

# インタラクティブなラインレースロボットシステムの開発： ロボット座標の推定とフィードバック

## Development of an interactive line following robot system: localization and feedback of the robot position

○藤原大樹, 岩谷靖

○ Taiki Fujiwara, Yasushi Iwatani

弘前大学

Hirosaki University

キーワード： ライントレースロボット (line following robot), ジェスチャー認識 (gesture recognition), インタラクティブデスクトップ (interactive desktop), プロジェクタ・カメラシステム (projector-camera system), ビジョン (vision)

連絡先： 〒 036-8561 青森県弘前市文京町 3, 弘前大学 大学院理工学研究科  
岩谷靖, Tel.: (0172)39-3697, Fax.: (0172)39-3697, E-mail: iwatani.at.cc.hirosaki-u.ac.jp

### 1. はじめに

ラインレースロボットは, 光センサなどによって地面に描かれた線を読み取り, その線に沿って移動を行う自律移動ロボットである. このロボットは, 比較的簡便に製作できる自律移動ロボットであり, かつ, センサによる情報取得・アクチュエータによる運動生成・プログラムによる移動計画などのロボット技術に欠かさない要素技術を学べるため, 初学者に対するロボット工学教育に適している.

本研究では, 小中高校生などの興味を機械工学へ誘うための試みとして, インタラクティブに操作可能なロボットシステムを構築する. 具体的には, ライントレースロボットとデジタルデスク, ジェスチャー認識インタフェースを組み合わせたシステムを構築する. デジタルデス

クとはカメラ, プロジェクタをコンピュータと接続したシステムであり, デスク上でのユーザの手の動きがコンピュータへの入力, プロジェクタからデスクへの投影が出力となる<sup>1)</sup>. デジタルデスクを用いると, カメラで読み取った手の形状認識によってデスク上に線を描画し, 自分の描いた線に沿ってラインレースロボットを走行させることができる<sup>2)</sup>.

筆者らはこれまで, ライントレースロボットとデジタルデスク, マーカレスジェスチャーインタフェースを組み合わせることで, より直感的にロボットを制御可能なシステムを構築してきた<sup>3)</sup>. 描画した線の上をロボットに走行させるときに, 線がすべて残っていても走行できる領域がいづれなくなってしまう. そのため, 文献<sup>3)</sup>のシステムでは, 設定した時間以前に描画した線が消えるようにしていた. しかし, それ

では指定したコースの形状によってロボットの走行速度が変化した際に、「線の消える速度がロボットの走行速度を上回り、ロボットが線を見失う」ことや、「ロボットの走行速度が線の消える速度を上回り、新しい線と古い線が交差してしまいロボットが行き止る」ことがあった。本研究では、文献<sup>3)</sup>で構築したシステムにロボット位置の検出機能と検出位置のプロジェクタ映像へのフィードバック機能を付与することでこれらの問題を解決し、恒常的にロボット操作が可能なシステムを構築する。

## 2. システムの概要

Fig. 1 に、作成したデジタルデスク環境を示す。デスク面には、白色半透明のアクリル板をスクリーンとして設置する。デスクの上方に斜め下向きのカメラを取り付け、ユーザの手の動きを取得する。デスク横には三脚を用いてプロジェクタを設置し、デスク下に設置した鏡を介してスクリーン下から投影を行う。デスク面の手前の端から奥行き 0.36 m のスペースは入力スペースとして利用する。他の仕様は表 1 に記載した。

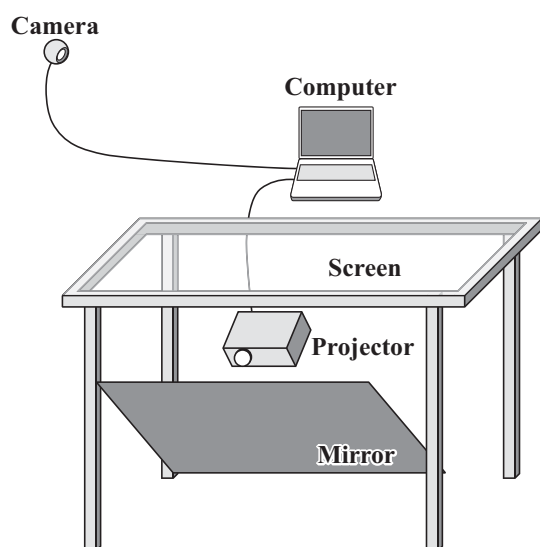


Fig. 1 System configuration

Table 1 Specifications of our system

Desk	1.120 m × 1.380 m × 0.700 m
Screen	White acrylic board with 3 mm thick
Mirror	1.100 m × 0.800 m
Camera	Logicool, Webcam C910
Projector	CASIO, XJ-S57
Robot	Lego Mindstorms NXT
PC	OS: Windows7 32bit CPU: Intel Core i3 (2.13 GHz) Memory: 4.00 GB RAM

## 3. ロボットの位置検出

プロジェクタの投影面をカメラ画像から読み取ることで、ロボットの位置検出を行う。まず、射影変換を用いて、取得画像をスクリーン真上から鉛直下向きに見た画像へと変換する。つぎに、射影変換画像からスクリーンの投影面のみを抽出する。さいごに、抽出画像をグレースケール化の後、二値化し、その重心を得ることでロボット座標を得ることができる。このような簡単な手順でロボット座標が取得できる理由は、以下の通りである。

- 1) 投影面上にはロボットしか存在しない。
- 2) 線の描画されていない箇所は白色光が投影されており、また、スクリーンは白色半透明なので、線の描画されていない箇所は白色に極めて近い輝度値を持つ。
- 3) ロボットのタイヤは黒色である（本稿で使用した Lego Mindstorms NXT のみならず、大抵のライントレースロボットに該当することが期待される）。
- 4) 投影線はスクリーン上にライトを照射しないことによって生じる受動的な黒線であるため、タイヤと比較して輝度値が白色に近い値である。

上記の 2)~4) の事実を示すために、Fig. 2 の左段に白色光投影面のみを撮影した画像、投影

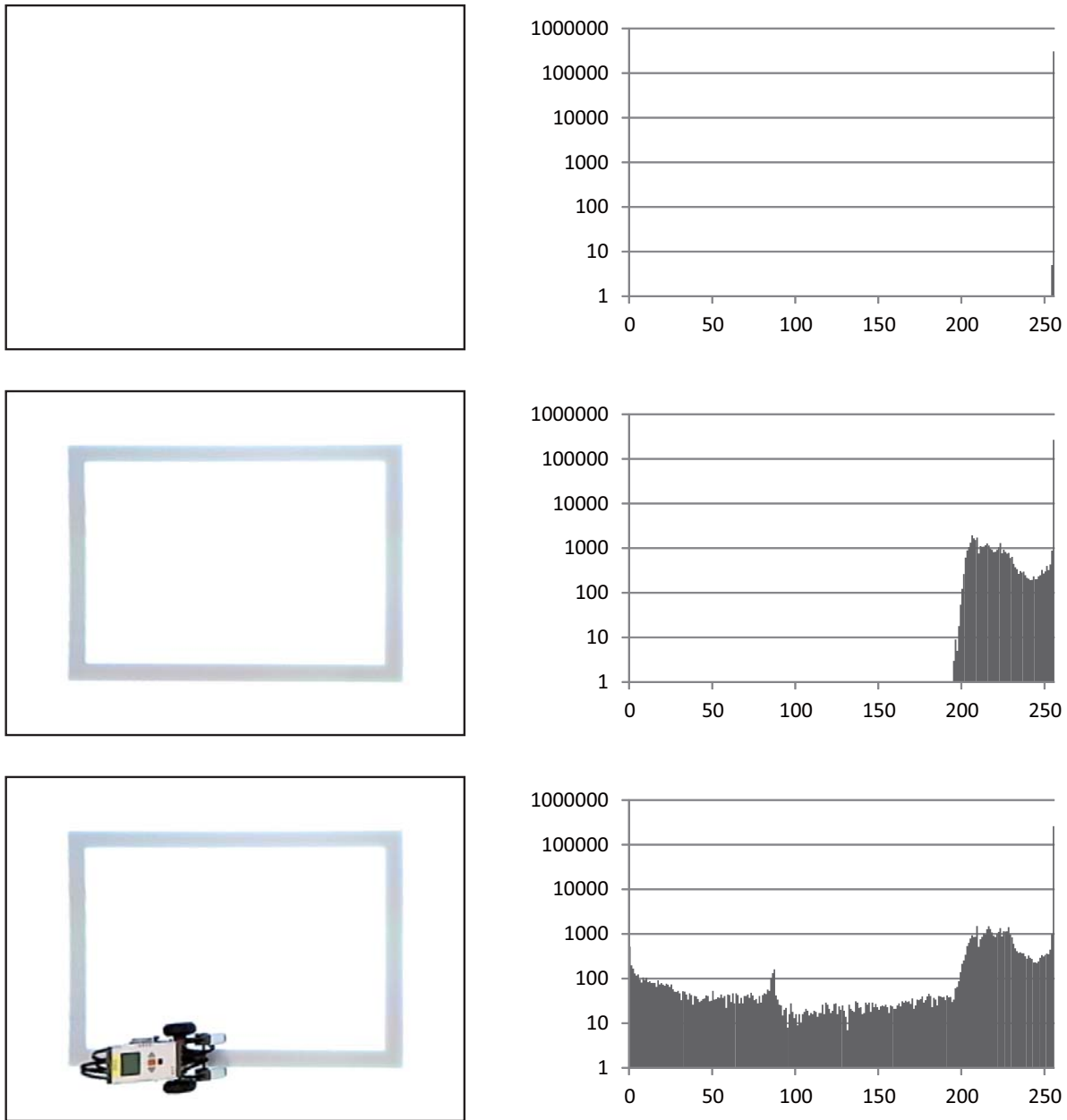


Fig. 2 Histograms of some snapshots of the projected field.

面に線を描画した画像，投影面に線を描画しロボットを置いた画像を，また，右段にその輝度値のヒストグラムを示す．Fig. 2 右段のヒストグラムは左端が輝度値 0（黒色），右端が輝度値 255（白色）を表す．Fig. 2 上図から，線が描画されていない状態では白色成分のみが観測されている．つぎに，Fig. 2 の上図と中図を比較すると，中図の輝度値 150 付近から 250 付近の位置に上図には無い分布が出ている．二枚の元画像の違いは線の描画の有無だけなので，これは線を表している．さいごに，Fig. 2 の中図と下図を比較すると，下図の輝度値 0 から 150 付近の位置に中図には無い分布が出ている．二枚の元画像の違いはロボットの有無だけなので，これはロボットを表している．以上の比較から上記 2)～4) の事実，すなわち，スクリーン・線・ロボットはそれぞれ，白色成分・白色寄りの灰色成分・黒色成分を多く含むことがわかる．したがって，二値化によって黒色成分を抽出し，その重心を求めることでロボット座標が求まる．

#### 4. 線の自動消去

ロボットが恒常的に指先を追跡できるようにするために，前章で得られたロボットの位置を利用して線を自動的に消去する機能を述べる．本システムで線を描画する仕組みは，描画モードにおける各時間ステップの指先座標を記憶し，ある点とそのひとつ前の点を線で繋ぐことを線の始点から終点までの間繰り返すものである<sup>3)</sup>．

いま，投影線が  $n$  個の点から構成されるものとし，各点を  $(X_i, Y_i)$  と書く．一般性を失うことなく，投影線の始点を  $(X_1, Y_1)$ ，終点を  $(X_n, Y_n)$  とする．また，前章の方法で取得されたロボット座標を  $(X_r, Y_r)$  と書く．なお，線の自動消去機能は，ロボットが投影面内に存在しているときのみ行う．

まず，ロボットが投影線に存在するか否かを判断する．これは，ロボット座標と投影線の構

成点との距離が与えられた閾値  $Th_1$  以下になるような構成点  $(X_i, Y_i)$  が存在するか否かを確認することで判断できる．すなわち，

$$\sqrt{(X_i - X_r)^2 + (Y_i - Y_r)^2} < Th_1 \quad (1)$$

を満たす点  $(X_i, Y_i)$  が存在する場合，ロボットが投影線上に存在するとみなす．

つぎに，(1) を満たす点が存在する場合に，投影線を自動消去する手法について述べる．いま，始点  $(X_1, Y_1)$  から昇順に，(1) を満たす最初の点を  $(X_m, Y_m)$  とする．ただし， $1 \leq m \leq n$  である．点  $(X_m, Y_m)$  から降順に，

$$\sqrt{(X_i - X_r)^2 + (Y_i - Y_r)^2} > Th_2, \quad i < m \quad (2)$$

を満たす最初の点を  $(X_\ell, Y_\ell)$  とする．ただし， $Th_2$  は与えられた閾値である．このとき，始点  $(X_1, Y_1)$  から点  $(X_\ell, Y_\ell)$  までをロボットが通過したものとみなし，自動消去する．始点  $(X_1, Y_1)$  が (2) を満たさない場合でも，自動消去の対象となることに注意されたい．円弧状の投影線に対しても適切に自動消去を行うことができる．

#### 5. 実験結果

ロボットが通過した後の線が消える設定とし，ロボットに指先を追いかせさせた．動作の様子を Fig.3 に示す．従来システムで問題となっていた「線の消える速度がロボットの走行速度を上回り，ロボットが線を見失う」ことや，「ロボットの走行速度が線の消える速度を上回り，新しい線と古い線が交差してしまいロボットが行き止る」ことがなくなり，恒常的なロボット操作が可能となった．

#### 6. おわりに

ユーザがインタラクティブにロボットと接することのできるシステムとして，デジタルデ

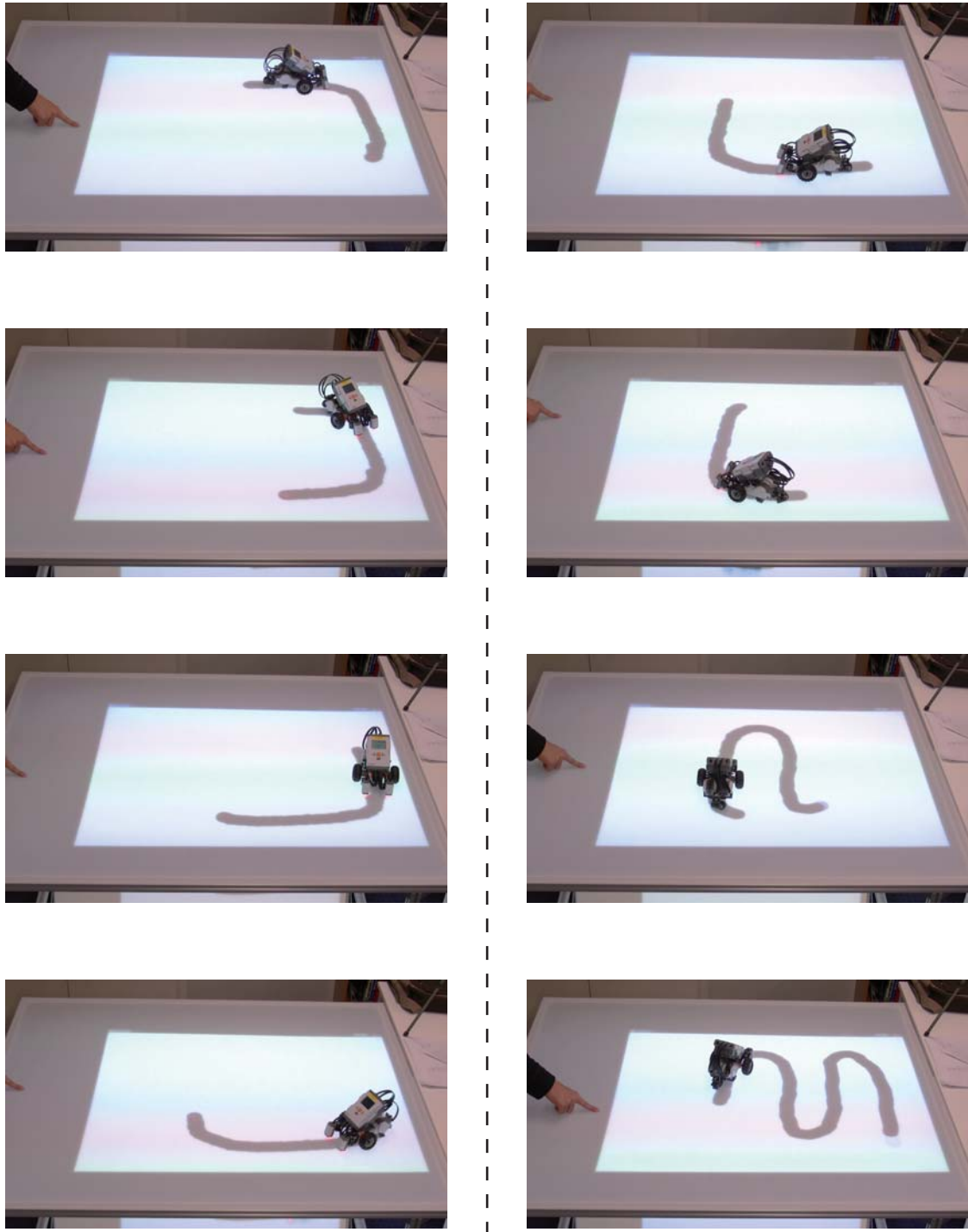


Fig. 3 Snapshots taken every two seconds. They are ordered from top to bottom then left to right.

スク型システムによるロボット操作機構を開発した。とくに，ライントレースロボットの位置を推定し，ロボットが通過した線を消去することで，恒常的にライントレースが可能となる機能を付与した。

## 参考文献

- 1) P. Wellner: Interacting with paper on the DigitalDesk, Commun. ACM, Vol. 36, No.7, pp.87-96 (1993)
- 2) 榎・戸潤・森田・鏡・橋本：人・ロボット・コンピュータのインタラクションのためのデジタルデスク環境の構築, 計測自動制御学会東北支部 45 周年記念学術講演会 (2009)
- 3) 藤原・岩谷：ジェスチャ認識で動作するインタラクティブライントレースロボットシステムの開発, 第 55 回システム制御情報学会研究発表講演会, pp. 153-154 (2011)