であり、倒すと失格、触れると減点となる。白線の太さは約 0.15[m]であり、2本の白線間隔は約 2[m] ~ 6[m]である。

ACC は日差しが強い屋外で開催されるので画像処理が非常に困難になっている。またパイロン、ポットホール、バケツの白色は白線検出の妨げとなるため、これらの検出及び除去をする必要がある。コースの白線は破線部、白線間隔の変化、急カーブ、などが存在する。このような環境に対してロバストな処理が求められる。

Photo.1 にコース環境を示す。

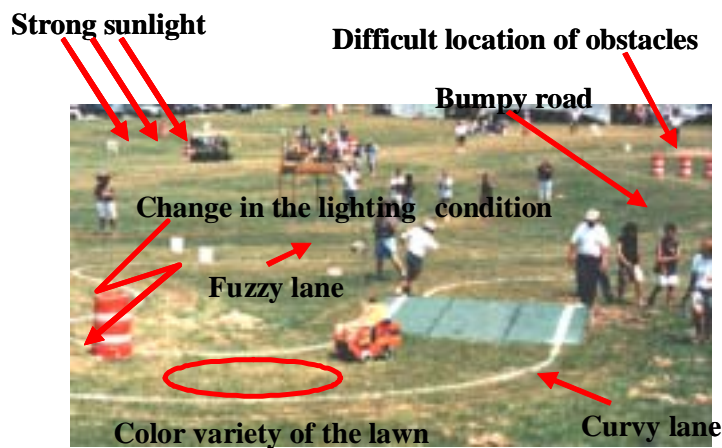


Photo.1 Course environment

Table 1 に Autonomous Challenge Competition の環境設定を, Table 2 に走行時のルールを示す.

Table 1 Regulations of Autonomous Challenge Competition

路面	芝生
白線の太さ	約 0.15m
白線間隔	約 2 ~ 6m
パイロン	赤と白の縞々模様, 高さ 1.2m, 半径 0.25m
バケツ	白色, 高さ 0.6m, 半径 0.15m
ポットホール	白色, 半径 約 0.3m
坂	傾斜 15 度以内

Table 2 Rules of Autonomous Challenge Competition

白線	踏むと減点. 車両が完全に白線外に出ると失格.
パイロン, バケツ	ぶつかって動かすと失格, かすると減点 5point
ポットホール	踏むと減点 5point

全ての自律走行車は安全確保のため 8[km/h]以内の速度で走行し, 非常停止装置を装備しなくてはならない. また, 競技中に車両を人為的に遠隔操作することはできない. 全てのコンピュータ電源, センサ, 制御装置は車両上になくてはならない. なお, 白線の幅と一般的な障害物の色と大きさは自律走行を目的とするためあらかじめ決められているが, コースレイアウトや障害物の位置などは当日まで知らされず, プログラム中に組み込むことはできない. また, 自律走行車の形態にも以下の条件がある.

- ・ 全長: 最小長 90[cm], 最大長 270[cm]
- ・ 車幅: 150[cm]以内
- ・ 車高: 180[cm]以内 (緊急停止用無線を含まない)
- ・ 推進規定: 車両はタイヤ, キャタピラなど直接地面に接地して推進しなくてはならない.
- ・ ペイロード: 自律走行車は, 10[kg]の重りが運べるだけのパワーを必要とする. ペイロードサイズは 46[cm]×20[cm]×20[cm]である.

ACCの勝利条件を以下に示す.

(a)コースを完走するチームが1チームもいなかった場合

走行距離の最も長いチームが勝利. ただし減点項目があると, 1ポイントに 1[foot]の減点となる.

(b)コースを完走するチームが1チームのみの場合

減点項目の有無に関わらず, 完走したチームが勝利.

(c)コースを完走するチームが2チーム以上いた場合

コースを一周するのにかかった時間が短いチームが勝利. ただし減点項目があると1ポイントにつき 1秒の減点となる.

ACCでは各チームともに1ヒートにつき3回走行する権利があり, 3ヒート設けられているので1日に合計9回走行することが可能である. 第1ヒート1回目の走行はくじ引きによって走行順序が決められ, すべてのチームの走行が終了すると, 後は順番を関係なく準備がととのったチームから走行できる. ただし, 1ヒートにつき3回の走行と定められているので, 3回走行してしまうと次のヒートまで走行することはできない. また第1ヒートに1回しか走行しなかった場合でも, 次のヒートに走行権利を繰り越すことはできない.

以上を踏まえた上で, どのように画像処理, 車両制御を行なうかが問題となってくる.

2. 車両システムと問題の記述

2.1 仮定

IGVC の競技ルールに従い以下の仮定を設ける .

- (A1)地面は局所的に水平な平面である .
- (A2)パイロンは赤白の縞模様をしている .
- (A3)ポットホールは半径約 0.3[m]の地面に描かれた白色の円である .
- (A4)バケツは白色であり、芝生に描かれた白線よりも強い白色成分を持つ .
- (A5)白線の太さは約 0.15[m]である .
- (A6)2本の白線間隔は約 2~6[m]である .
- (A7)白線は近似的に平行に存在する .
- (A8)移動障害物は存在しない .

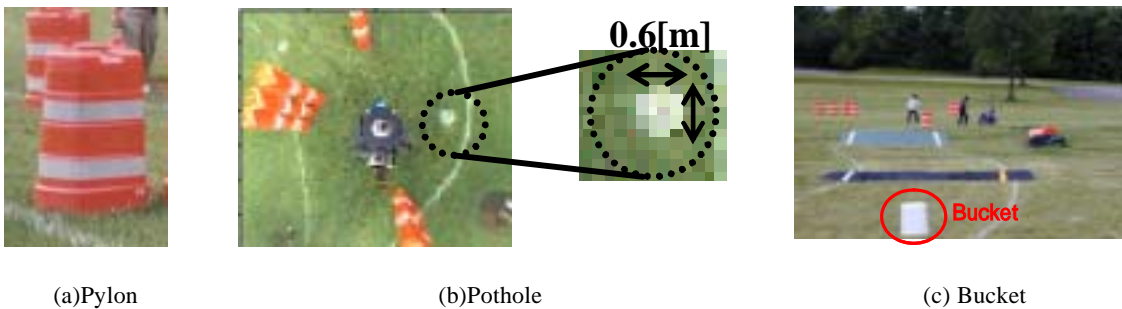


Fig.1 Obstacles that disturb vehicle's way

2.2 問題

ACC において安定した自律走行を実現するためには、以下の 3 種類の問題を解決することが必要である .

(P1)どのようにして画像中からパイロン・ポットホール・バケツを検出するか .

パイロン・ポットホール・バケツは白色成分を含んでいるので、白線検出の妨げとならないように画像中から検出・除去する必要がある . また、レンジセンサではポットホールを検出することができず、バケツを検出することも困難であるから、画像を使ってそれぞれ位置を検出し、回避をしなければならない .

(P2)どのように安定した白線検出を実現するか .

屋外環境では光量変化が激しいので 2 値化やエッジ検出など画像処理が非常に困難である . そこで光量変化に強い白線検出法を実現しなければならない .

(P3)障害物を回避しながらコース内を走行するためには、どのような走行制御を行えばよいか .

コース中に存在する様々な障害物を回避しながらゴールを目指すためには、最適な経路を選択しなければならない . また、最適な経路を追従するためにはロバストな車両制御が必要となる .

2.3 システムの記述

本研究では電動車椅子にオムニディレクショナルカメラ(以下、オムニカメラ)とレーザレンジファインダを搭載した車両を使用する . Photo.2 にその概観を、Fig.2 に車両を制御するための信号の流れを示す .

競技では 2 本の白線検出・追従、障害物の衝突回避が重要な課題であるため、分解能は粗くても環境全体の大局的情報がリアルタイムで獲得できる必要がある . そこで 1 台のカメラで車両周辺 360 度認識可能なオムニカメラを採用した . また屋外環境では光量変化が激しく画像処理が難しいことから、パイロン検出用に安定した角度・距離検出が可能なレーザレンジファインダを採用した .

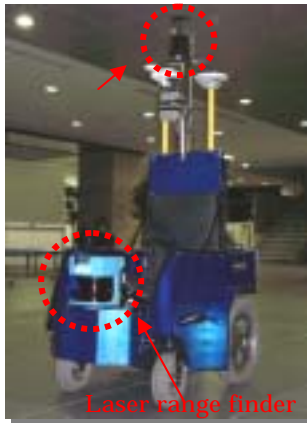


Photo 2 Vehicle appearance

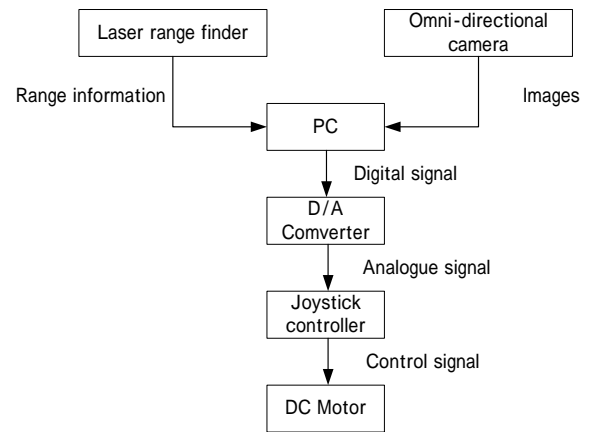


Fig.2 System description

2.2 の問題を解決するための自律走行のアルゴリズムを Fig.3 に示す。

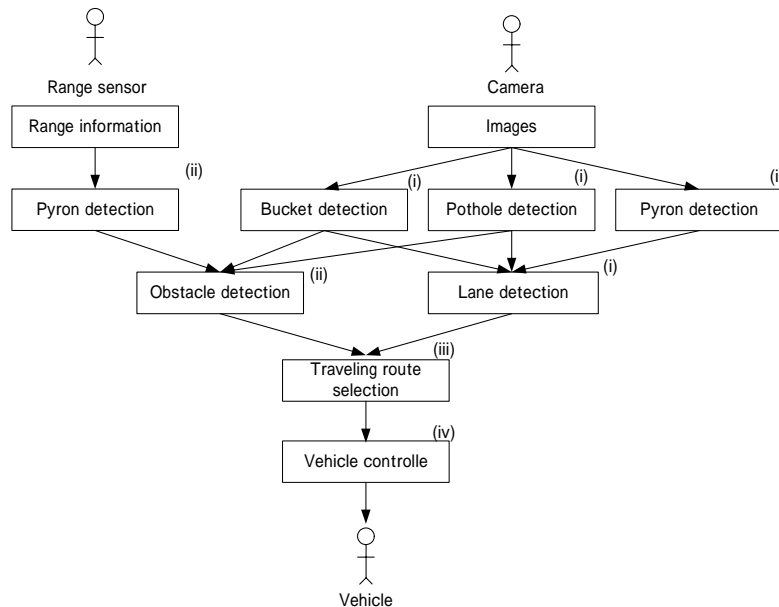


Fig.3 Algorithm for Autonomous traveling

(i)白線のみを検出するために、画像中から白色成分を持つパイロン・ポットホール・バケツを検出する。これらを画像中から除去し、画像から白線を検出する。

(ii)レンジセンサから得られた距離データの中からパイロンを検出。先の画像処理から得られたバケツ、ポットホールの位置と合成し、回避すべき障害物を検出する。

(iii)白線情報と回避すべき障害物のデータを合成し、回避経路を算出する。

(iv)走行経路が算出されたら制御則に入力し、車両のモータを動かす信号を出力する。

本論文の構成は3章では(P1)を解決するために、パイロン・ポットホール・バケツなど画像を使った障害物の検出について述べる。4章では(P2)を解決するための、画像処理による白線検出方法について、5章では(P3)を解決するために、レンジセンサや画像から得られた障害物の情報をもとに、どのように車両を走行させるかについて、6章では前章までに述べてきたすべての処理を使った自律走行実験とその結果を、7章では結論を述べる

2.4 オムニカメラについて

オムニカメラは、鉛直下向きと鉛直上向きの双曲面ミラーと鉛直上向きのカメラから構成されており、周辺 360 度の全方向視野画像を一画面として得ることが可能である。双曲面とは双曲線を実軸(Z 軸)周りに回転することで得られる曲面である。そして、以下の双曲線を持つ特徴も双曲面において保持される。

$$(0,0,+c) \text{ と } (0,0,-c) \text{ の 2 つの焦点を持つ。 } (C = \sqrt{a^2 + b^2})$$

ここで Fig.4 に示すように Z 軸を鉛直軸とする 3 次元座標系 O-XYZ を考える。この時 2 葉双曲面は次の式で表せる。

$$\frac{X^2 + Y^2}{a^2} - \frac{Z^2}{b^2} = -1 \quad \dots(1)$$

なお、 a, b は双曲線の形状を定義する定数である。本視覚系では、2 葉のうち $Z > 0$ の領域にある双曲面をミラーとして利用する。

オムニカメラは、Fig.5 に示すように鉛直下向きに設置した $Z > 0$ の領域にある双曲面ミラーとその下に鉛直上向きに設置したカメラから構成される。このとき、ミラーの焦点 O_M およびカメラのレンズの中心 O_c からカメラの焦点 F だけ離れた平面とする。ミラー面とその焦点 O_M 、カメラのレンズの中心 O_c は次式で表される。

$$\begin{aligned} \text{ミラー面} & \quad \frac{X^2 + Y^2}{a^2} - \frac{Z^2}{b^2} = -1 \\ \text{ミラー焦点 } O_M & \quad (0,0,+C) \\ \text{カメラレンズの中心 } O_c & \quad (0,0,-C) \quad \dots(2) \end{aligned}$$

空間内の任意の点 $P(X,Y,Z)$ に対する画像上での写像点を $p(x,y)$ としたとき、点 P の方位角 θ は次の式で表される。

$$\tan \theta = Y/X = y/x \quad \dots(3)$$

すなわち Y/X で定まる点 P の方位角 θ は、 y/x で定まる写像点 p の方位角 θ_p を算出することで得られる(Fig.6)。このように 360 度パノラマ状の領域内にある対象物体の方位角 θ が、その物体の画像面上の写像の方位として直接現れる。

また Fig.7 のように点 P と Z 軸を含む鉛直断面を想定すると、点 P と写像点 p の間には次式の関係が成り立つ。

$$\begin{aligned} Z &= \sqrt{X^2 + Y^2} \tan \alpha + c \\ \alpha &= \tan^{-1} \frac{(b^2 + c^2) \sin \gamma - 2bc}{(b^2 - c^2) \cos \gamma} \\ \gamma &= \tan^{-1} \left(\frac{F}{x^2 + y^2} \right) \quad \dots(4) \end{aligned}$$

すなわちミラーの焦点 O_M からの点 P の包圍角および伏角 α は、カメラのレンズの中心 O_c を双曲面の焦点位置にすることで、写像点 $p(x,y)$ より、一意に求まる。このとき、ミラーの焦点 O_M は固定なため、入力画像をミラーの焦点 O_M から見たカメラを鉛直軸周りに回転して得られる画像や一般のカメラの画像に変換できる。変換画像面が地面と同一になるように、変換画像の光軸を鉛直下向きにし、地面からカメラの焦点までの距離 H_r を変換画像の焦点距離とする。

このとき、変換画像面は式(3)、(4)より次式が成り立つ。

$$\begin{aligned} \tan \theta_p &= Y/X = y/x \\ R_p &= \frac{-(b^2 - c^2)H_r r_p}{(b^2 + c^2)F - 2bc\sqrt{r_p^2 + F^2}} \\ R_p &= \sqrt{X^2 + Y^2} \quad r_p = \sqrt{x^2 + y^2} \\ (R_p, \theta_p) &: P(X,Y) \text{ の極座標表現} \\ (r_p, \theta_p) &: p(x,y) \text{ の極座標表現} \quad \dots(5) \end{aligned}$$

この変換公式により地面平面への変換画像を作る。以降本研究ではオムニカメラより取得された画像を地面平面へ変化し、画像処理を行うものとする。

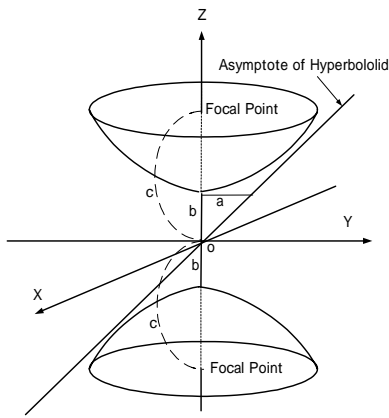


Fig. 4 Hyperboloid

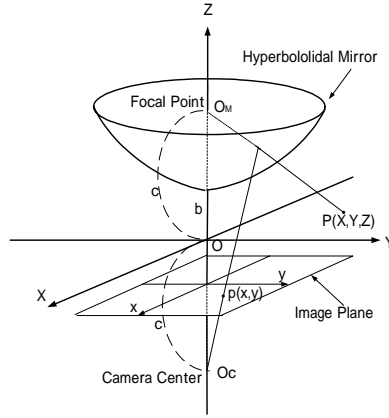


Fig.5 Hyper Omni Vision configuration

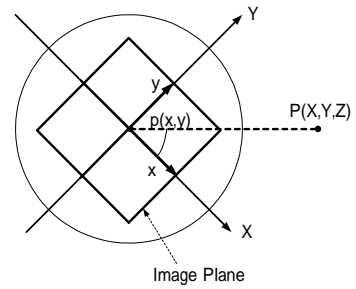


Fig. 6 Invariability of azimuth angle

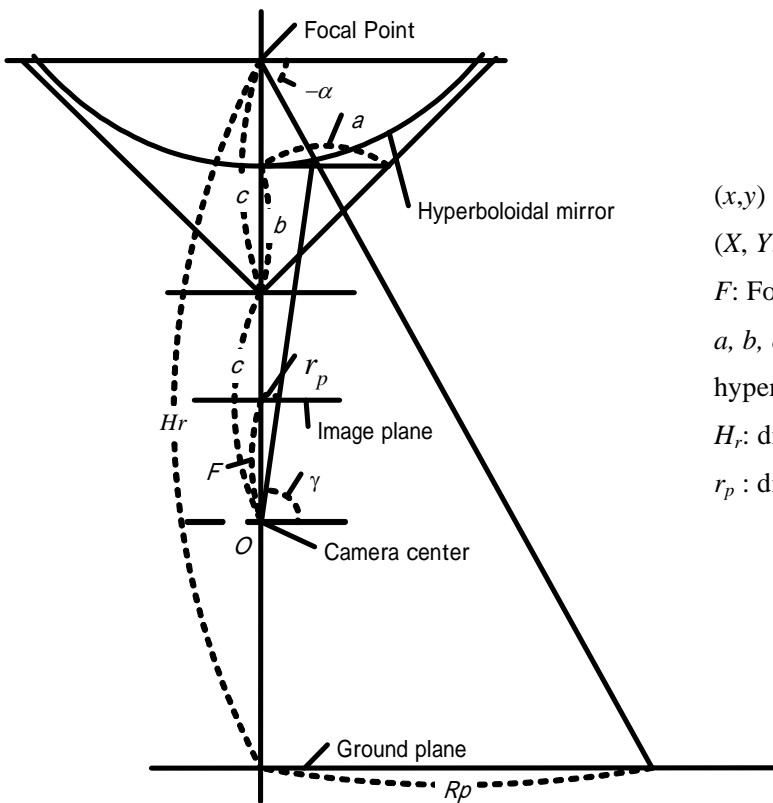


Fig. 7 Transformed to ground of tilt angle

(x,y) : x - y coordinates (for input image plane)
 (X, Y, Z) : absolute coordinates (for ground plane)
 F : Focus imaging camera
 a, b, c, α, γ : parameters determined by the hyperboloid curve
 H_r : distance from mirror to the ground
 r_p : distance of image (input)

3.画像処理による障害物検出

3.1 バイロン検出

パイロンは倒すと失格となることからレーザレンジファインダを使用して検出・回避を行う。しかし、白線検出に使用する画像にパイロンが写る場合、パイロンの白色部分は白線と誤認識されかねない。そこでパイロンは回避するだけでなく、画像中から検出・除去しなければならない。

オムニカメラによって撮影された物体は、その存在位置によって画像への写り方が異なるので、パターンマッチング等でパイロンを検出するためには複数のパターンを使用しなければならず、処理に時間がかかってしまう。レーザレンジファインダによってパイロンを検出し、その位置情報から画像中のパイロンを除去する方法も考えられるが、レーザレンジファインダは車両の前方しか計測することができないので、後方のパイロンを画像から除去できない。

そこで本研究では(A2)のパイロンは赤白の縞模様であるという仮定を利用し、画像中から赤色成分を抜き出し、その周辺を除去する手法を提案する。パイロン検出には(1)RGB成分画像のR成分を使う手法、(2)YUV成分画像のU成分を使う手法、の2種類を提案し、比較を行った。

3.1.1 R成分画像によるパイロン検出

R成分画像によるパイロン検出の処理手順を以下に示す。

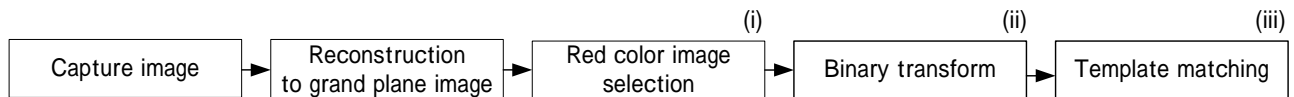


Fig.8 Pylon detection algorithm by using Red color image

RGB成分画像のR成分画像では、赤色成分と白色成分の濃淡値が高いという特徴を持っている。そこで、(i)RGB成分画像からR成分画像のみを抽出し、(ii)2値化処理を行うことでパイロンを大きな白色領域として検出できる(Fig.9(c))。しかし若干ながら、同時に白線も検出してしまっているので大きい白色領域だけを検出し除去する。

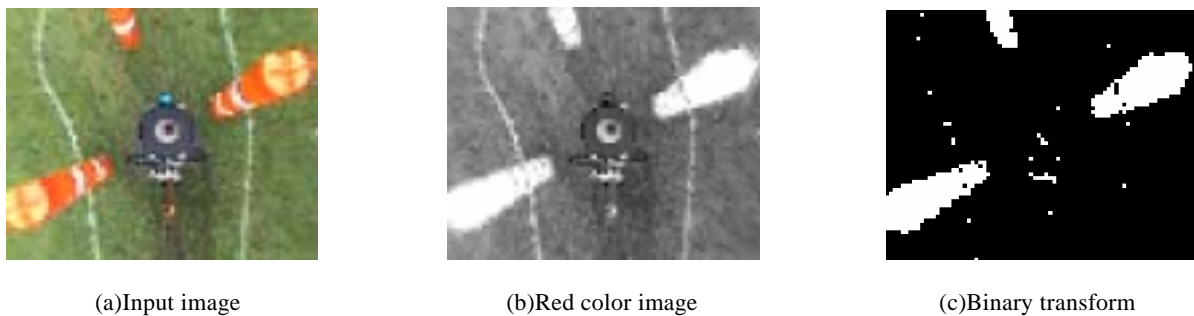


Fig.9 Pylon detection using by Red color image

ラベリング処理によって白色領域の面積を算出し、一定以上の面積を持つ領域をパイロンとみなす手法も考えられるが、Fig.10(b)のように白線とパイロンが2値化の際に融合してしまった場合、白線とパイロンを一つの塊とみなし、共に除去してしまう可能性がある。そこで、(iii)テンプレートマッチングによって大きい白色領域だけを検出する。テンプレートマッチングについては付録に示した。

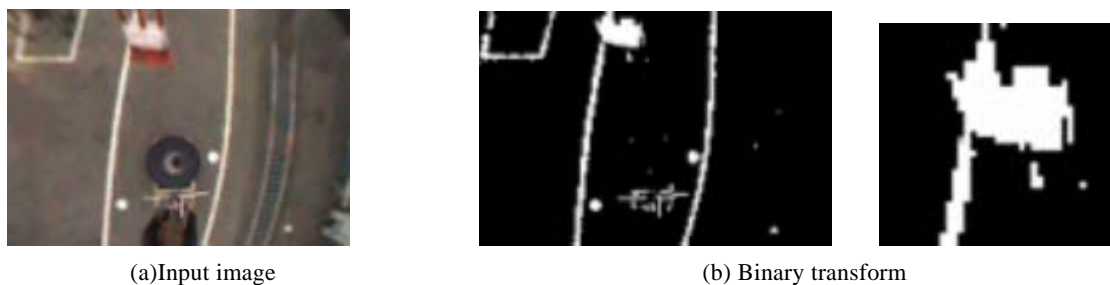


Fig.10 The case that pylon and white line are detected as one thing

3.1.2 U 成分画像によるパイロン検出

U 成分画像によるパイロン検出について処理手順を以下に示す。

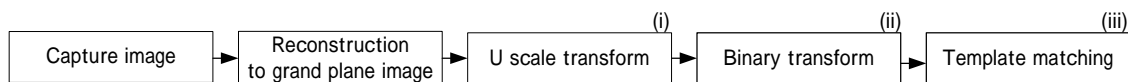


Fig.11 Pylon detection algorithm by using U color image

YUV 成分画像の色信号 U 成分画像を利用することによって、輝度によらず赤色を検出することができる。そこで、(i)RGB 成分画像から YUV 成分画像に変換し U 成分画像のみを抽出、(ii) 2 値化処理を行うことでパイロンを大きな白色領域として検出する。RGB 成分画像から YUV 成分画像へ変換は下式を用いる。

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad \cdots(6)$$

R 成分画像を使った時と異なりパイロンの白色部分は検出されていないが(Fig.12(c))、検出された領域のみではなくその周辺も除去対象とすれば、パイロンの白色領域も画像中から除去することができる。R 成分画像を使用した場合と異なり、白線を誤って検出することはないが、黄色線を検出してしまう恐れがある(Fig.13(c))。そこで R 成分画像を使った場合と同じくテンプレートマッチングによって、大きい白色領域だけを抜き出した。

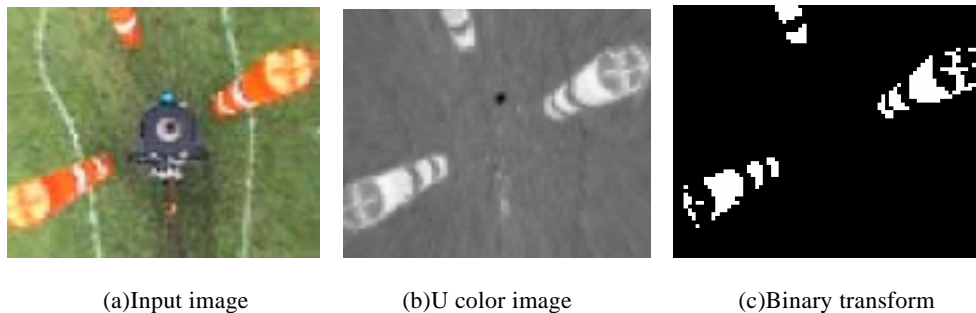


Fig.12 Pylon detection by using U color image

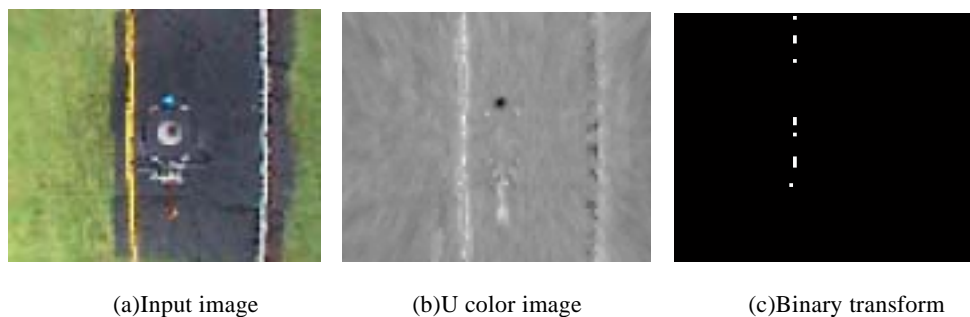


Fig.13 The case that yell or line is detected as Pylon

3.1.3 比較実験

画像データ 4730 枚に対して従来の手法と今回提案する手法を適応した。画像サイズは 65x80[pixel]。撮影場所は IGVC2002, 2004 競技コース、天候は晴天又は曇りである。R 成分画像を使った手法では、検出された領域の ± 2 [pixel] を除去対象とし、U 成分画像を使った手法では検出された領域の ± 5 [pixel] を除去対象とした。

3.1.4 実験結果

Fig.14 に示した画像ではどちらもパイロンを除去できていることがわかる。R 成分画像を使った手法では、パイロンの赤色部と白色部の両方を検出しているので、検出された領域とその周辺を多少除去すればよい。しかし、U 成分画像を使った手法では赤色部のみしか検出されないの、パイロンの白色部を除去するためには検出された領域周辺を大きめに除去しなければなら

ない。そのため、R成分画像を使用した場合と比べると、若干ながらパイロン周辺の白線まで一緒に除去してしまった。

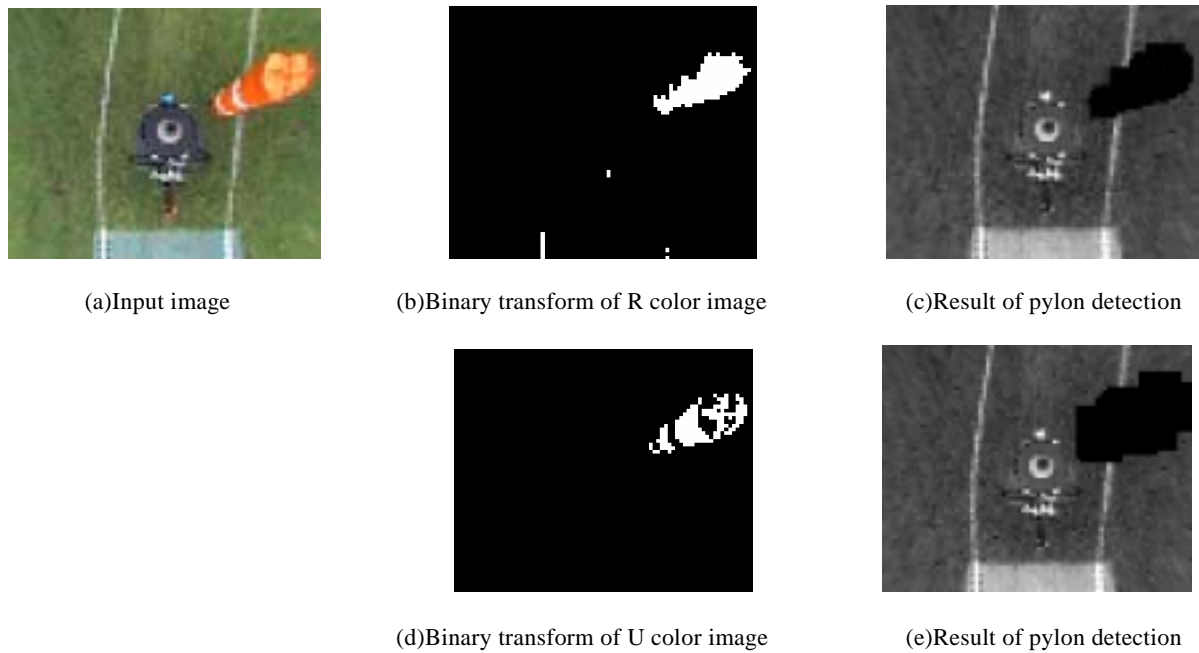


Fig.14 Pylon detection 1

Fig.15 に示すように太陽光の影響で極端に明るい場所が存在すると、R成分画像を使った手法では2値化処理によってパイロンを検出することができず、誤って明るい領域を除去してしまった(Fig.15(c))。U成分画像はR成分画像と違い白色領域を検出することがないため(Fig.15(d))、どの画像でも安定してパイロンを検出・除去することができた。

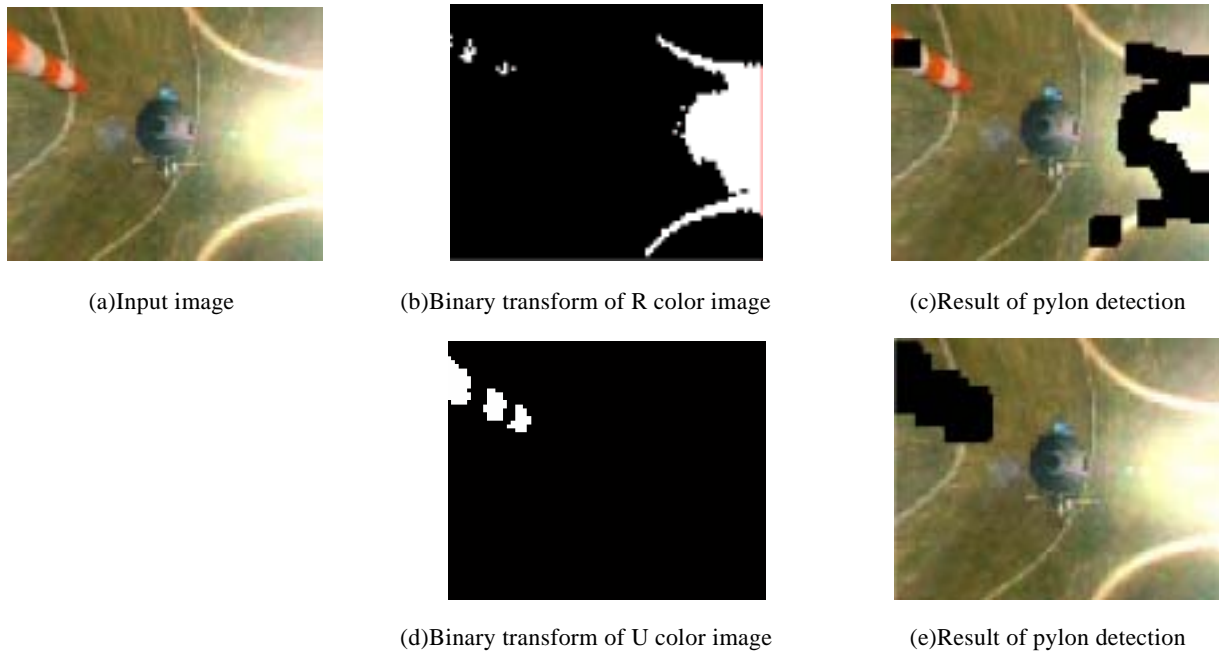


Fig.15 Pylon detection 2

3.1.5 結論

どのような画像でも安定してパイロンの検出が行えるのはU成分画像を使用した手法であった。IGVCは光量の変化が激しい屋外で開催されるため、U成分画像を利用したパイロン除去を行うこととする。

現在の処理では検出された領域周辺を単純に正形状に除去しているだけなので、パイロンと共に若干白線まで除去してしまっている。白線を除去しないように、画面に写るパイロンの方向性や形状考慮して除去するべきだと考えられる。

3.2 ポットホール検出

ポットホールは踏むと5ポイントの減点となるので回避しなければならない。そのためには正確な位置を検出することが必要である。また、ポットホールの白色は白線検出の際に悪影響を及ぼすので、画像中から検出・除去する必要がある。

パイロンは立体であるからレンジセンサによって検知することが可能であるが、地面に描かれているポットホールは検出することができない。そこで画像を使ってポットホールを検出する。仮定(A3)よりポットホールは半径0.3[m]の円であり、オムニカメラから得られた画像を地面平面に変換した画像上ではポットホールがどこに存在しても半径0.3[m]の円形状に映っている。そこで画像中から半径0.3[m]の円形を検出し、ポットホールとみなすこととした。円形検出法として(1)円形度と面積を用いた手法、(2)円形検出 Hough 変換を用いた手法、(3) テンプレートマッチングによる手法、の3つを提案する。どの手法もRGB成分画像を濃淡画像に変換した後に適応した。

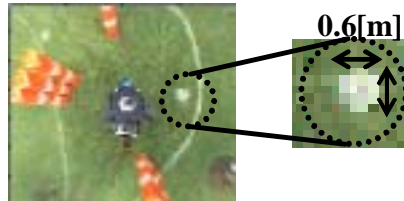


Fig.16 Pothole

3.2.1 円形度と面積を用いた手法

以下に面積と円形度を使ったポットホール検出の処理手順を示す。

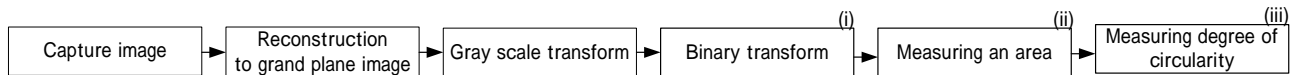


Fig.17 The algorithm of pothole detection by using circularity

(i)濃淡画像に2値化処理を行うと、ポットホールと白線が Fig.18(c)のように白色領域として検出される。ポットホールの半径は0.3[m]であるから面積は約0.28[m²]である。そこで、(ii)それぞれの白色領域の面積をラベリング処理によって算出し、約0.28[m²]の白色領域だけを検出する。次に(iii)各白色領域の円形度を算出し、閾値よりも大きなものをポットホールとして認識する。

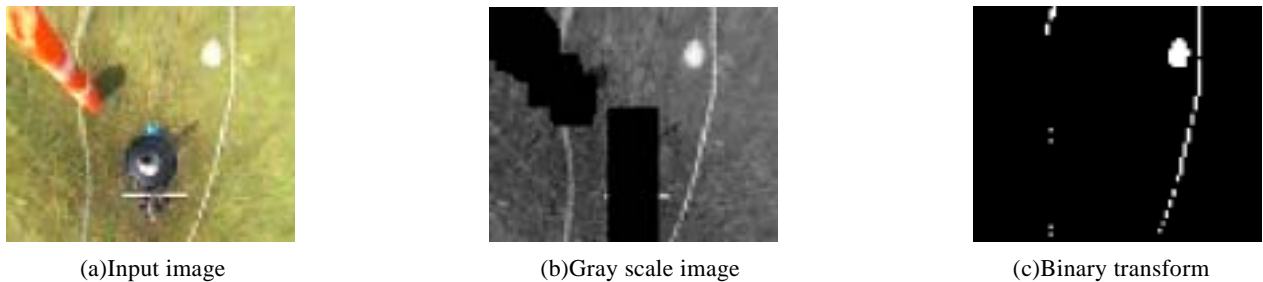


Fig.18 Result of Binary transform

円形度とは、面積・周囲長を基に、形状の複雑さを測る特徴量である。S:面積, L:周囲長 e:円形度とくと、円形度は

$$e = \frac{4\pi S}{L^2} \dots(7)$$

と定義され、0 < e < 1 である。以下に代表的な図形の円形度を示す。

Table 3 Circularity of major figure

	Circle	Square	Regular triangle
An area	πr^2	r^2	$\sqrt{3}r^2 / 4$
Length of Circumference	$2\pi r$	$4r$	$3r$
Circularity	1	$\pi / 4 = 0.79$	$\pi\sqrt{3} / 4 = 0.6$

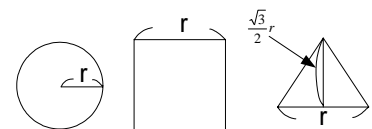


Fig.19 Sample of figure

以下に周囲長を求めるアルゴリズムを示す。

Fig.20 の画素 P_1 は、原画像をラスタ走査して最初に検出した図形画素であるとする。このとき P_1 を中心に反時計回りに探索し、最初に検出した画素を P_2 とする。これを繰り返し実行し、検出した画素 P_n が最初の画素 P_1 に一致した時点で処理を終了する。 P_1 から P_{n-1} までを領域の周囲長とする。

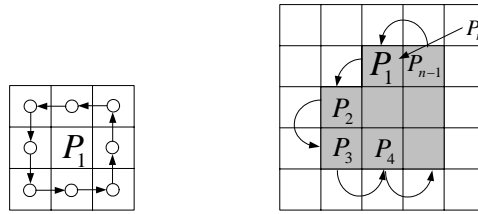


Fig.20 The algorithm of measuring length of circumferences

3.2.2 円形検出 Hough 変換

Hough 変換を用いたポットホール検出法の処理手順を以下に示す。

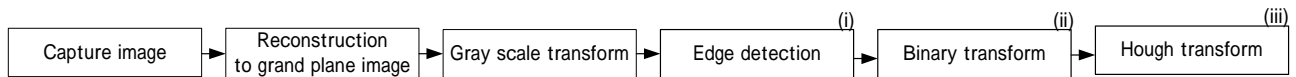


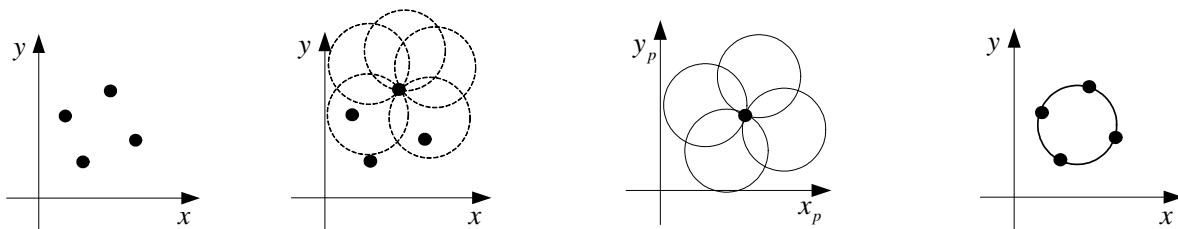
Fig.21 The algorithm of pothole detection by using Hough transform

(i)濃淡画像にエッジ検出を適応し、(ii)2 値化後、(iii)Hough 変換によって円形を検出する。Hough 変換とは直線、円、などパラメータで記述できる図形を画像中から検出するための手法である。物体の境目になっている不連続な線分を連続した境界線として抽出することが可能であり、雑音の多い図形や、途切れた図形の検出に強いという特徴を持っている。以下に円形検出 Hough 変換の処理手順を示す。

半径が r で、中心点が $P(x_p, y_p)$ である円上に、座標上の点 (x_i, y_i) が存在するならば、

$$(x_i - x_p)^2 + (y_i - y_p)^2 = r^2 \quad \dots(8)$$

の関係が成り立つ。上の式を用いて、直角座標上の点 (x, y) を新しい三次元空間 (x_p, y_p, r) 上に変換すると、直角座標上の一点は $x_p - y_p - r$ 空間上の1枚の面に対応する。直角座標上の点が多数あると、 $x_p - y_p - r$ 空間上に多数の曲面が得られる。それらの曲面が共有する点があれば、それは元の $x - y$ 直角座標上では一個の円上に並ぶことになる。逆変換は点 (x_p, y_p) を中心に半径 r の円を描けばよい。



(a)x-y coordinate (b) The examples of circle to pass one point (c)Locus of circle center (d)Detected circle

Fig.22 Hough transform for circle detection

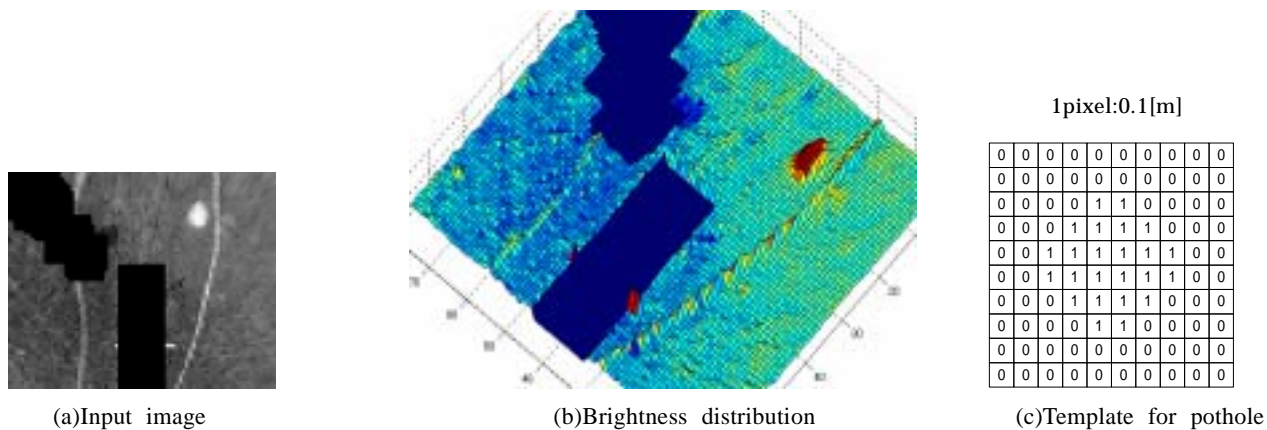
3.2.3 テンプレートマッチングによる検出

テンプレートマッチングを用いたポットホール検出法の処理手順を以下に示す。



Fig.23 The algorithm of Pothole detection by using template matching

ポットホールは白色であるから、周囲の緑色の芝よりも濃淡値が高く円形状のピークを持つという特徴を持っている。そこで Fig.24(c)に示すテンプレートをを用い、(i)濃淡画像にテンプレートマッチングを適応し、半径 0.3[m]の円形状の濃淡値のピークを検出する。



detection

Fig.24 Brightness distribution and the template for pothole detection

3.2.4 実験

画像データ 4730 枚(うちポットホールの映っている画像数は 351 枚)に対し、3つの手法をそれぞれ適応した。撮影環境は 2002 年、2004 年度IGVC 競技コースで天気は晴天、曇りである。エッジ検出には Log フィルタを、2 値化処理には判別分析法を用いた。円形度を用いた手法では円形度 0.6 以上で面積が 25[pixel]以上 35[pixel]のものを、Hough 変換を使用した手法では半径 0.2[m],0.3[m],0.4[m]の円を、テンプレートマッチングを用いた手法では Fig.24(c)に示すテンプレートをを用い相関 0.6 以上のものをポットホールとして検出した。以下にポットホール検出結果を示す。

実験結果1

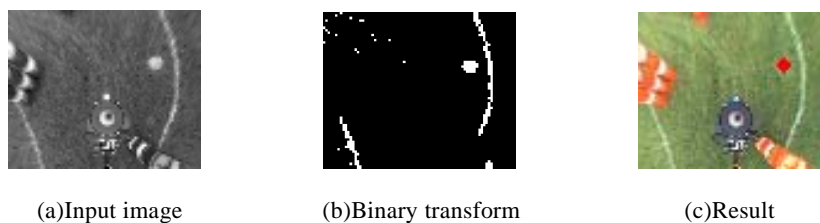


Fig.25 Pothole detection by using degree of circularity

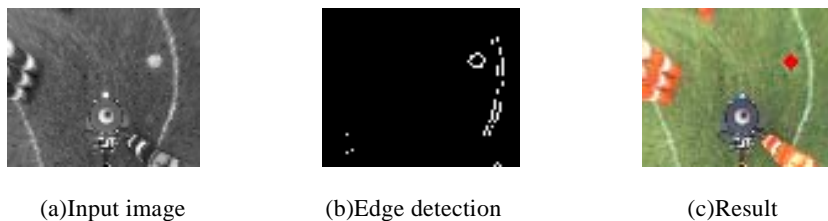


Fig.26 Pothole detection by using Hough transform

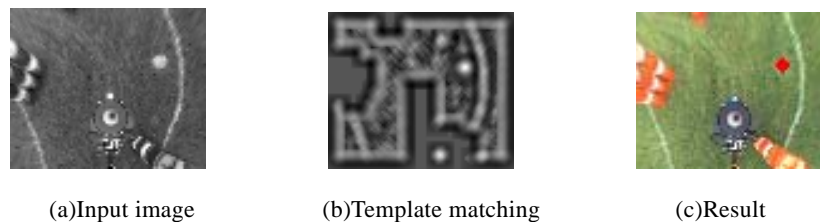


Fig.27 Pothole detection by using Template matching

実験結果 2

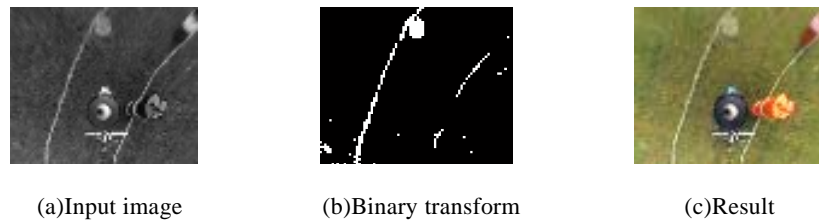


Fig.28 Pothole detection by using degree of circularity

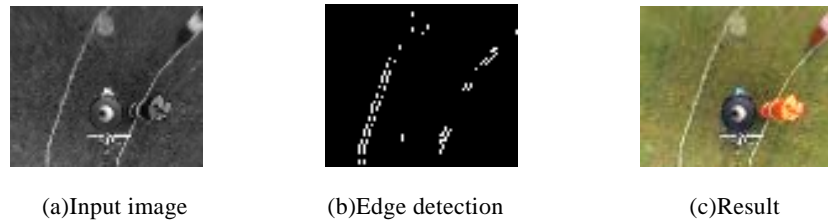


Fig.29 Pothole detection by using Hough transform

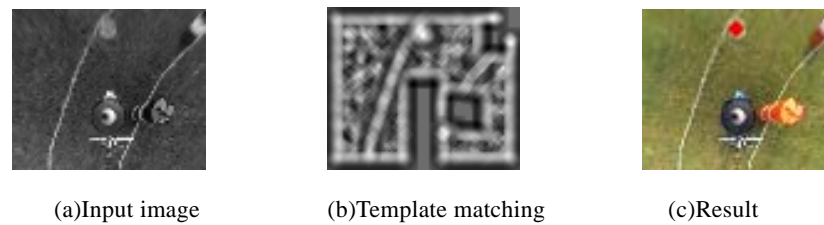


Fig.30 Pothole detection by using Template matching

実験結果 3

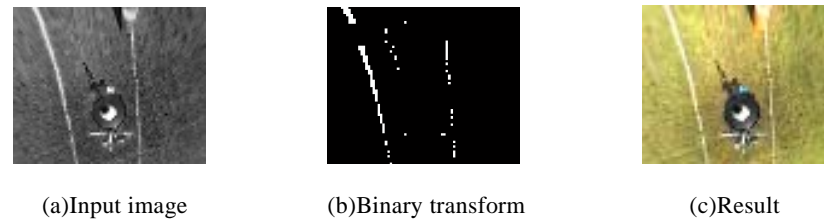


Fig.31 Pothole detection by using degree of circularity

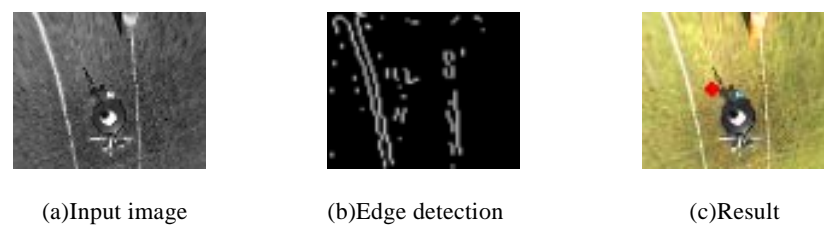


Fig.32 Pothole detection by using Hough transform

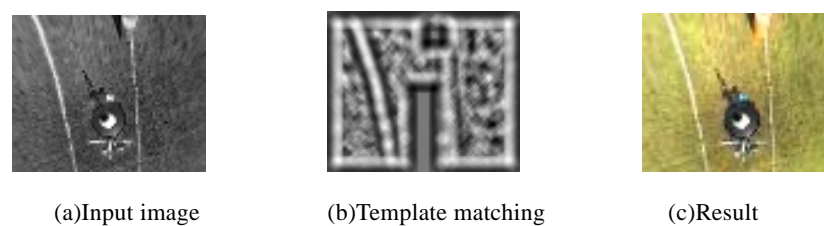


Fig.3.2 Pothole detection by using Template matching

ポットホール認識率の結果を Table 4 に示す。

Table 4 Pothole recognition rate

Method	Correct rate	Failure rate (Detected anything which was not pothole)
Detection by using degree of circularity	23%	0.5%
Detection by using Hough transform	49%	4.70%
Detection by using Template matching	83%	1.60%

3.2.5 考察

円形度と面積を用いた手法では Fig.34 のように 2 値化の際にポットホールと白線が融合して検出されてしまった場合、ポットホールと白線を一つの大きな物体と認識してしまうため、ポットホールとして検出することができなかった。また、2 値化処理によってポットホールがうまく円形状に検出されなかった場合も、円形度が極端に小さくなってしまい検出することができなかった。

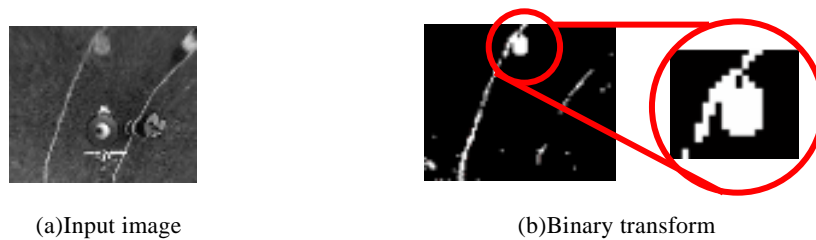


Fig.34 The case that pothole wasn't detected by using circularity

円形検出 Hough 変換を用いた手法では Fig.35 のようにポットホールのエッジがうまくとれていない場合、Hough 変換で検出することができなかった。また Fig.36 では芝生のエッジを誤ってポットホールと認識している。

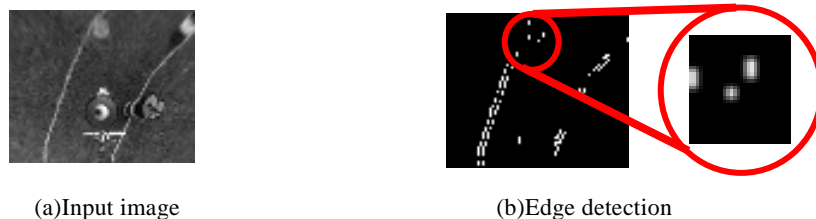


Fig.35 The case that pothole wasn't detected by using Hough transform



Fig.36 The case that lawn was detected as pothole by using Hough transform

円形度を用いた手法も円形検出 Hough 変換を用いた手法も、前処理であるエッジ検出や 2 値化のきによって結果が大きく左右されてしまう。光量変化の激しい屋外環境ではこれらの手法を適応するのは難しいと考えられる。

他の 2 つの手法と異なり、テンプレートマッチングを用いた手法はエッジ検出や 2 値化処理といった前処理を必要としないため、安定してポットホールを認識することができた。

3.2.6 結論

光量変化の激しい屋外環境では 2 値化やエッジ検出といった前処理を必要とする手法の適応は困難である。前処理を必要とせず、どの画像でも安定して検出できたテンプレートマッチングを用いた手法を自律走行に採用する。

3.3 バケツ検出

パイロンやポットホールと同じくバケツも回避しなければならない。またバケツは白色であるので、パイロンやポットホールと同様に画像中から除去することも必要である。バケツは立体であるからレーザレンジファインダによって検出することは可能であるが、バケツを検出するためにはレーザレンジファインダを低い位置に設置しなければならない。しかしながら低い位置に設置してしまうと、地面や芝生を誤って検出してしまいう危険性が生じる。そこで、本研究では画像を利用してバケツの位置を検出する。

バケツはポットホールと違い高さ成分を持っているので、存在する位置によって画像への写り方が異なる。そのためテンプレートマッチングで検出するためには、多くのテンプレートが必要になり、マッチングに時間がかかってしまうという問題がある。そこで、バケツは芝生に描かれた白線よりも白という特徴に着目し、画像中から極端に白い領域を検出する。

RGBすべての成分が閾値よりも大きい濃淡値を持つ場所を検出する。しかしそれだけでは Fig.37(b)に示すように白線まで一緒に検出してしまいう可能性がある。ラベリング処理によって白色領域の面積を算出し、一定以上の面積を持つ領域をバケツとみなす手法も考えられるが、2値化処理によって白線とバケツが融合して検出された場合、白線とバケツを一つの塊として共に除去してしまう可能性がある。そこで、テンプレートマッチングによって大きい白色領域だけを検出する。

検出された白色領域をクラスタリングし、その重心をバケツの位置とし回避制御に用いる。検出された位置を赤三角形で画像上に示した。また Fig.36(d)に示すように、検出された領域周辺を除去することによって画像中からバケツを除去することができる。

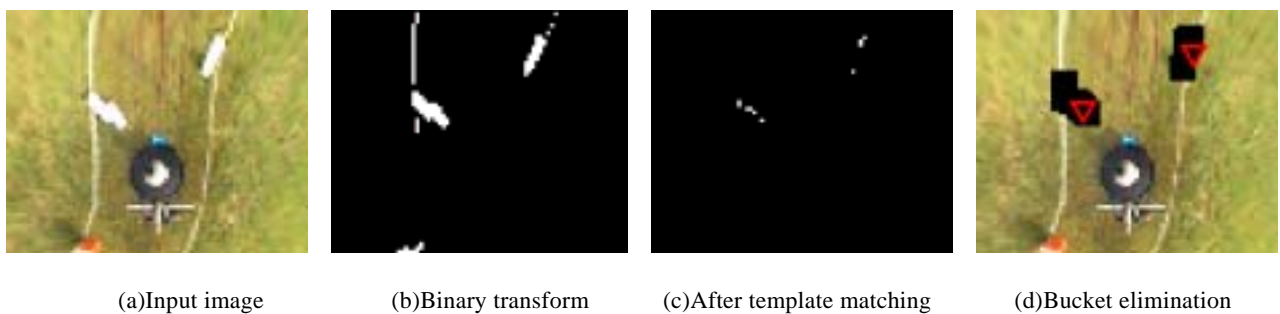


Fig.37 Bucket detection

しかし、このままでは Fig.37(a)に示すように太陽光の影響で極端に明るい場所が存在するとバケツとして誤検出してしまいう。そこで、テンプレートマッチングによって抜き出された領域に膨張処理を適応した後(Fig.38(b))、白色領域の面積を算出し、面積が閾値よりも大きいものは太陽光とみなし除去を行った。Fig.37(c)ではバケツとして検出されていないことがわかる。

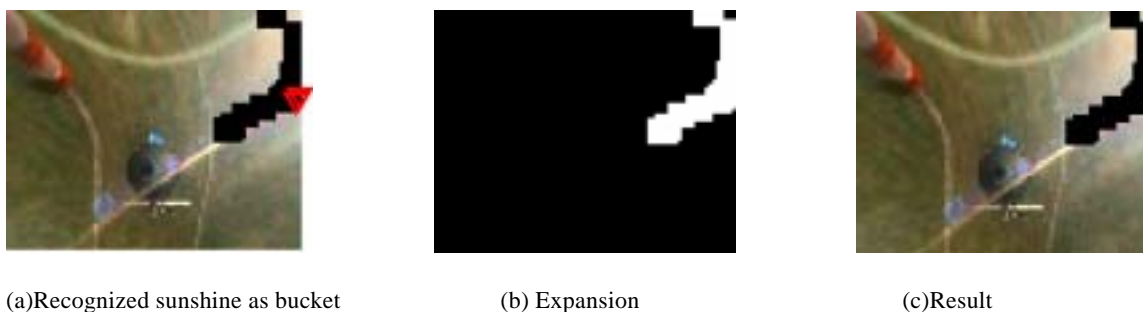


Fig.38 Bucket detection 2

4.白線検出

前章までで述べてきた手法を使い、画像中からパイロン・ポットホール・バケツを除去した画像から白線を検出する。

4.1 従来手法

これまでの研究では濃淡画像の濃淡値の大きさのみによって白線を検出してきた²⁾。しかし、日光が強く当たる明るい芝や、白色の枯れ芝があると、白線と誤認識してしまうことがあった。固有空間法を利用しRGB画像の色によって白線を検出する手法も考えられているが³⁾、ある画像では白線であった色が、別の画像では芝生の色であるという状況もありえ、色によって白線と芝生を識別することは困難であると考えられる。

本論では白線検出を濃淡値の大きさのみではなく、形状という新しい指標を導入して行なう。

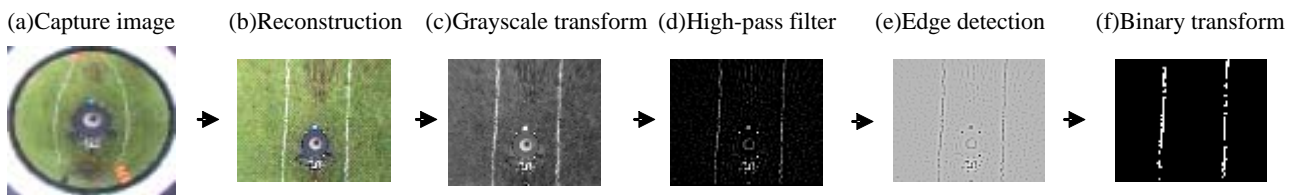


Fig.39 Conventional method of line detection



Fig.40 An example of mistaken line detection in dry lawn area



Fig.41 An example of mistaken line in strong sunshine area

4.2 形状の検出

ACC では仮定(A5)より白線の幅に規定が設けられている。さらに白色であるということから、光の当たり加減によって濃淡値の大きさは場所によって異なるものの、周りの芝生よりも濃淡値が大きいという特徴を持っている。

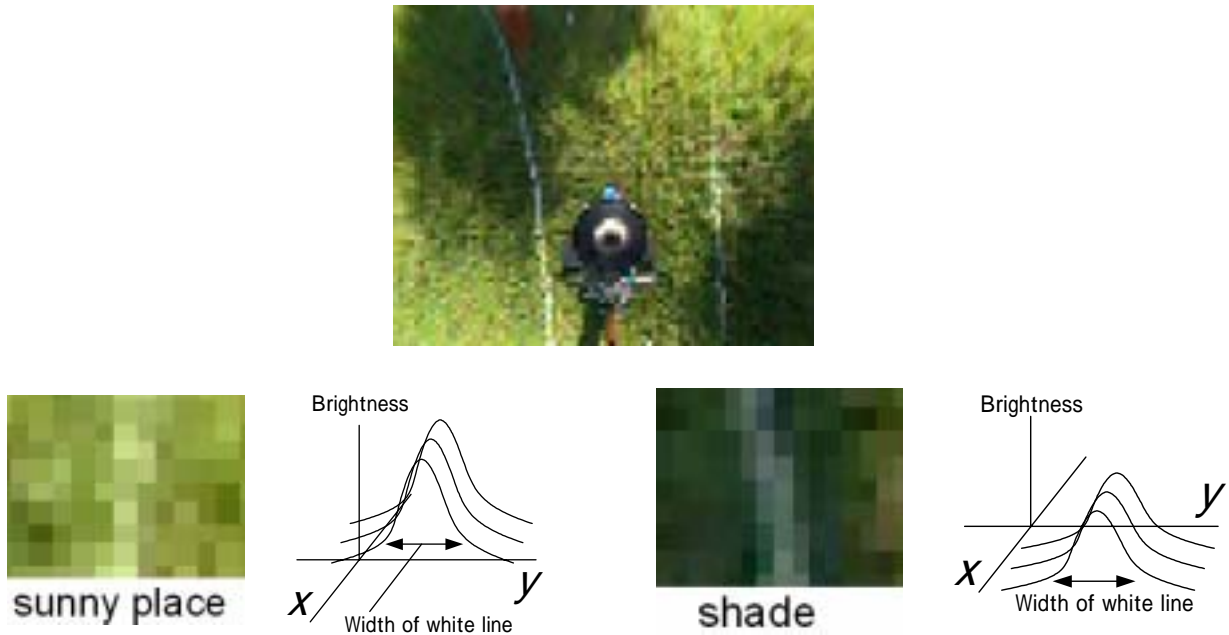


Fig.42 Brightness distribution of white line

そこで本研究ではテンプレートマッチングによって、Fig.42 に示す2種類のテンプレートを用い、縦方向もしくは横方向に0.15[m]幅の濃淡値のピークを持つエリアを検出する。形状で白線を抜き出しているため、誤って白線検出の際にポットホールを検出してしまわない。従って、ポットホールは踏んでしまう可能性のある領域だけ検出すればよく、車両の側方や後方のポットホールを検出する必要がなくなるため処理速度の向上にもつながる。パイロンの白色部やバケツについては、存在する位置によっては白線と同じ太さに写る場合があり、誤って白線として検出されかねないので引き続き画像全体から除去する必要がある。

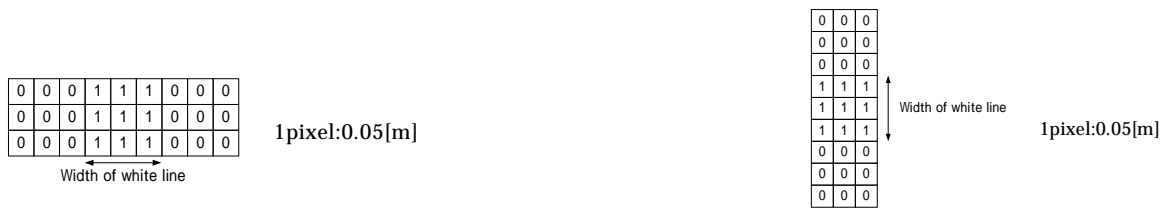


Fig.43 The templates for white line detection

相関が閾値よりも大きいエリアを白線として検出すれば良い。以下に示した画像は相関 0.6 以上のものを検出した。

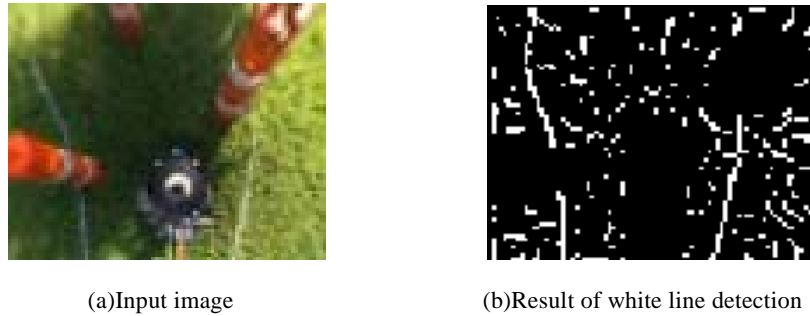


Fig.44 Result of binary transform by template matching 1

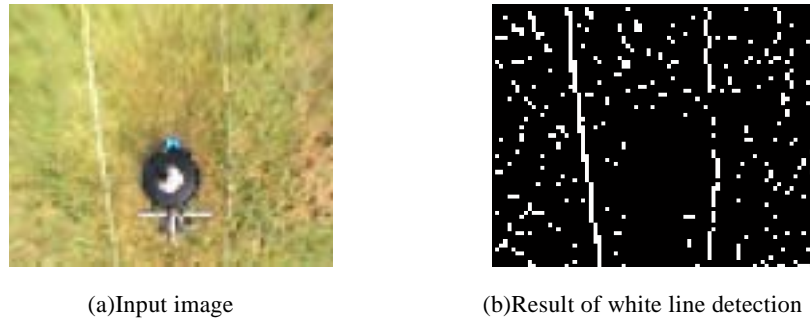


Fig.45 Result of binary transform by template matching 2

どのような画像でもほぼ確実に白線を検出することができた。しかし上図に示すようにノイズを多く含んでいるのでこれらを除去する必要がある。以下2種類のノイズ除去を提案する。

4.3 ノイズ除去

4.3.1 濃淡値の分布によるノイズ除去

ノイズの原因としては白線と同じ形・幅のピークは持つが、あまり周囲と濃淡値に差がないものまで検出してしまっているということが挙げられる。そこで以下で述べる手法によってテンプレートと同じ形状を持っていても、ピークの勾配が低いものは白線ではないとして除去を行う。テンプレートマッチングで検出された座標の濃淡値を M 、その周囲 24 近傍の中でテンプレートマッチングによって白線として検出されていない座標の平均濃淡値を N 、閾値を α 、 β とおいた時、式(9a)(9b)をみたさないものを除去する。

$$N/M > \alpha \quad \dots(9a)$$

$$M - N > \beta \quad \dots(9b)$$

しかし、光や枯れ芝の影響で芝生が比較的白く写る場合は、白線と芝生の濃淡値に大きな差がみられない。そこで、比較的明るい領域では閾値をそれぞれ α_2 と β_2 に変更する。本研究では画像全体の濃淡値の平均値よりも、24 近傍の平均値が高く、画像全体の分散よりも 24 近傍の分散が小さい場所を明るい場所と定義し閾値の変更を行った。

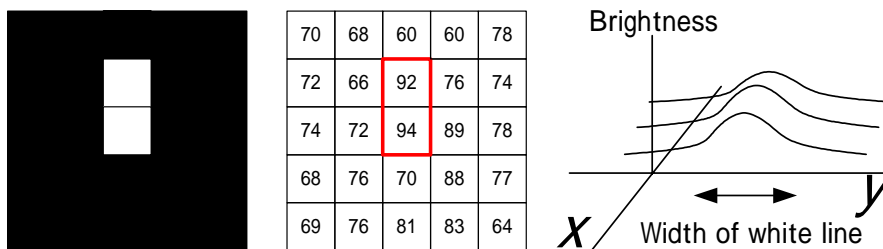


Fig.46 Brightness distribution of noise

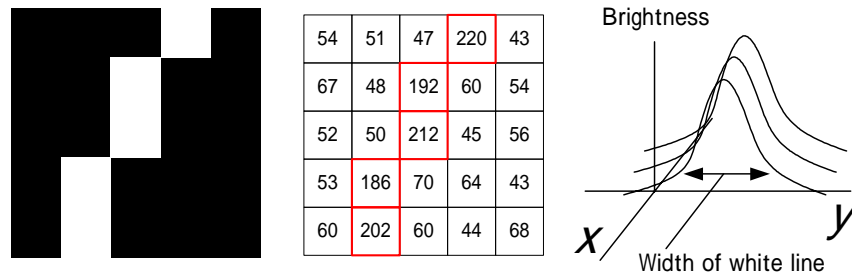


Fig.47 Brightness distribution of lane

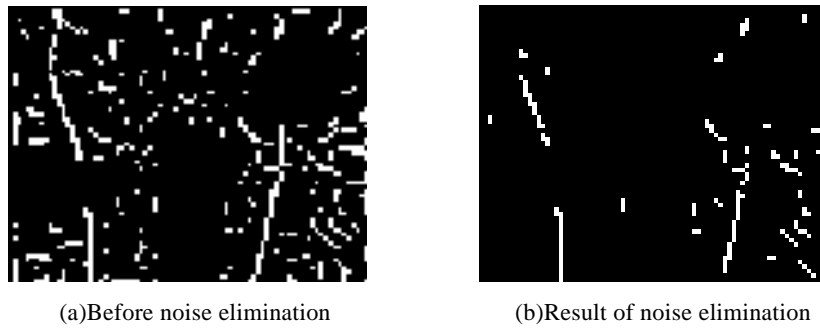


Fig.48 Noise elimination by using brightness distribution

4.3.2 時系列画像によるノイズ除去

白線は連続していることから、現在取得した画像に写っている白線は、1サンプリング前に取得した画像の白線の周辺に存在する。そこで、1サンプリング前の画像に膨張処理を行い、現在の画像から得られた 2 値化画像との積をとる。これによって現在得られた画像から、1サンプリング前の画像の白線周辺にない白線を除去することができる。



Fig.49 Noise elimination by using chronological image

Fig.50 に提案する白線検出法の手順を示した。

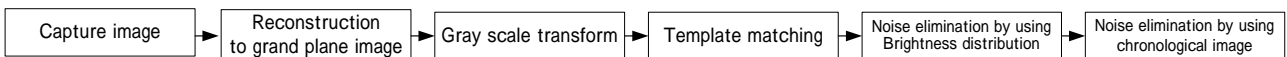


Fig.50 Line detection algorithm

4.4 実験

画像データ 4730 枚に対し、3つの手法をそれぞれ適応した。撮影環境は 2002 年、2004 年度IGVC競技コースであり、天気は晴天、曇りである。テンプレートマッチングでは 0.6 以上の相関を持つ領域を検出し $\alpha = 0.8$, $\beta = 15$ $\alpha_2 = 0.9$ $\beta_2 = 5$ とした。

4.5 結果

白線検出結果を以下に示した。

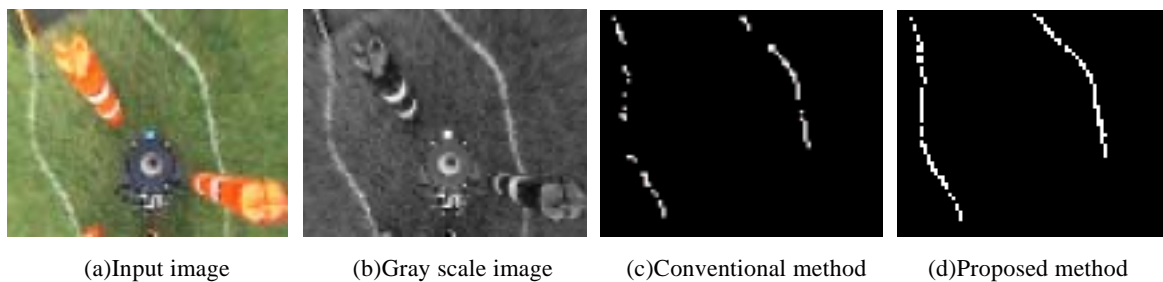


Fig.51 Result of white line detection 1

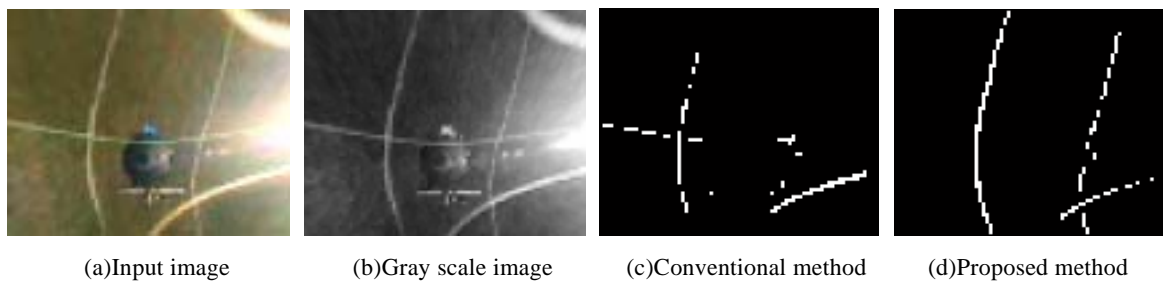


Fig.52 Result of white line detection 2

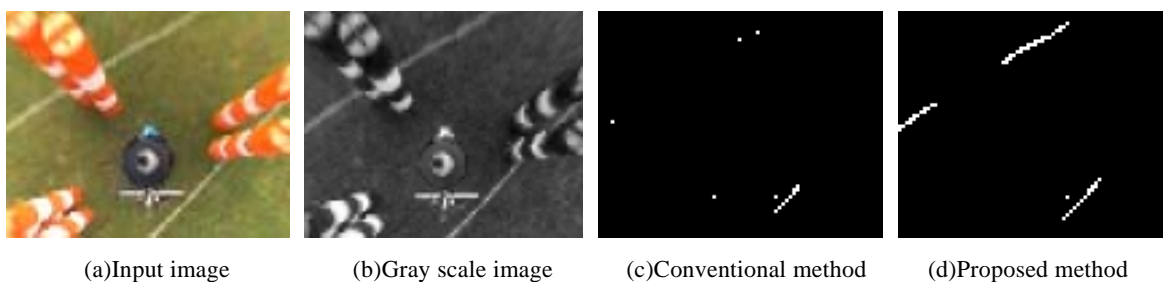


Fig.53 Result of white line detection 3

Table 5 White line recognition rate

	Correct rate	Failure rate (Detected anything which was not white line)
Conventional method	49%	12%
Proposed method	68%	13%

4.6 考察

従来の手法と比べて検出率を20%近く改善することができた。また従来法では日光の強さによって検出精度にばらつきが生じたが、提案する手法ではどのような画像でも比較的安定した結果が得られた。しかし従来法の方が検出精度の高い画像も相当数存在した事から、従来法と共存できるアルゴリズムも検討する必要があると考えられる。また、一概に誤検出といっても誤って検出した位置やその方向性によってはその後の処理に与える影響は異なるため、詳しく分析する必要があると考えられる。

と はそれぞれともに実験より最適と思われる値に定めたが、白線とその周囲の芝の色によって理論的に求めることができると考えられる。そこで今後は色空間を使った閾値決定法などを検討する必要があると考えられる。

4.7 Hough 変換による直線検出

前述までの手法によって画像中から白線を検出することができた。しかし、作成された 2 値化画像には Fig.54(a)のようにノイズが入っていたり、Fig.54(b)のように白線が不連続に検出されていたりする場合がある。そこで、雑音の多い図形や、途切れた図形の検出に強い Hough 変換の特徴を利用して、2 値化画像中から直線成分を検出しコースレーンとする。



(a) Binary transformed image include sunshine noise

(b) Binary transformed image include line of discontinuity

Fig.54 Lane detection

Hough 変換の結果より単純に強い直線成分 2 つを白線として選んでしまうと、Fig.55(a)のようにノイズを白線と誤認識してしまう場合が生じる。そこで、本研究では仮定(A6) (A7)を利用し、2[m] ~ 6[m]幅を持ち平行に存在する直線を検出することにした。



(a) The case that noise was detected as course lane

(b) The case that correct lane is detected

Fig.55 Linear detection

4.7.1 提案する手法

2 値化画像から平行な直線を検出する手法を以下に示す。

- () ρ, θ, l パラメータ空間を適当に量子化する。
- () ある 1 点(x,y)を通る直線の式を $\rho = x \sin \theta + y \cos \theta$ で表す。
- () ρ, θ を $\rho - \theta - l$ 空間のすべてに投票する。
- () 直線から l 離れた場所に平行な直線を想定し、その直線上に存在する点の数を $\rho - \theta - l$ 空間に追加投票する。
- () l の値を変え、同様に追加投票する。
- () 投票数が最大になった ρ, θ, l を元の直角座標に戻したのが、 l だけ離れて存在する平行な直線らしい点の集まりである。

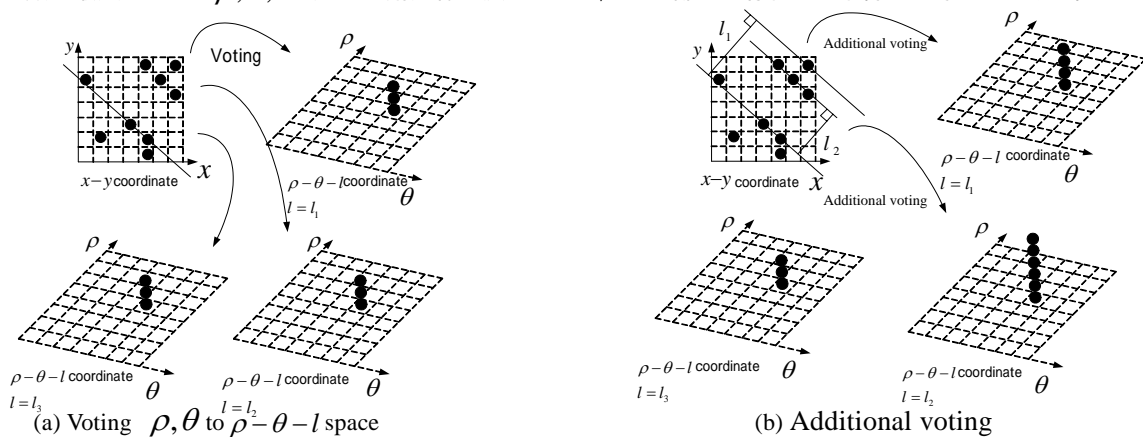


Fig.56 Proposed method of lane detection based on Hough transform

4.7.2 レーン検出実験

画像データ 4730 枚に対し、3つの手法をそれぞれ適応した。撮影環境は 2002 年、2004 年度IGVC競技コースであり、天気は晴天、曇りである。Hough 変換によって 2.5[m],3[m],3.5[m],4[m],4.5[m],5[m],5.5[m],6[m]の間隔で平行に存在する直線を検出した。

4.7.3 実験結果

レーン検出結果を以下に示す。

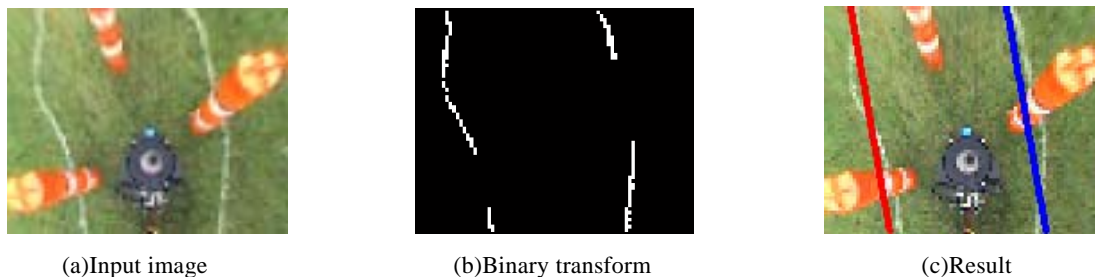


Fig.57 Result of Lane detection 1

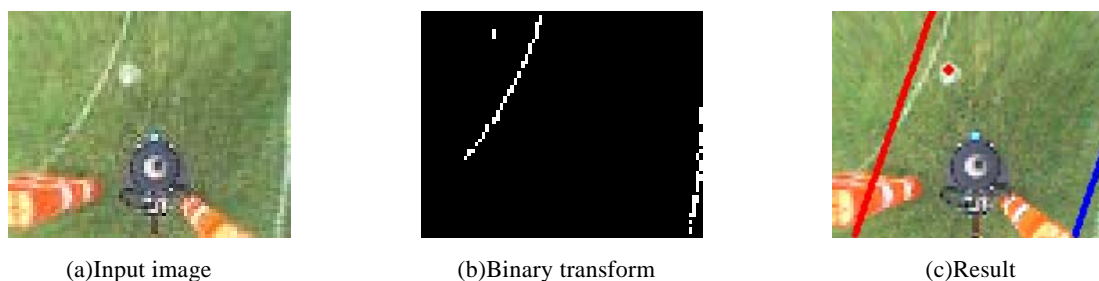


Fig.58 Result of Lane detection 2

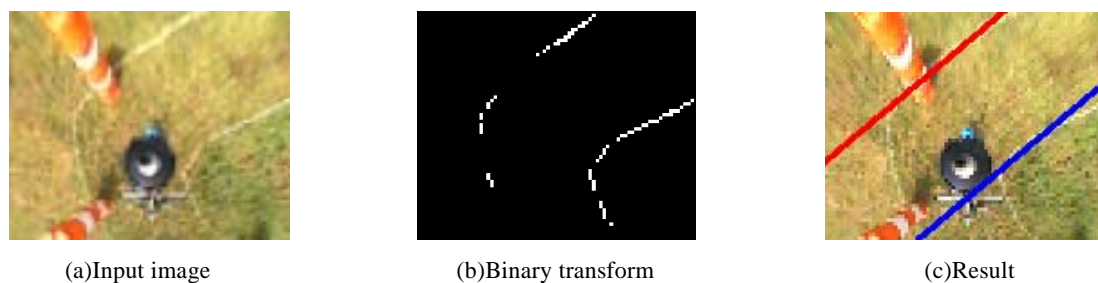


Fig.59 Result of Lane detection 3

4.7.4 考察

白線間隔の変化するコースに対して正確にレーンを検出することができた。しかし、Fig.59 に示すようなカーブでは直前までカーブ先のレーンを検出することができないので、1組の直線を検出するだけでなく、複数の平行な直線を検出するべきであると考えられる。

本手法ではあらかじめ指定した間隔で存在する平行線しか検出できないので(上記実験では 2.5[m]から 0.5[m]刻みで 6[m]までの平行線のみ)、白線間隔の細かい変動には対応することができない。また、2本の白線は実際には常に平行に存在するわけではないため、検出されたレーンと実際の白線には若干の誤差が生じた。そこで、2本の白線の傾き具合や間隔に関してある程度自由度のある平行線検出法の構築が今後の課題となる。

5. 走行制御

5.1 章ではレーザレンジファインダを使用したパイロンの検出法について, 5.2 章では目標走行ラインの決定法について, 5.3 章では回避経路に追従するための制御則について述べる.

5.1 レーザレンジファインダによるパイロンの検出

3章までの画像処理によってレーン・ポットホール・バケツの位置が得られている. ポットホールやバケツと同様にパイロンの位置も画像中から検出することは可能であるが, レーザレンジファインダを使えば安定して高い精度で検出することが可能である.

5.1.1 探索エリア

レーザレンジファインダは一定の角度サンプル間隔で角度情報と距離情報を取得できる. レーザレンジファインダでパイロンを検出するにあたり, 本研究ではある一定の r_a 以内の警告エリアに立体が存在する時のみ, パイロンの検出を行うことにした. これは r_a 以上のエリアにパイロンが存在しても走行上影響がなく, 余分な計算処理を行わないようにするためである. また, 4 章で述べた画像処理によって白線の位置は検出されていることを利用し, 警告エリア内に存在しても白線内に存在しないパイロンも同様に無視するものとした.

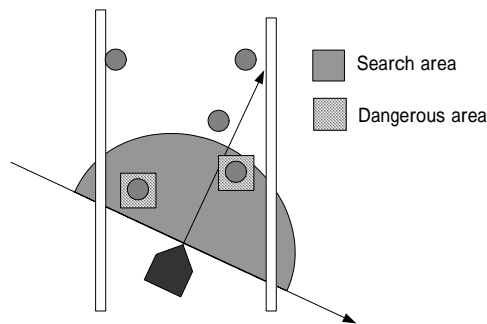


Fig.60 An example of pylon arrangement

5.1.2 トラップの回避

IGVCでは Fig.61(a)のような袋小路によるパイロンのトラップが存在する. この回避のため, 警告エリアに障害物が存在するときは探索距離を r_a から r_e に切り替えることにする. これにより, より先のパイロンを探索することができ, トラップ回避可能な情報を得ることができる. このようにして得られる情報を基に, パイロンをクラスタリングし, 一つの大きなパイロンとして認識し, 回避経路を探索する(Fig.61(c), (d)).

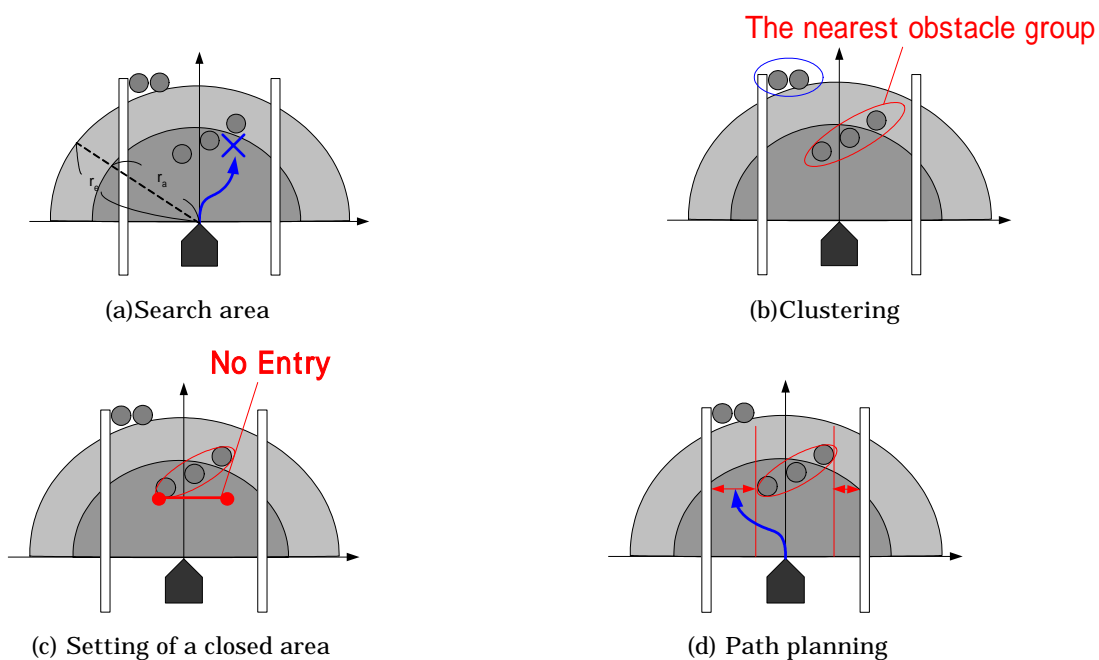


Fig.61 Pylon detection and path planning

5.1.3 坂道により地面を検出した場合の対処法

Fig.62 の場合において、地面を障害物として誤認識してしまう。この対処法として、クラスタリングされた障害物の長さを測定し、コースのレーン幅よりも大きい立体は地面を検出したと判別し、パイロンとはみなさないものとした。

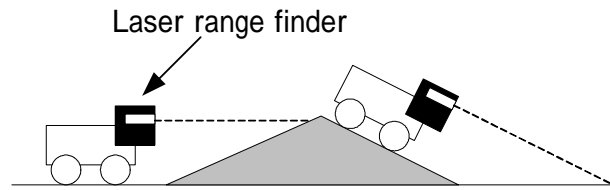


Fig.62 The cases vehicle found the ground as obstacles

5.2 目標走行ラインの選択

画像処理で得られた白線情報とポットホール位置、バケツ位置、レーザレンジファインダから得られたパイロン位置の情報を基に、目標走行ラインを決定する。IGVCでは以下の3種類の状況が考えられるため、それぞれ以下のように走行ラインを置く。

- (a) 回避すべき障害物が存在しない。障害物が存在しないのでどこを走行してもよい。
- (b) 障害物(群)が1つだけ存在する。(i)左白線と障害物の間、(ii)障害物と右白線との2通りの走行ラインがある。
- (c) 障害物(群)が2つ存在する。(i)左白線と左障害物の間、(ii)左障害物と右障害物の間、(iii)右障害物と右白線の間、の3通りの走行ラインが考えられる。

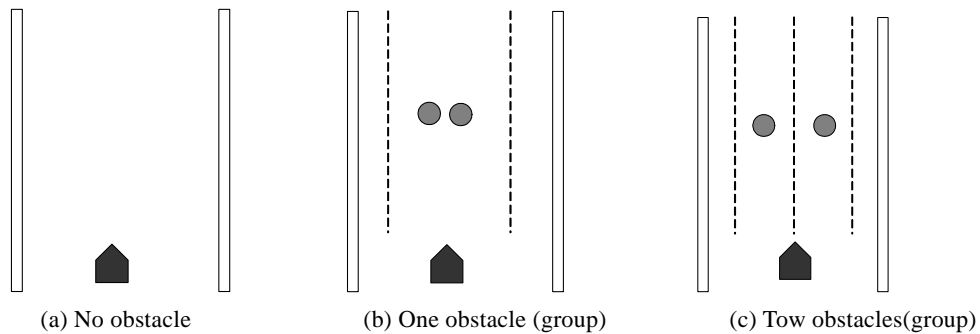


Fig.63 Obstacles and objective traveling line

- (a)の障害物が存在しない場合は、左右白線間の中心を目標走行ラインとし、そこに追従するよう制御を行なう。
- (b)(c)では左白線と障害物の間、障害物と右白線の間、障害物間に目標通過点を置き、その地点を通過する際に車体が白線と平行になるように走行できる経路を地点追従法によって算出する。障害物と衝突する危険性のある走行ラインや、白線外に出てしまう可能性のある走行ラインを追従の選択肢から除外する。残った走行ラインの中で、最も通行可能幅の広いところを追従する目標走行ラインとして選択する。

どの経路を走行しても障害物に衝突したり、白線外に出てしまったりすると判断された場合は車両を後退させ、再度白線認識からやり直すものとした。

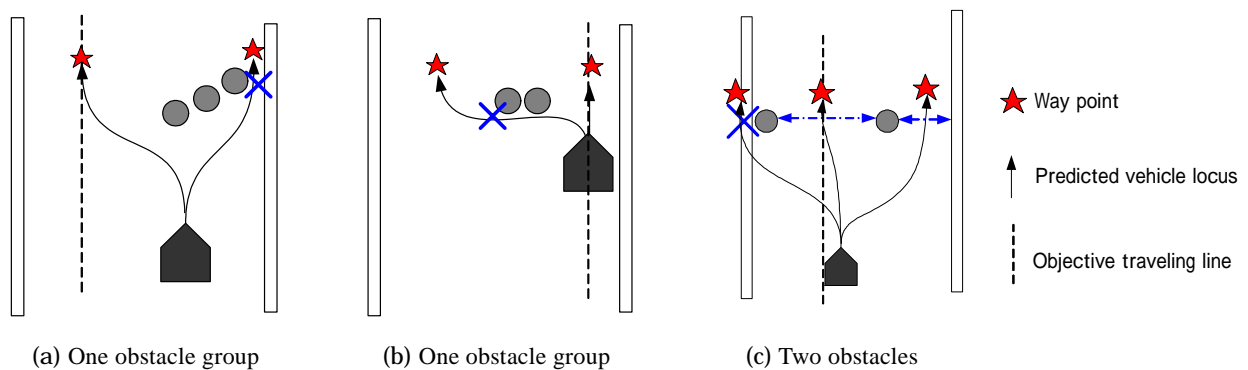


Fig.64 Path planning

5.3 目標ラインの追従(微分先行型PD制御)

目標走行ラインに追従するよう制御を行なう。目標走行ラインの ℓ [m]先の地点を目標点とし、そこへ向かうよう制御を行う。車両から目標ラインまでの位置偏差を d とおけば、

$$\theta_1 = \tan^{-1} \frac{d}{\ell} \quad \dots(10)$$

さらに車両と白線のなす角を θ_2 とおけば、目標ステアリング角 θ_{ref} は

$$\theta_{ref} = \theta_1 + \theta_2 \quad \dots(11)$$

と表すことができる。

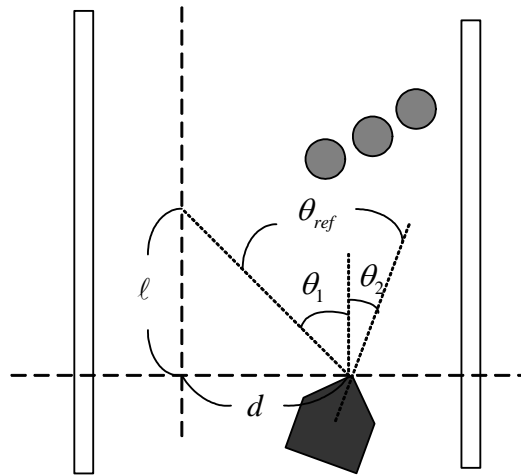


Fig.65 Steering angle determination

次に車両の運動方程式は以下ようになる。

$$J\ddot{\theta} = -c\dot{\theta}(t) + \tau(t) \quad \tau(t) = k_f v(t) \quad \dots(12)$$

J [kg·m]: 車輪の慣性モーメント

c : 転がり摩擦抵抗

$\tau(t)$ [Nm]: 車輪に加わるトルク

k_f : トルク/電圧変換係数

$v(t)$: コントローラに加える電圧

角度制御を考えた場合、制御量は $\theta(t)$ 、操作量は $v(t)$ であるから伝達関数は以下ようになる。

$$P(s) = \frac{\theta(s)}{v(s)} = \frac{b}{s(s+a)}, \quad a = \frac{c}{J}, b = \frac{k_f}{J} \quad \dots(13)$$

(13)式に含まれるパラメータ a, b は測定器で図ることができない未知のパラメータであるから実験によりパラメータ同定を行う。

$y_d(t) = \dot{\theta}(t)$ とすると(13)式は

$$J\dot{y}_d(t) = -cy_d(t) + k_f v(t) \quad \dots(14)$$

$$\frac{y_d(s)}{v(s)} = R_d(s) = \frac{b}{s+a}, \quad a = \frac{c}{J}, b = \frac{k_f}{J} \quad \dots(15)$$

となる。上式を1次遅れ要素の標準形で書き表すと

$$R_d(s) = \frac{K}{1+sT}, T = \frac{1}{a}, K = \frac{b}{a} \quad \dots(16)$$

であるから, $v(t)=1[V]$ を加えたときの加速度 $y_d(t) = \dot{\theta}(t)$ は Fig.66 のようになる. したがって, Fig.66 から時定数 T , ゲイン K を求めれば, 未知パラメータ a, b が次式のように定まる

$$a = \frac{1}{T}, b = K \cdot a = \frac{K}{T} \quad \dots(17)$$

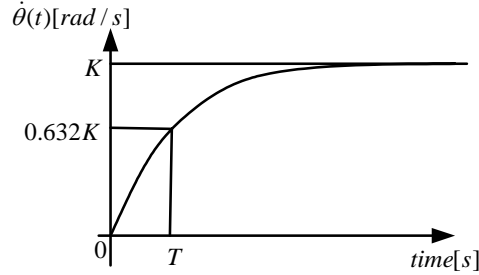


Fig.66 Approximated step response of angular velocity

次にコントローラの設計を考える. P-D コントローラ(微分先行型 PD コントローラ)は

$$v(t) = k_p e(t) - k_D \frac{d\theta(t)}{dt} \Leftrightarrow v(s) = k_p e(s) - k_D s \theta(s) \quad \dots(18)$$

$$e(s) = \theta_{ref}(s) - \theta(s)$$

θ_{ref} : 目標角度 θ : 動作角度

で与えられる.

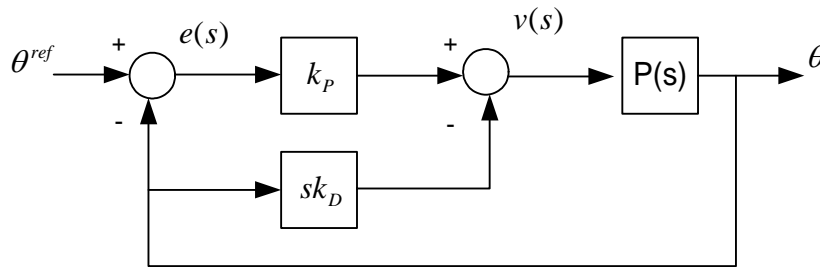


Fig.67 P-D controller

ここで $\theta^{ref}(s)$ から $\theta(s)$ への伝達関数は2次遅れ要素

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \omega_n = \sqrt{bk_p}, \zeta = \frac{a + bk_D}{2\omega_n} \quad \dots(19)$$

となる. したがって, 上式を2次遅れ要素の規範モデル

$$T_M(s) = \frac{\omega_M^2}{s^2 + 2\zeta_M \omega_M s + \omega_M^2} \quad \dots(20)$$

と完全に一致させるには, k_p, k_D を

$$k_p = \frac{\omega_M^2}{b}, k_D = \frac{2\zeta_M \omega_M - a}{b} \quad \dots(21)$$

と定めればよい.

本車両では, 移動速度が 0.6[m/s]の時, $a=1.45, b=33.3$ であった.

車両制御に用いたゲインは $\omega_M = 3, \zeta_M = 0.7$ とした.

6. 走行実験

第3章から第5章までに提案してきた手法を用い、実際に自律走行実験を行った。Fig.68 に自律走行のためのアルゴリズムを示した。

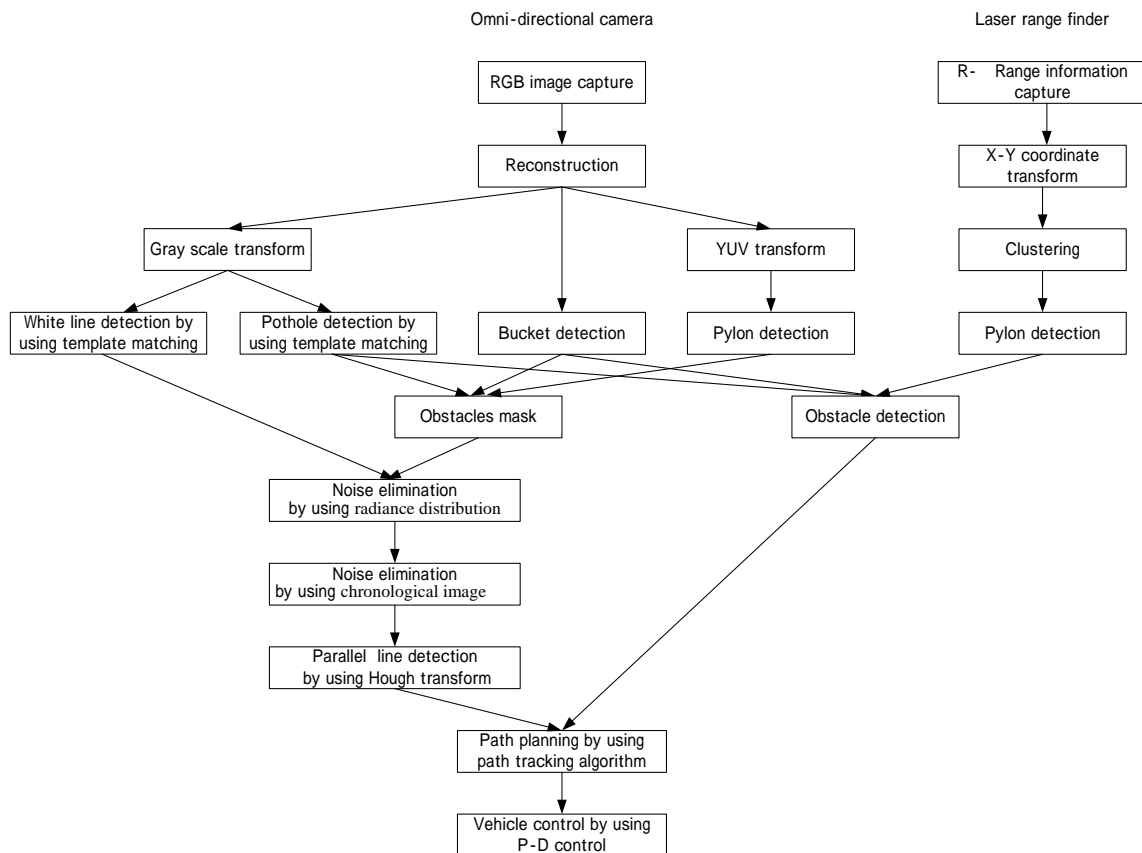


Fig.68 Autonomous traveling algorithm

6.1 実験環境

走行環境は法政大学 B2F ドライエリアであり天候は晴天である。サンプリングタイムは 0.25 秒
 テンプレートマッチングでは 0.6 以上の相関を持つ領域を検出し $\alpha = 0.8$, $\beta = 15$ $\alpha_2 = 0.9$ $\beta_2 = 5$ とした。
 Hough 変換によって 2.5[m],3[m],3.5[m] ,4[m],4.5[m] ,5[m],5.5[m],6[m]の間隔で平行に存在する直線を検出した。



Fig.69 Experiment environment

6.2 結果

白線検出結果を Fig.70(d), Fig.71(d)に示す．地点追従法による車両の経路予測，および目標走行ラインの選択結果を Fig.70(e), Fig.71(e)に示す．赤線が左白線，青線が右白線，緑線が目標走行ライン，青丸がパイロン，赤丸がポットホール，青線の矩形枠は進入禁止領域をそれぞれ示している．

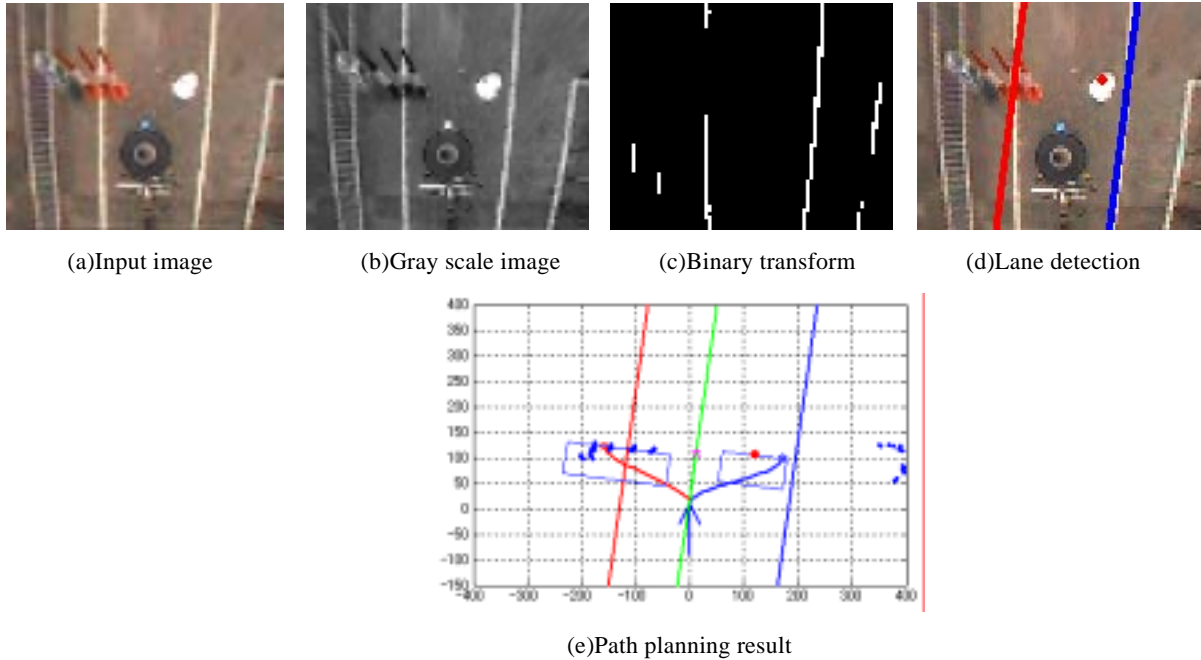


Fig.70 Experiment result in dry area in Hosei University 1

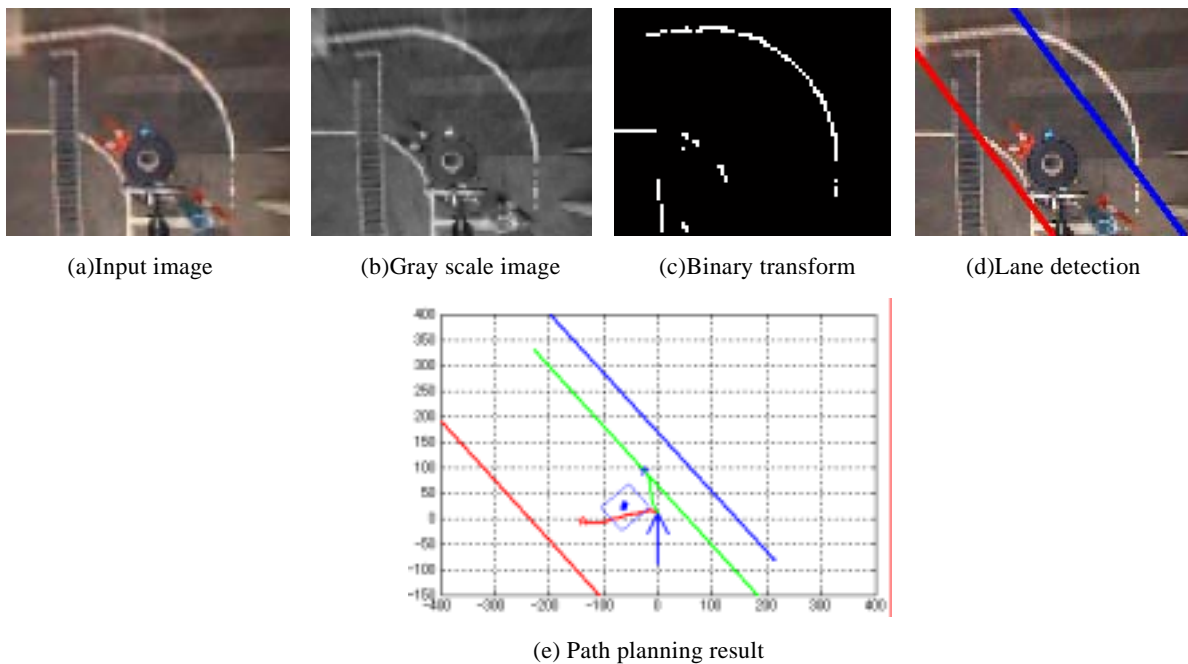


Fig.71 Experiment result in dry area in Hosei University 2

6.3 考察

障害物に衝突することなく，レーン内を正確に走行することができたが，ドライエリアでは排水溝の縁が白線と似たような幅をしており，誤って 2 値化の際に検出してしまうことがあった．しかし，白線が正確に認識できていることや，IGVC 大会中には白線と似たような幅を持つものは存在しないということから，さほど影響はないものと思われる．

7.まとめ

本研究では以下の IGVC における ACC のための自律走行車両の開発について述べた。

- ・テンプレートマッチングをベースとした画像処理
- ・車両の進路予測アルゴリズム・フィードバック制御を用いた車両制御

によって、屋外環境でも安定した走行を実現することができた。

参考文献

- 1) 小森谷 清, 大山 英明 : 光ファイバージャイロを利用した移動ロボットの位置推定の位置手法, 「日本ロボット学会誌」, Vol.14, No.4, pp.532-537 (1996)
- 2) Ken Ishikawa, Kazuyuki Kobayashi, Kajiyo Watanabe : A study of lane detection method for Intelligent Ground Vehicle Competition, SICE in 2003 pp.1406-1409
- 3) 石川謙 : オムニカメラを利用した屋外自律走行車両の開発, 平成 15 年度法政大学大学院システム工学専攻修士論文 (2003)

付録1 テンプレートマッチング

テンプレートマッチングは、事前に登録されている $M \times N$ 画素のテンプレート画像 $T(i,j)$ を探索対象画像 $W(i,j)$ 内で画素単位に逐次移動させ、探索対象画像内におけるテンプレート画像と類似した部分画像を発見する手法である。Fig5 に示すように、探索画像 $W(i,j)$ を $L \times L$ 画素、テンプレート画像 $T(i,j)$ を $M \times N$ 画素とし、両者の間の相関度 $R(x,y)$ を探索空間内にて次式で計算する。ただし、 \bar{W}, \bar{T} はともにそれぞれの画像の平均値である。正規化相関係数 R は二つのパターンの相関が高ければ 1、0 に近づき、低ければ -1、0 に近づく。

$$R(x,y) = \frac{\sum_{i=1}^N \sum_{j=1}^M (W(x+i,y+j) - \bar{W})(T(i,j) - \bar{T})}{\sqrt{\sum_{i=1}^N \sum_{j=1}^M (W(x+i,y+j) - \bar{W})^2} \cdot \sqrt{\sum_{i=1}^N \sum_{j=1}^M (T(i,j) - \bar{T})^2}}$$

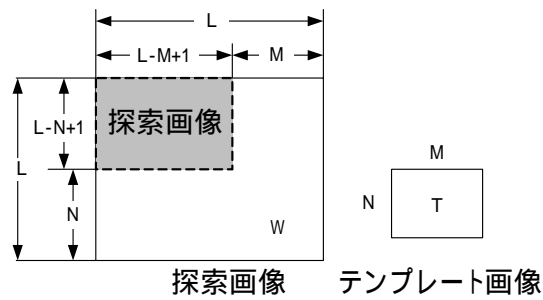


Fig.72 テンプレートマッチング

付録2 ハードウェア仕様

車両の仕様をTable6に示す

Table 6 車両の仕様

型式		MC16P
寸法(全長×全幅×全高)		930mm
重量	バッテリーを含む	83kg
	バッテリーを除く	54kg
駆動モータ(30分定格出力)		DC24V・190W×2
バッテリー		12V35AH(ユアサSEV35)×2
充電器		DC24V6A 電子タイマー付自動充電
前輪		2.50-4-4PR
後輪		2.50-10-2PR
駆動方式		後2輪直接駆動方式
制動方式		電磁ブレーキおよび発電制動・逆転制動
速度制御方式		無段階電子制御方式
操舵制御方式		ステアリングサーボ機構
最高速度	前進	高速6.0 中速4.0 低速2.0 km/h
	後進	高速2.8 中速1.9 低速1.0 km/h
実用登坂角		8°
最小回転半径		0.77 m
段差乗越高さ		50mm
溝乗越幅		100mm
連続走行時間		6時間
連続走行距離		約 33 km
使用者最大体重(積載物含む)		100 kg

レーザレンジファインダの仕様を Table 7 に示す

Table 7 レーザレンジファインダ

角度分解能	0.25 度/0.5 度/1 度 設定可能
外形寸法	155 × 210 × 156mm (W × H × L)
インタフェース	R S 232/R S 422
レーザ保護クラス	1 (本質的に安全)
最大角度	180 度
使用周囲温度	0 度 ~ + 50 度
保護構造	IP65
測定距離(m)	80
分解能(mm)	10
応答時間	53ms/26ms/13ms 角度分解能による
保管周囲温度	-30 度 ~ + 70 度
電源電圧	24V DC ± 15% (最大リップル 500mV)
スイッチング出力	PNP3 点; 代表値 24V DC
システム誤差	代表値 ± 15mm
伝送速度	9.6/19.2/38.4/500 kbps
重量	約 4.5kg
備考	LMS200 は、レーザレーダのように周囲を 2 次元的にスキャンする、非接触レーザ測定システムです。温度範囲 0 度 ~ 50 度で作動し、アクティブ・スキャニング・システムのため、パッシブの機器のようにターゲットを必要としません。

謝辞

本研究を進めるにあたり,適切な教示,指導をしていただいた渡辺嘉二郎教授,小林一行助教授に深く感謝します. 研究室の方々にはとてもお世話になりました. 深く感謝御礼を申し上げます. 特にビークル班のメンバーにはデモ, 大会等でいろいろ迷惑をかけ,大変申し訳なく思っております. 失礼とは存じますが書面にて返させていただきます.

業績書

法政大学大学院工学研究科システム工学専攻修士課程

飯倉 裕樹

2002年(学部4年)

国際会議

・10th Intelligent Ground Vehicle Competition Design Competition

第10回国際自律走行車競技大会デザインコンペティションにて論文提出及び発表 2002年7月

・10th Intelligent Ground Vehicle Competition Autonomous Challenge Competition

第10回国際自律走行車競技大会オートノーマスチャレンジコンペティションに貢献 優勝 2002年7月

2003年(修士1年)

国内会議

・画像処理による障害物・白線検出

第44回知的制御研究会(SIC03-1)にて発表 2003年7月

・Real-time 2D Map Building for an Unmanned Vehicle in a Closed Area

SICE Annual Conference 2003 in Fukui (SICE2003)にて発表 2003年8月

2004年(修士2年)

国際会議

・12th Intelligent Ground Vehicle Autonomous Challenge Competition

第12回国際自律走行車競技大会オートノーマスチャレンジコンペティションにメインプログラマーとして参加
準優勝 2004年6月

国内会議

・Development of Intelligent Wheelchair employing omni-directional camera and slightly tilted laser range
finder

SICE Annual Conference 2004 in Sapporo (SICE2004)にて発表 2004年8月