

2次元画像からの指先位置検出に基づく 実時間手形状認識と 移動ロボット遠隔操縦タスクへの応用

丸谷 誠慶 西川 敦 宮崎 文夫

大阪大学大学院 基礎工学研究科 システム人間系専攻 機械科学分野

本研究では、1台のカメラから得られる2次元画像を用いて実時間で手形状を認識する手法を提案し、移動ロボットの遠隔操縦を行うインタフェースとして利用することを考える。まずカメラから得られる2次元画像から、色情報を用いて手領域を抽出する。次に得られた手領域の輪郭線を求め、その曲率を用いて指先位置を検出する。さらに検出された指先の数や手領域の重心との相対的な位置関係から、手形状を認識する。この手法により認識できる数種類の手形状を用いて、移動ロボットの遠隔操縦を行う。ロボットの操縦は、直進、旋回などの行動の決定だけでなく、速度の調節を行うことでよりの確な動作を実現する。

Real-time Hand Pose Recognition based on Fingertip Detection from 2-dimensional Image and Its Application to Remote Control of a Mobile Robot

Masachika Maruya, Atsushi Nishikawa and Fumio Miyazaki

Division of Mechanical Science, Department of Systems and Human Science
Graduate School of Engineering Science, Osaka University

This paper describes a method for recognizing hand pose in real-time using a 2-dimensional image from a CCD camera. First, hand area is extracted using the color data of a 2-dimensional image. Secondly, the fingertip position is detected by obtaining the contour of the hand area. The hand pose is then recognized using the number of the detected fingertips and the position of the center of gravity of hand area. Finally, remote control of a mobile robot is carried out with some kinds of hand poses which can be recognized by the proposed method.

1 はじめに

人のジェスチャを認識する研究は数多く行われているが、中でも画像を用いた非接触型の手法は、データグローブなどのようなセンサを装着する必要がなく、人に拘束を与えないため、より自然なインタフェースとして注目されている。西川ら [1] は、オプティカルフローを用いて手の移動方向の変化率を元に、あらかじめ用意された基本動作の列として符

号化を行うことで手ぶりの認識を行う手法を提案している。また、上田ら [2] は、多視点画像を用いて、2種類の3次元手形状モデル(関節モデル、Voxelモデル)に基づく任意の手形状推定を行っている。このように、ジェスチャとして手の動きを用いたものと、手の形状を用いたものに分けられる。手の形状を用いた場合、ある物体の形状や数などを容易に表現することができ、動きだけでは表現することが難

しい複雑な指示を行うことができると考えられる。

画像を用いた手形状認識において、任意の手形状を認識する場合、手の3次元モデルなどが用いられる[2]。あらかじめ3次元モデルで可能な形状を生成しておき、画像から得られた手形状に最も近いものを探索する手法である。この手法は任意の手形状を推定できるため形状認識に適しているが、計算量が多く実時間での処理は難しい。しかし限られた形状パターンだけを認識する場合、複雑な3次元手形状モデルによる形状推定を行わずに、より単純な手法を用いた実時間での処理も可能であると考えられる。

そこで本研究では、画像を用いて実時間で手形状を認識する手法を提案し、移動ロボットの遠隔操縦を行うインターフェースとして利用することを考える。人が移動ロボットを操縦する場合、「直進」、「旋回」、「停止」といった数種類の指示だけでよく、任意の手形状推定は必要ないと考えられる。そこで、1台のカメラから得られる2次元画像の特徴を利用し、数種類の形状パターンを認識することによりロボットを操縦することを考える。

ところで、人が自動車を運転する場合、高速道路のように広く、直線的な道ではスピードを出し、駐車場の中など狭い場所ではゆっくりと移動する。このように、自動車の運転において「速度」の調節は不可欠である。同様に、移動ロボットの操縦でも「速度」は重要だと考えられる。長い距離を直線的に移動する場合は速く、正確な移動が必要な場合はゆっくりと、というように周囲の状況に応じて「速度」を調節することで、よりの確な動作を実現できると考えられる。そのため、「直進」、「旋回」、「停止」といった指示に加え、「速度」の調節を行うことも考える。

2次元画像から取得できる特徴として、手領域の面積、重心位置、慣性主軸、輪郭線などがあげられるが、本研究では特に輪郭線に注目する。輪郭線は、指先の部分で大きく曲がっているため、この曲がり具合(曲率)を用いて指先位置を検出する。さらに、指先の数や手領域の重心座標と指先位置の相対的な関係を用いて手形状を認識する手法を提案する。相対的な特徴を用いることにより、画像中の手領域の大きさや向きによらず、安定して形状を認識できると考えられる。

2 手形状認識手法

2.1 画像処理による手領域の抽出

手領域を抽出するための画像処理の手順を以下に示す。

1. 画像データ変換
RGBデータとして得られるカメラ画像をHSVデータに変換する。
2. 2値化
得られたHSVデータと設定した閾値を比較し、肌色領域は1、それ以外の領域は0で2値化する。
3. 収縮
画素値が0である画素に隣接している周囲の4つの画素値を0にする。ここで、周囲4つの画素とは注目している画素の上下左右の画素のことである。これにより、小さいノイズが除去される。
4. 膨張
画素値が1である画素に隣接している周囲の4つの画素値を1にする。収縮と同じ回数だけ行う。
5. ラベリング
画素値が1である画素について、隣接しているすべての画素に同一のラベルを付ける。
6. 手領域抽出
それぞれのラベルの面積を求め、最大の面積を持つ領域を手領域と考える。また、手領域の重心座標ならびに輪郭線を求める。輪郭線追跡には文献[3]で示されている手法を用いた。

2.2 曲率を用いた指先検出

前節の手法で得られた手領域の輪郭線上のすべての点について曲率を求め、曲率の大きい点が集中している部分を指先と考える。

図1は輪郭線上のある一点 $P_i(x_i, y_i)$ を拡大したものである。点 $P_{i-n}(x_{i-n}, y_{i-n})$ 、 $P_{i+n}(x_{i+n}, y_{i+n})$ は点 P_i から n 点離れた輪郭線上の点である。また

$\theta_{i-n}^i, \theta_i^{i+n}$ はそれぞれベクトル $\overrightarrow{P_{i-n}P_i}, \overrightarrow{P_iP_{i+n}}$ と X 軸とがなす角度である .

$\theta_{i-n}^i, \theta_i^{i+n}$ はそれぞれ

$$\theta_{i-n}^i = \tan^{-1} \frac{y_i - y_{i-n}}{x_i - x_{i-n}}$$

$$\theta_i^{i+n} = \tan^{-1} \frac{y_{i+n} - y_i}{x_{i+n} - x_i}$$

と表される . また, $\theta_{i-n}^i, \theta_i^{i+n}$ を用いて θ_i を表すと,

$$\theta_i = \theta_i^{i+n} - \theta_{i-n}^i$$

この θ_i を, 点 P_i における曲率と定義する [4] .

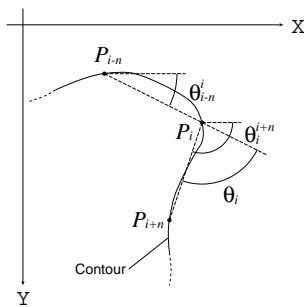


図 1: 輪郭線上の点 P_i での曲率 θ_i

本研究では, 輪郭線を反時計回りに追跡しているため, 指先での曲率は常に負の値となる . そこで, まず輪郭線上から, 曲率が負でありその絶対値が大きい点 (高曲率点) を抽出する (図 2(b)) . 次に抽出された高曲率点に対して, 隣接したすべての点に同じラベルを付け, k 点以上連続している点列のみを抽出する (図 2(c)) . 最後に, 連続した点列の X 座標, Y 座標それぞれ平均値を求めることにより, 点列の中心座標を求め, これを指先位置座標とする (図 2(d)) .

2.3 指先位置を用いた手形状認識

図 3 のように, 手領域の重心と指先位置座標から特徴量 $\phi_1, \phi_2, \dots, \phi_{fing}$ ($fing$ は指先として検出された部分の数) を求める . この ϕ_i と指先の数を用いて, 手形状認識を行う . ここで扱う手形状として, 指は伸ばしている状態と曲げている状態の 2 種類とする .

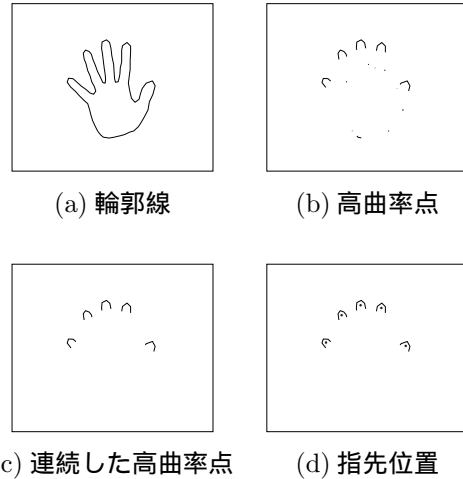


図 2: 指先位置検出

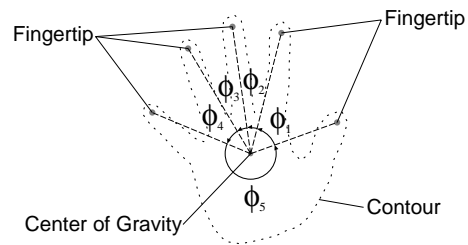


図 3: 手形状認識に用いる特徴量

1. 手領域の重心と指先位置座標を用いて, 特徴量を求める . 重心を $O(x_0, y_0)$, 指先位置座標を $F_i(x_i, y_i)$ とし, 次式により ϕ_i を求める .

$$\phi_i = \begin{cases} \tan^{-1} \frac{y_i - y_0}{x_i - x_0} - \tan^{-1} \frac{y_{i+1} - y_0}{x_{i+1} - x_0} & (1 \leq i \leq fing - 1) \\ \tan^{-1} \frac{y_i - y_0}{x_i - x_0} - \tan^{-1} \frac{y_1 - y_0}{x_1 - x_0} & (i = fing) \end{cases}$$

(ただし, $0 \leq \phi_i < 2\phi$)

2. ϕ_i のうち最も大きいものを ϕ_{fing} とし, 反時計回りに $\phi_1, \phi_2, \dots, \phi_{fing-1}$ の順で添字を付け直す . これにより, 同一の形状であれば, カメラ画像中での手の角度によらず常に決まった添字を用いることができる .
3. あらかじめ, 認識に用いる $form$ という手形状について, 指先の数 num とそれぞれの指の

間の角度の閾値を設定しておく。ここで、この形状の j 番目 ($1 \leq j < num$) の特徴量 ϕ_j の上限値を ϕ_{j_upper} , 下限値を ϕ_{j_lower} と表す。

4. 検出された指先の数 $fing$ が $form$ の指先の数 num に等しいか調べる。等しい場合は 5. へ移動する。異なる場合は、未知の手形状として出力し終了する。
5. 得られた ϕ_i が、すべての $i \in 1, 2, \dots, num$ について

$$\phi_{i_lower} < \phi_i < \phi_{i_upper}$$

を満たすならば、形状 $form$ として出力する。そうでなければ未知の手形状として出力する。

2.4 認識実験

前節で述べた認識手法を用いて、認識実験を行った。表 1、図 4 に示す手形状について、それぞれ 300 回ずつ連続して画像を取り込み、認識率を求めた (表 2)。

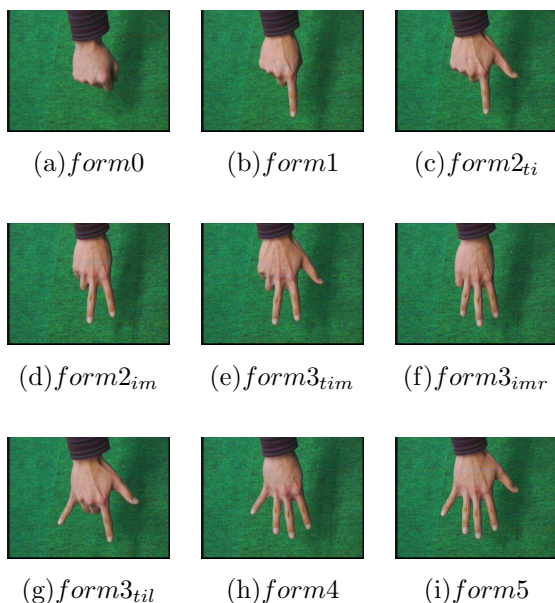


図 4: 実験で用いる手形状

表 2 を見ると、認識率がほぼ 100% に近いことがわかる。被験者 C の $form4$ の認識率が 54.0% とかなり低いが、これは画像処理の時点で手領域の抽出

表 1: 実験で用いる手形状

指先の数	形状名	手の形状
0	$form0$	手を握っている
1	$form1$	人差し指を立てている
2	$form2_{ti}$	親指, 人差し指を立てている
	$form2_{im}$	人差し指, 中指を立てている
3	$form3_{tim}$	親指, 人差し指, 中指を立てている
	$form3_{imr}$	人差し指, 中指, 薬指を立てている
	$form3_{til}$	親指, 人差し指, 小指を立てている
4	$form4$	親指以外を立てている
5	$form5$	手を開いている

表 2: 認識率 (単位 [%])

手形状	被験者 A	被験者 B	被験者 C
$form0$	100.0	99.0	100.0
$form1$	100.0	100.0	100.0
$form2_{ti}$	100.0	100.0	100.0
$form2_{im}$	100.0	100.0	100.0
$form3_{tim}$	100.0	100.0	100.0
$form3_{imr}$	100.0	99.3	95.7
$form3_{til}$	100.0	98.0	100.0
$form4$	99.0	97.7	54.0
$form5$	100.0	100.0	98.3

に失敗し、指先の数を正確に検出できなかったためだと考えられる。原因としては、照明条件により肌色の閾値が変化し、手の一部であるにも関わらず肌色として検出されないことが挙げられる。また、光の反射によって手の一部が白く写ってしまう場合にも、抽出に失敗することが多かった。

3 移動ロボットの遠隔操縦

これまでに述べてきた手形状認識手法を用いて、移動ロボットの遠隔操縦を行い、この手法の実用性について検証する。なお、この実験では片手ではなく両手を用いる。2.1 節で述べた手領域抽出手法を拡張し、最大面積を持つ肌色領域とその次におおきな面積を持つ肌色領域を抽出することで両手をとらえ、さらにそれぞれの手領域について独立に形状認識を行う。主に左手の形状でロボットの行動を決定

し、右手の指の間隔で車輪の回転速度を調節する。また、指示者は三脚に固定されたカメラの前に座り、直接ロボットを見ながら指示を与える。

実験に用いたシステムを図5に、カメラ画像と両手抽出結果の一例を図6に、実験の様子を図7に示す。

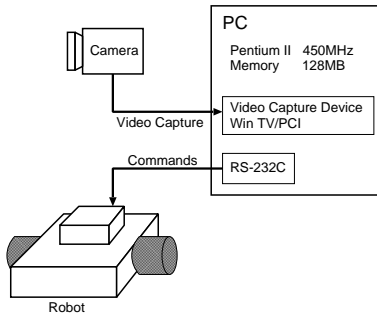
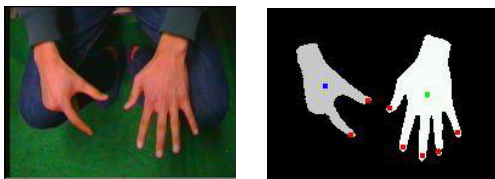
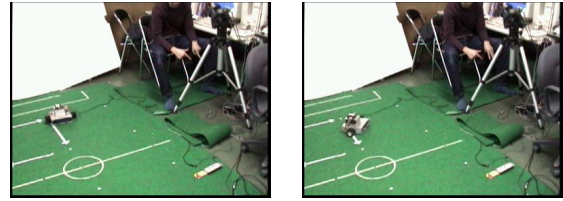


図 5: システム構成

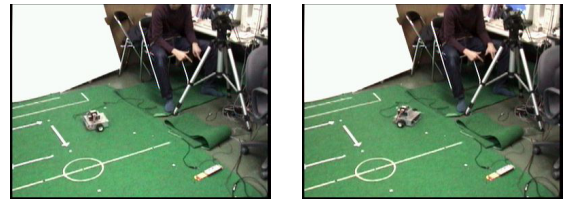


(a) カメラ画像 (b) 両手抽出結果

図 6: 両手の抽出



(1) (2)



(3) (4)

図 7: 実験の様子

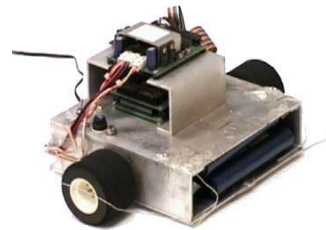


図 8: 移動ロボット

3.1 移動ロボット

実験で使用する移動ロボット(図8)は、NEC製のCPU、V25を使用した日本システムデザイン社製のCPUボードと制御用OS、JSDOSを搭載している[5]。大きさは約270mm×180mm×150mmであり、独立したモータによって駆動される2つの車輪によって移動する。また、外界との通信が可能なシリアルポートを持っており、外部のPC等からプログラムのダウンロードや、実行中のプログラムの通信用の用途として利用可能である。ロボットシステム全体は7.2VのNi-Cdバッテリーで機能する。

移動ロボットは、PCから送信される左の車輪の回転速度 $send_l$ と、右の車輪の回転速度 $send_r$ をシリアルポートを経由して受け取る。 $send_l, send_r$ は

整数で、

$$-100 \leq send_l, send_r \leq 100$$

の範囲で与える。 $send_l, send_r$ が0のときモータは停止し、正のときは前進方向、負のときは後退方向に回転する。また、 $send_l, send_r$ の絶対値が大きいくほど、車輪の回転速度が速くなる。

3.2 指示方法

3.2.1 手の形状によるロボットの速度調節

人は大きさや量を表現する場合、親指と人差指の間隔を用いることがあるが、この表現を利用してロボットの速度を調節することを考える。

親指と人差指の間隔の情報を得るために、特徴量 ϕ_{ti} を用いる。 ϕ_{ti} とは、親指(thumb)と人差指

(index finger) の間の角度を表す．ここで，親指の指先と人差指の指先間の距離を用いていないのは，手とカメラの間の距離が異なる場合，画像中での手領域の大きさが変化し指先間の距離も変わってしまうためである．角度を用いた場合，手領域の大きさの変化によらず安定して情報をとらえることができると考えられるため，これを採用した．

特徴量 ϕ_{ti} からロボットへ送信する回転速度 rot への変換を次に示す．

$$rot = \begin{cases} 0 & \phi_{ti} \leq 0.3 \\ \frac{(\phi_{ti} - 0.3)}{1.5 - 0.3} \times 100 & 0.3 < \phi_{ti} < 1.5 \\ 100 & \phi_{ti} \geq 1.5 \end{cases}$$

この式で用いている ϕ_{ti} の単位は [rad] である．また，得られた rot の値が整数でない場合は，小数点以下を切り捨て整数として扱う．

左手，右手どちらの場合も，指先の数が 2 のとき rot の値を計算する．なお，左手から得られる値を rot_l ，右手から得られる値を rot_r とする．また， rot_l, rot_r は整数で，

$$0 \leq rot_l, rot_r \leq 100$$

を満たす．

3.2.2 左手の形状によるロボットの行動決定

左手の形状と，そのとき PC から送信される $send_l, send_r$ の値について表 3 に示す．

表 3: 左手の形状に対する $send_l, send_r$ の値

左手の形状	$send_l$ の値	$send_r$ の値
<i>form1</i>	rot_r	rot_r
<i>form5</i>	$-rot_r$	$-rot_r$
<i>form3_{tim}</i>	rot_r	$-rot_r$
<i>form3_{imr}</i>	$-rot_r$	rot_r
<i>form2_{ti}</i>	rot_l	rot_r
上記以外	0	0

表 3 のように，左手の形状が *form1* の場合はまっすぐ前進し，*form5* の場合はまっすぐ後退する．また，*form3_{tim}, form3_{imr}* の場合はそれぞれ右，左にその場で回転する．さらに，左手の指先の数が 2 の場合，左手の ϕ_{ti} を用いて左の車輪の速度を調節し，右手の ϕ_{ti} を用いて右の車輪の速度を調節する．

そのため，なめらかな曲線を描くように移動することも可能である．一方，左手の形状が表 3 に示す形状以外の場合，あるいは右手の指先の数が 2 以外の場合は， $send_l = send_r = 0$ とし，常に停止する．

3.3 実験方法

図 9 に示すようなコースを，ロボットが移動できるフィールド上に白いラインで描き，そのラインに沿ってロボットを遠隔操縦するタスクを行う．

- コース 1(図 9(a))
S からスタートし，600mm 前進した後 8 の字を描くように移動，さらに 600mm 前進して G で停止．
- コース 2(図 9(b))
S からスタートし，1 辺 600mm の正方形を描くように右回りに移動，その後同様に左回りに移動して G で停止．
- コース 3(図 9(c))
S からスタート，600mm 前進した後右に 90° 回転し，600mm 後退して G で停止．

被験者は，フィールド上の移動ロボットを直接観測し，S,1,2,...,G の順にロボットの遠隔操縦を行う．また，ロボットの操縦の様子を，SONY 製デジタルビデオカメラ DCR-TRV20 で撮影しておき，QuickMAG System III[6] の 2 次元カラーシステム¹ を用いて，ロボットの上部に取り付けられた 2 つのマーカを追跡し，移動軌跡を記録する．マーカの配置を図 10 に示す．

3.4 結果と考察

3 名の被験者 A, B, C に対して，ロボットの移動軌跡とロボットが移動を開始してから停止するまでの所要時間を計測した．被験者 A の結果を図 11

¹使用した QuickMAG SystemIII のシステム構成では，2 つの CCD カメラを用いて 60fps での 3 次元のリアルタイム位置計測を行う能力をもつ．最高 16 の対象を色情報を用いて追跡することが可能である．今回利用したのは，2 チャンネルある映像入力の方だけを用いた平面運動の解析機能である．任意のビデオカメラについて DLT 法を用いたキャリブレーションを行って解析することが可能である．

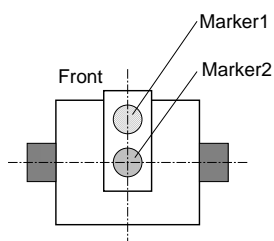


図 10: マーカの配置

に、被験者 B の結果を図 12 に、被験者 C の結果を図 13 に示す。

図中の実線はマーカ 1 の軌跡、円はマーカ 2 の軌跡、破線はコースである。マーカの軌跡は毎秒 10 回記録した。

図 11(a)、図 12(a)、図 13(a) を見ると、コース 1 ではどの被験者も主に曲線を描くように操縦していることがわかる。これは、左手で左の車輪の速度を、右手で右の車輪の速度を調節し操縦した結果である。また、被験者 B の軌跡は被験者 A、C の軌跡よりも密度が濃く、所要時間も長いので、ゆっくりとした速度で操縦したことがわかる。次に、図 11(b)、図 12(b)、図 13(b) を見ると、コース 1 とは異なり主に直進と旋回を用いていることがわかる。また、図 11(c)、図 12(c)、図 13(c) では、後退する際に速度を落としていることがわかる。最後に、どの被験者もほぼ決められたコース通りに操縦できていること、さらに行動の変更や速度の調節を有効に用いていることから手形状認識が確実に行われたと考えられる。

4 まとめ

本研究では、実時間で手形状を認識する手法として、2次元画像を用いた手法について述べ、移動ロボットの遠隔操縦を行った。

まず、2次元画像から手領域を抽出し、輪郭線における曲率を用いて指先位置を検出する手法について説明し、検出された指先位置に基づいて特徴量を計算し、手形状を認識する手法を提案した。実験では、数種類の手形状について検出された指先位置から求まる特徴量 ϕ の値を求め、これに基づいて認識実験を行い高い認識率が得られることを確認した。

また、移動ロボットの遠隔操縦に応用し、被験者の指示通りロボットを操縦できることを実証した。

次に、今後の課題について述べる。まず、本手法では2次元画像の色情報を用いて手領域を抽出しているため、照明条件が変化すると手領域の抽出が困難になる。そのため、色以外にも何らかの情報を用いて、周囲の環境の変化に強い手領域抽出手法を開発する必要があると考えられる。また手形状の認識では、特徴量の平均値によって閾値の範囲を設定し、すべての形状の特徴量に対して同じ範囲を用いているが、それぞれの形状に応じた適切な閾値を設定することで、誤認識を抑えた認識率の向上が期待される。さらに、2次元画像から得られる様々な情報を利用することで、より多くの手形状の認識につながると考えられるため、平面内を移動するロボットだけでなくより自由度の高いロボットマニピュレータの遠隔操縦にも応用したい。

参考文献

- [1] 西川敦, 大西映生, 西村正典, 平野敦士, 小荒健吾, 宮崎文夫: 局所相関演算に基づくオプティカルフローを用いた身振り動作の認識手法, 情報処理学会論文誌, Vol. 40, No. 8, pp. 3118-3133 (1999).
- [2] 上田悦子: 多視点シルエット画像を用いた手の形状推定,
<http://robotics.aist-nara.ac.jp/~etsuko-u/hand1.html>.
- [3] 安居院猛, 長尾智晴: C 言語による画像処理入門, 昭晃堂 (2000).
- [4] 手塚慶一, 北橋忠宏, 小川秀夫: デジタル画像処理工学, 日刊工業新聞社 (1985).
- [5] 升谷保博, 福留正一, 宮崎文夫: 車輪型移動ロボットを用いたメカトロニクス実習, 日本機械学会ロボティクス・メカトロニクス講演会'96 講演論文集, Vol. A, pp. 401-404 (1996).
- [6] 株式会社応用計測研究所: リアルタイム動作解析システム Quick MAG System III ユーザーズマニュアル.

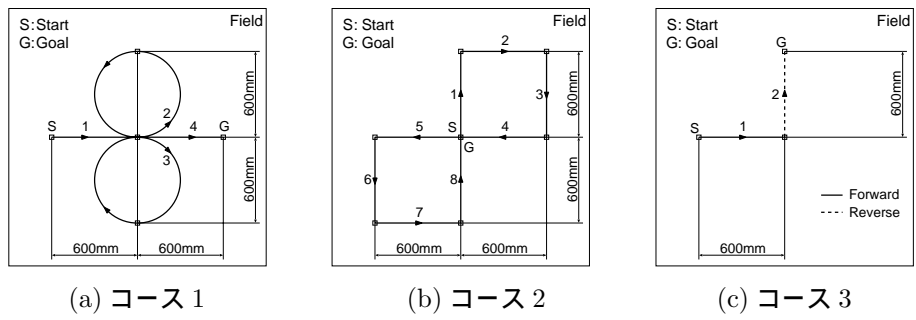


図 9: 移動ロボット遠隔操縦タスクに用いたコース

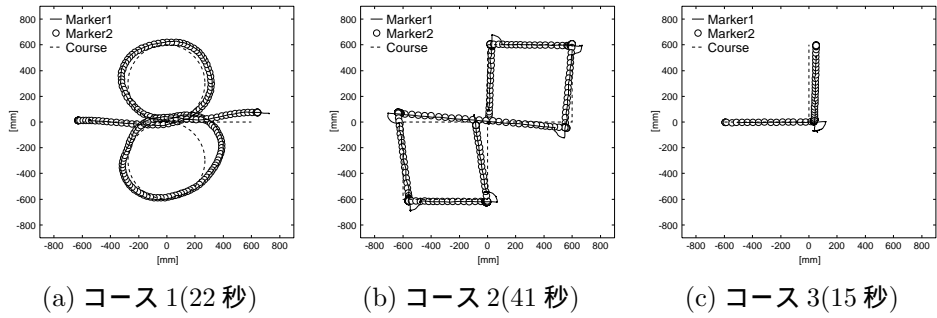


図 11: ロボットの軌跡：被験者 A

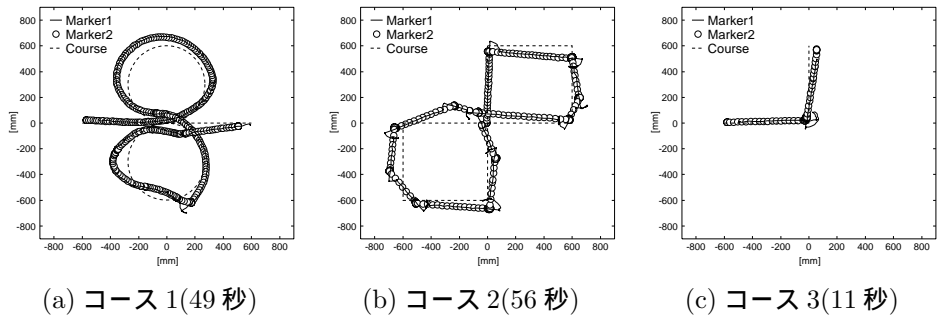


図 12: ロボットの軌跡：被験者 B

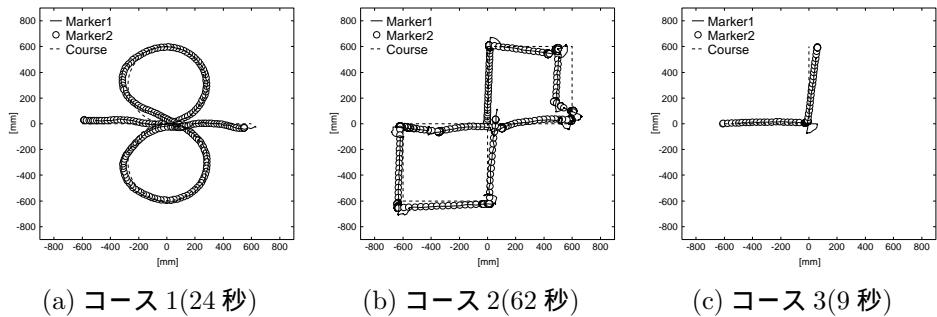


図 13: ロボットの軌跡：被験者 C