

# 視覚センサを用いた自律移動ロボットによるライントレース

## Line Tracing By an Autonomous Mobile Robot Using a Vision Sensor

山口智充\*, 及川一美\*, 大久保重範\*, 高橋達也\*

Tomomitsu Yamaguchi\*, Kazumi Oikawa\*, Sigenori Okubo\*, Tatsuya Takahashi\*

\*山形大学

\*Yamagata University

キーワード： 自律移動(autonomous mobile robot), 画像処理(image processing), 視覚センサ(vision sensor),  
ライントレース(line tracing)

連絡先： 〒992-8510 山形県米沢市城南4-3-16 山形大学 工学部 機械システム工学科 大久保研究室  
大久保重範, Tel.: (0238)26-3245, E-mail: sokubo@yz.yamagata-u.ac.jp

### 1. 緒言

視覚センサ(カメラ)を用いたライントレースを行うロボットにおいて, 比較的簡易な制御方法を用いてのトレース手法について述べる. これまで本研究室では, カメラから取り込んだ画像に画像処理を施すことによって, ロボットにボールを追従させたり, 把持させることに成功してきた. そして現在はこれを実用的にするために, 研究室内に貼られたラインを辿って書類や物の運搬を行う研究を行っている. その過程で, 現在は有線で接続されているロボットを自律化のために無線化を行うことが条件となる. そして自律化を行うにあたり, 比較的性能の低いCPUを搭載する予定である. そのため, 処理速度が問題に上がり, 処理の速い制御方法が必要になってくる. 今回は比較的簡易な制御方法の提案とその実行結果について述べる.

### 2. PCおよびカメラの仕様

今回の実験で使用したPC, およびカメラの概略を, Table1, Table2に示す. ホストコンピュータとカメラはUSBにより接続し, カメラより得られた画像はホストコンピュータを用いて画像処理を行う.

Table1 : Host computer

CPU	Duron 750MHz
OS	Linux kernel 2.4.31-Ov11.8
Compiler	gcc version 3.32

Table2 : Camera

Company name	クリエイティブメディア株式会社
Type number	Video BLASTER WEBCAM Plus
Resolution	320 × 240
Focal length	15cm ~

## 2.1 ロボットの仕様

ロボットの画像をFig1に示す．実機には本研究室で作成したロボットを使用する．現在このロボットは有線の移動ロボットであり，2個のステッピングモータを搭載し，車輪を用いて移動を行っている．また速度データはrs232cを介してホストコンピュータから受け取り，左右のステッピングモータの回転速度を決定する．

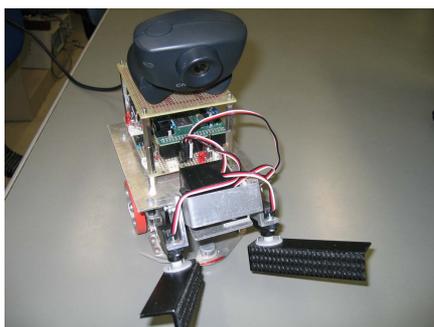


Fig. 1 Robot

## 3. 実験環境

実験環境は研究室の床に貼られた黄色い一本のラインを用いる．このラインは直線から始まり，カーブを描いた後直線に戻る簡易な構想である．このラインをロボットに取り付けたカメラで読取り，ライン上を辿って走行させる．このラインの上部からの外観をFig2に示す．

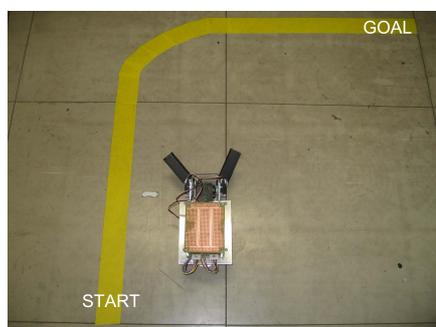


Fig. 2 Experimental environment

## 4. 経路の設定

ここでは比較的処理速度の速い経路の設定方法を提案する．まずライトレースにおいてロボットが進む経路の求め方である．辿るべきラインが直線の場合はラインの中央をロボットの経路に取るのが安定した走行ができると考えられる．問題はラインがカーブのときである．カーブに近似した多項式を用いて経路を表すことも考えられるが，これは多項式の計算のために処理速度が落ちてしまう問題が考えられる．そこで今回は，Fig3のように辿るべきラインの一定間隔に2点を取り，その2点を結ぶ直線を経路として定義する手法を提案する．

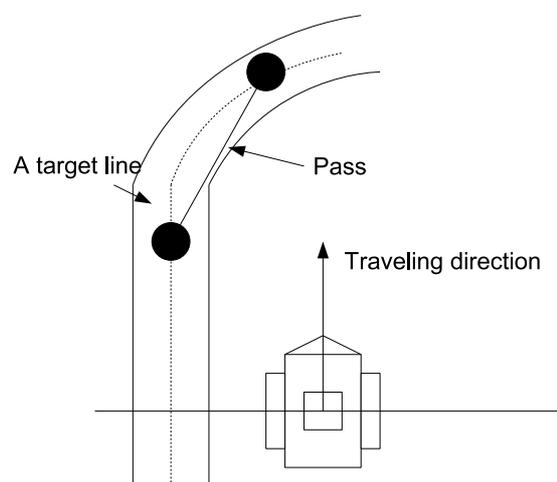


Fig. 3 Setting of path

## 5. 抽出点を用いたライトレース

ここでは上で述べた2点の定義方法，またその抽出方法を述べる．まず一定間隔をどのように定義するかを述べる．本研究ではカメラを用いており，ラインをカメラで見た場合Fig4のように写る．カメラの画像は2次元のものであり，正確な距離を測ることはできないものの，Fig4のように画像に直線を引き，辿るべきラインとの交点を求めることで，一定距離の2点を得ることができる．この2

点を結ぶ直線を経路とすれば，ロボットの位置と姿勢角度とを比較する事でライントレースが可能になると考えた．以降，画像に引いた2直線を抽出線，その2直線と辿るべきラインとの交点を抽出点と呼ぶ．

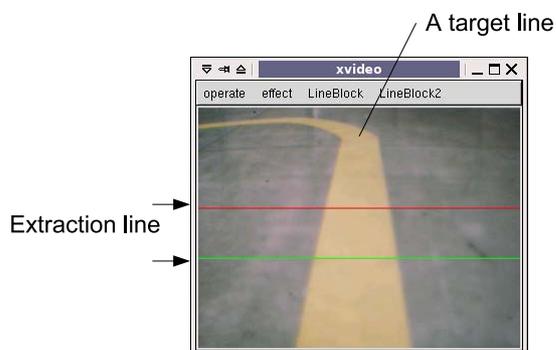


Fig. 4 Extraction line

## 6. 画像処理による抽出点の抽出

抽出点の抽出はカメラから取得した画像に，2値化，ラベリングを行った後に行う．(Fig5) 以下に2値化と本研究で用いているラインブロックラベリング手法を簡単に説明する．

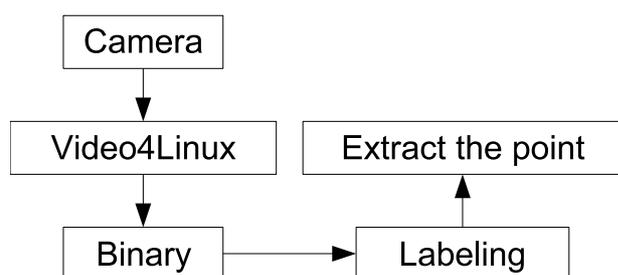


Fig. 5 Flow of image processing

### 6.1 2値化

2値化とは各画素の明るさを一定の基準値により，2つの値(1と0)に変換する処理のことである．また，一定の基準値の事を閾値と呼ぶ．これによ

てFig6のように辿るべきラインの色とそれ以外の色の2つの値に分ける事ができる．



Fig. 6 Making to binary

### 6.2 ラベリング

ラベリング処理とは，2値化された画像内の連結している画素に同じラベル(札)を付与することで複数の領域をグループとして分類する処理のことである．各グループに貼られた「0, 1, 2…」というラベルにより目的とするオブジェクトを識別し，抽出することができる．ラベリングによりノイズを除去することで辿るべきラインのみを扱うことができる．また，本研究ではラインブロックを用いたラベリング手法を用いてラベリングを行う．この手法はFig7のように隣合う画素の塊をブロックとして捉え，上下のブロックの最大値，最小値の位置関係で繋がりを調べる手法であり，通常用いられるラベリング手法(4近傍等)よりも処理が速いために，この手法を用いる．

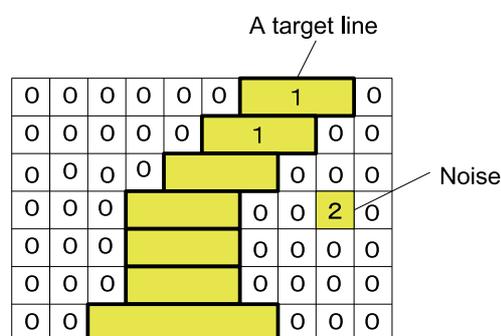


Fig. 7 Lineblock labeling

### 6.3 抽出点

ここでは抽出点の抽出方法について述べる．抽出点はラインの中央に取るためFig8のようになる．しかしこれは辿るべきラインが枝別れしていない事が条件であり，枝分かれのあるラインには対応していない．今回は簡単にするために枝分かれしていないラインを用いる．また抽出線の位置であるが，これは画面の中央付近に設置した．この位置は現在も調整中であるが，抽出点がうまく得られる位置に配置している．これ以降2本の抽出線のうち上側を上抽出線，下側を下抽出線と呼び，上抽出線と辿るべきラインの交点を上抽出点，下抽出線との交点を下抽出点と呼ぶ．

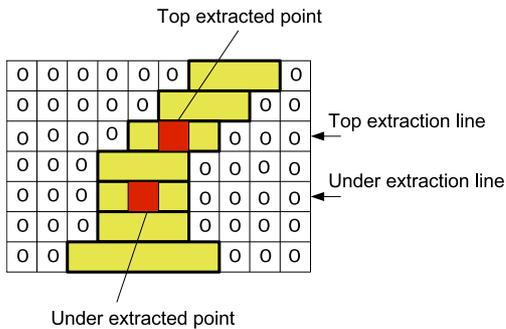


Fig. 8 Pick out the center point

## 7. ロボットの制御

まずロボットの車輪速度であるが，今回は簡単にするため，一定速度の左右旋回，前進のみを用いて行う．

次にロボットの行動決定方法であるが，上抽出点を優先的に画面中央(ロボットの進行方向)に誘導するよう旋回し，その後に下抽出点を中央に誘導することで，ロボットの進行方向とラインの接線が一致するように制御する方法を提案する．これは車を運転する時を想像すると理解しやすい．我々が車を運転するとき，現在の車線から隣の車線へと移るとき，まず前輪を隣の車線へ移動させ，

その後後輪を車線上に載せることで移動することができる．画像の横幅を $w$ としたとき，具体的な制御方法をFig9に示す．

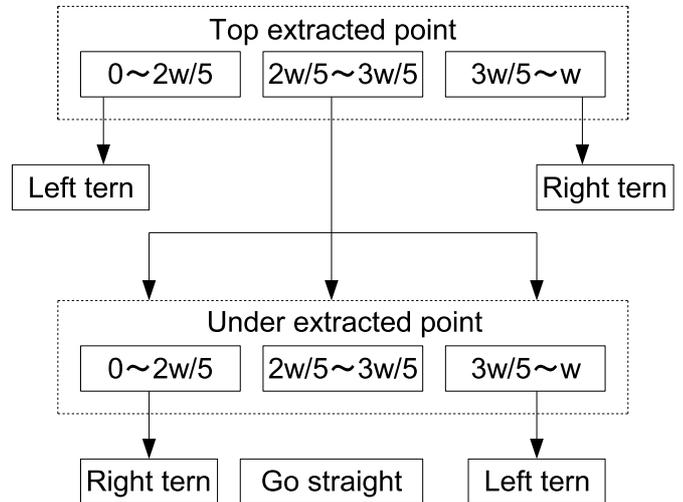


Fig. 9 Control of robot

## 8. 走行結果

2点の抽出点を用いてロボットを制御し，ライントレースの実験を行った．抽出点の抜き出しを行った画像の映像をFig10に示し，実機の走行状況(Fig11)の詳細は当日紹介する．

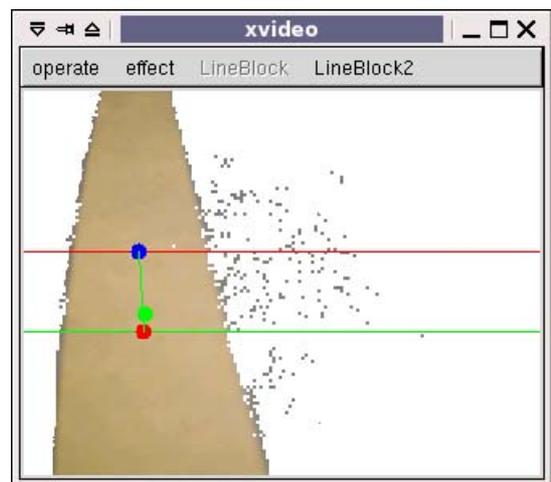


Fig. 10 Result

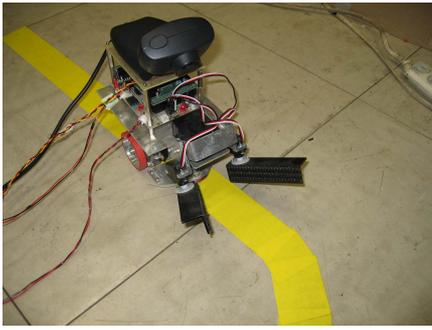


Fig. 11 One scene of experiment

## 参考文献

- 1) 飯尾淳:Linuxによる画像処理プログラミング (2000)
- 2) 藏前智映:博士前期課程修士論文 視覚センサを用いた自律移動ロボットの開発(2006)

## 9. 考察

抽出点の抽出の問題点としては、2点の抽出点の位置が安定して得られないという事がある。この原因は現在調査中であるが、一つの解決策として、抽出線にある程度の幅を持たせ、その重心位置を求めることによって、安定させる事ができるのではないかと考えている。

また実機の問題点としては、カーブにおいて滑らかにラインに追従できていないという事がある。この問題の大きな原因の一つは、速度を3パターン(一定の左右旋回+前進)しか用意していない事である。これによって進行方向の補正の際に動きが離散的になってしまっているのではないかと考えている。この解決策としては回転速度のパターンを増やし、ロボットを経路へ細かに修正することで解決を図る。

## 10. 結言

今回は2点の抽出点を用いたラインレースの実験を行った。今後は安定したカーブラインのトレースと、ロボットの無線化を行っていきたいと考えている。