

ロボットによる柔軟物操作における複数視点撮影のための ハンドアイシステム

○渡部知俊 松野隆幸 白川智也 見浪護 (岡山大学)

Hand-eye Camera System for Capturing Distance Image from Multi Viewpoints in Deformable Object Manipulation by Robot

*T. Watanabe and T. Matsuno, T. Shirakawa, M. Minami (Okayama University)

Abstract— Our research purpose is manipulation of deformable object by a robot. This paper describes both mechanism of end-effector and view direction of a distance camera. The distance camera is mounted to end-effector to obtain point cloud data of string surface from various viewpoints. In order to obtain low-deficit surface of a string without occlusion, distance images are captured from five different viewpoints. Those five images are merged with ICP algorithm, and improvement of ICP requires information concerning each viewpoints. Therefore, homogeneous transformation matrix is used to decide the photographing direction of the distance camera. At final, the result of string untying experiment is reported.

Key Words: Deformable object, Hand-eye system

1 はじめに

我々の身の回りに存在する布やひもといった物体は柔らかく形状が一定に定まりにくい不定形物体と呼ばれる。この不定形物体のマニピュレーションは様々に変化する形状に対して柔軟に対応する必要があり、画一的な動作で操作することができないため困難である。近年増加しているセル生産方式ではロボットマニピュレータが作業の自動化に貢献している。このような生産の現場にもケーブルのような不定形物体が存在しており、ロボットによる配線作業の自動化が期待されている¹⁾。また家庭用ロボットが作業を行う人々の生活圏には様々な柔軟物が存在しそれらを自由に操作することは必要不可欠である²⁾。本研究は不定形物体であるひもに着目する。この分野に関する先行研究としては高松ら³⁾によってひもの形状や状態の記述法に関する研究がなされている。また森田ら⁴⁾や若松ら⁵⁾、松野ら⁶⁾はロボットマニピュレータによるひもの形状操作を目的とした研究を行い、カメラで床に置かれたひもの形状を認識し、結び目理論やトポロジーを応用した状態遷移の理論、それらを使ったロボットの動作計画などを提案している。それらに対して我々の研究では、これまで距離カメラで対象となる紐を三次元の点群として取得し、そこから紐の形状を位置ベクトルの列で構成する点連鎖モデルと呼ばれるものを生成するためのアルゴリズムを提案した⁷⁾。このアルゴリズムでは、複数の視点から得られた点群データを合成するICPアルゴリズム⁸⁾、物体表面の法線推定、法線ベクトル情報をもとに紐の中心軸を推定する中心軸推定、紐の太さ情報を利用したマッチング法などの複数の点群処理を施すことによって、紐の交点での誤認識を解消し、オクルージョンのないひもの3次元形状を復元できることを示した。

我々の研究ではひもの形状操作に実現のためにFig.1に示すように形状認識、形状操作のための計画、ロボットの動作生成、形状操作の実行という4つの段階を設定した。第1段階の形状認識フェイズでは操作の対象となるひもの形状を認識している。認識された形状と

目標形状との差異をフィードバックすることで対象となるひもの形状を制御する。第2段階は形状操作計画の生成フェイズである。ここでは現在のひもの形状から目標形状に変形させるための最適な形状操作を決定する。第3段階の動作生成フェイズでは生成された形状操作計画を実現するためのロボットの動作を生成する。そして第4段階の形状操作実行フェイズでは生成されたロボットの動作を実行する。形状操作が完了されたら再度形状認識フェイズに戻り、目標形状に一致するまでこれらの処理を繰り返す。この際、ロボットによるひもの形状操作を自動化するためにエンドエフェクタに距離カメラを取り付けている。そのためハンドアイシステムを必要とする。本論文では実装されているハンドアイシステムについて述べ、それを用いたひも解き操作実験の結果についても報告する。

2 ひも形状操作システムの概要

本研究で用いているひも形状操作システムのフローチャートをFig.2に示す。本システムは形状認識、操作計画の導出、形状操作に大別され、形状認識、操作計画の導出、形状操作の処理を複数回繰り返すことでひもを任意の形状に操作する。

初期条件として目標となるひもの形状が入力され、処理を開始する。最初の処理は形状認識であり、対象となるひもの形状を3次元点群データとして取得する。ひも形状認識法によってひもの位置や形状を表す点連鎖モデルが生成される。次に、ひもの形状を目標形状に近づけるための最適な形状操作の計画を立てる。本研究では、形状操作の基本的な操作としてライデマイスター移動およびクロスを採用している。ライデマイスター移動、クロスは結び目理論で定義された操作であり、それらを組み合わせることであらゆるひもの形状を実現することが証明されている。これらの形状操作の難しさをコストとして定義し、コストが最小となるように形状操作計画を決定する。ロボットの動作生成においては形状操作計画に基づき動作を生成する。最後に生成されたロボットの動作が実行され、再び形状認識から処理を行う。

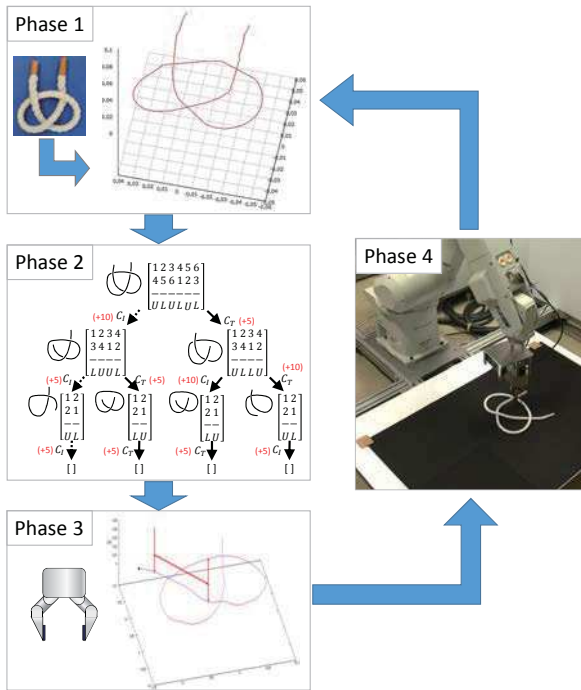


Fig. 1: ひも形状操作の流れ

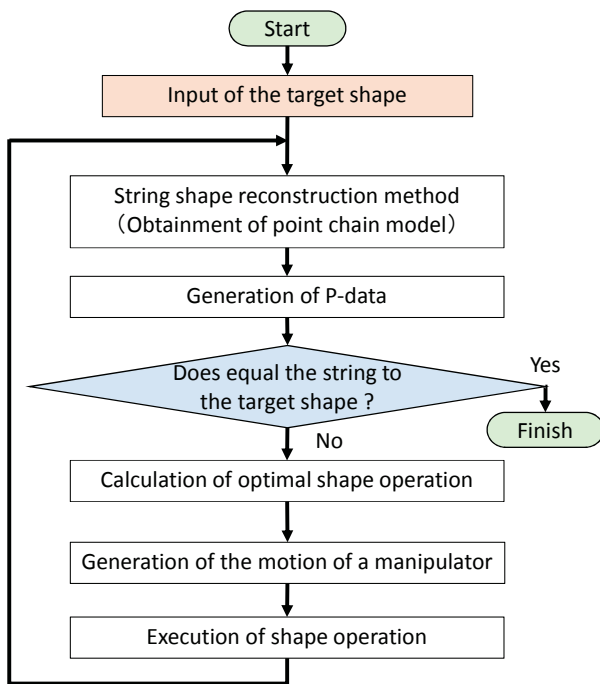


Fig. 2: ひも形状操作システム

2.1 形状認識

形状操作の概要を Fig.3 に示す。形状認識においては、はじめに対象となるひもの表面形状を距離カメラを用いて点群として取得する。この際一視点からの計測ではひもの交差点においてオクルージョンが発生してしまう。そのため異なる 5 つの視点からひもの表面形状を取得し、ICP アルゴリズムを用いて 5 つの 3 次元点群データを合成する。合成された点群データをもとにひもの中心軸を推定し、それらの点を連鎖させることでひもの形状を復元する。ここで生成されたモデ

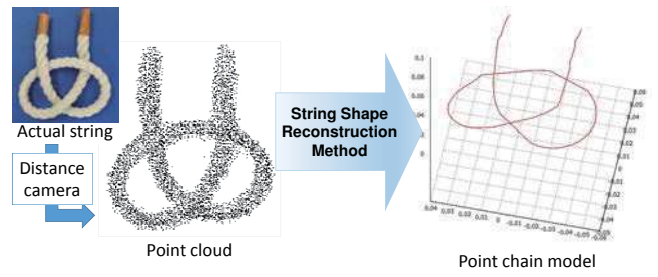


Fig. 3: 形状認識

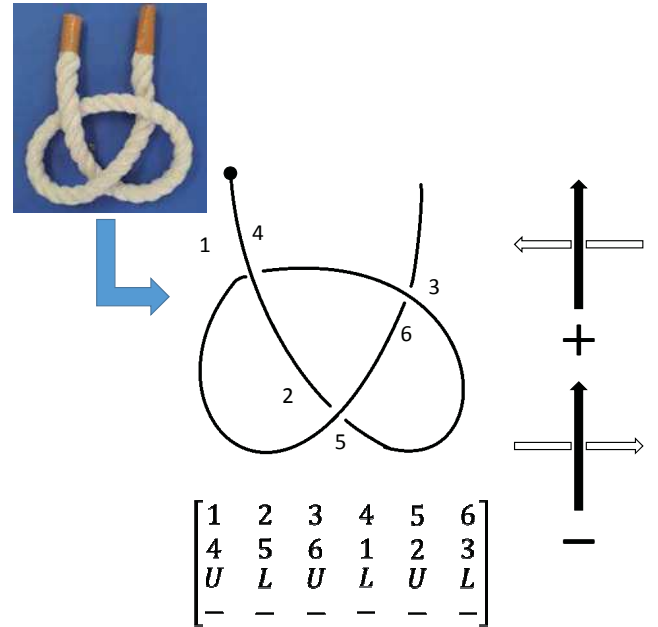


Fig. 4: P-data

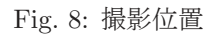
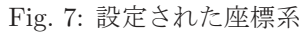
ルを点連鎖モデルと呼ぶ。

2.2 形状操作計画の導出

本システムでは結び目理論で定義されているライデマイスター移動およびクロスと呼ばれる操作を複数回行うことでひものを任意の形状に操作を行う。ライデマイスター移動およびクロスの変形を Fig.5 に示す。操作計画においては、点連鎖モデルからトポロジーのみの情報を抽出し Fig.4 のような P-data と呼ばれる行列で表現する。Fig.4 にあるひもの結び目の射影図とその P-data を示す。P-data はひもの射影図から生成される。3 次元空間内のひものを 2 次元平面に射影して射影図として表現する。射影図の中でひもの端を端点と、ひも同士が交差した点を交点と、端点や交点で区切られた線分をセグメントと呼ぶ。P-data 中の第 1 行および第 2 行の数字は交点の番号を示しており、一交点には 2 つの番号が対応している。第 3 行は交点における交差の方向を表しており、+ または - で表現される。第 4 行は交点におけるひもの上下方向を表している。U は交点において上側であることを、L は交点において下側であることを示している。

前述したライデマイスター移動およびクロスという操作は P-data に行列の変換という形で行うことが出来、行数を増減させることで処理が行われる。与えられた目標形状と現在の形状を P-data 上で比較しどのよ

Viewpoint	RM I	RM II	CR(U,U)	CR(U,L)	CR(L,U)
Decline of nodes	fair	good	fair	fair	fair
Success rate	fair	fair	good	fair	good
Interference between segments	fair	good	good	fair	fair
Interference between segments and the robot's hand	fair	good	good	bad	good
Cost	8	1	3	10	5


$$w_{TV3} = \begin{bmatrix} 0.259 & 0 & 0.966 & 116.47 \\ 0 & 1 & 0 & 0 \\ -0.966 & 0 & 0.259 & 434.67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$
$$w_{TV4} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0.259 & 0.966 & 0 & -116.47 \\ -0.966 & 0.259 & 0 & 434.67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$
$$w_{TV5} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -0.259 & 0.966 & 0 & 116.47 \\ -0.966 & -0.259 & 0 & 434.67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$wT_{V2} = \begin{bmatrix} -0.259 & 0 & 0.966 & -116.47 \\ 0 & 1 & 0 & 0 \\ -0.966 & 0 & -0.259 & 434.67 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

距離カメラによる対象物体の計測の際に 1 方向からのみで計測を行うとオクルージョンが生じ、対象物体の全体形状を復元することは困難である．そのためさまざまな視点から計測された 3 次元点群データを ICP(Iterative Closest Point) アルゴリズムを用いることで 1 つの点群データに合成する．ICP アルゴリズムを用いるうえで、合成する点群データの位置をある程度近づける必要があり、そのために前述の同次変換行

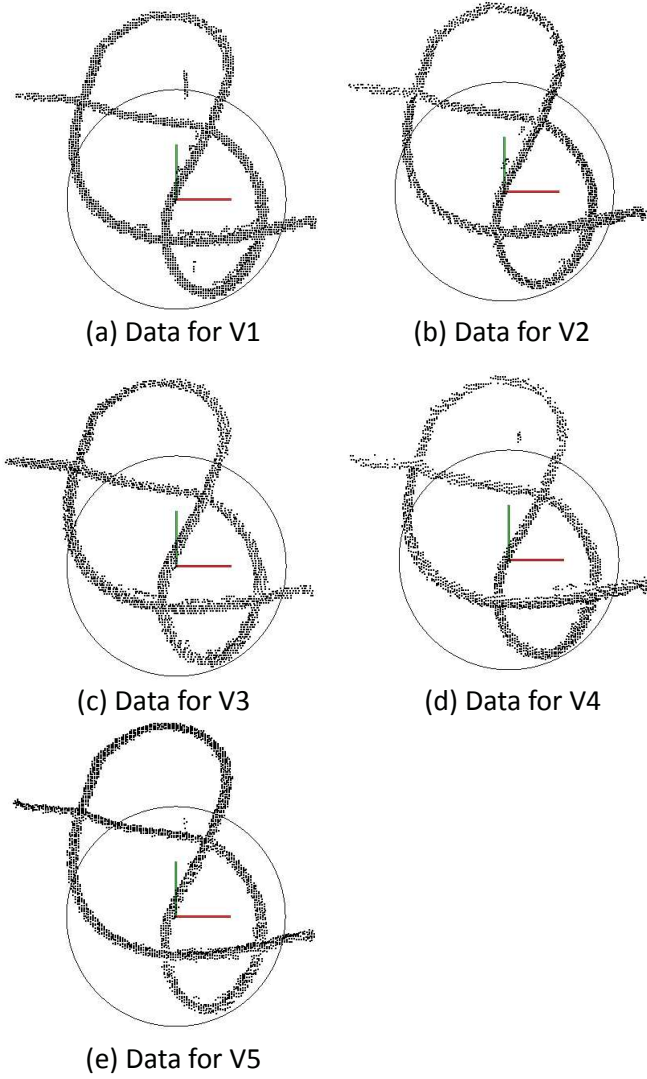


Fig. 9: サンプルデータ

列を用いて初期位置あわせを行う。今回、ICP アルゴリズムは初期位置あわせで出た誤差を修正するために使用する。式 (7) の f が最小となるように $\mathbf{R}_i$ と $\mathbf{t}_i$ を決定する ($i = 2, \dots, 5$)。

$$f(\mathbf{R}_i, \mathbf{t}_i) = \sum |\mathbf{R}_i \mathbf{x}_i + \mathbf{t}_i - \mathbf{y}_i|^2 \quad (7)$$

$\mathbf{R}_i$ は回転行列、 $\mathbf{t}_i$ は並進ベクトル、 $\mathbf{x}_i$ と $\mathbf{y}_i$ は2つの点群データの対応点を表している。ICP アルゴリズムは外れ値に影響されやすいという問題があるが、前処理としてあらかじめ外れ値を除去している。

4 ICP アルゴリズムの有効性検証

ICP アルゴリズムの有効性を検証するためのサンプルデータを Fig.9 に示す。これらの3次元点群データは初期位置あわせを行った後のデータである。

式 (2)~(6) を用いて初期位置あわせを行った際に、誤差が残る。このときの各3次元点群データの座標系を順に Σ_{w1} , Σ_{w2} , Σ_{w3} , Σ_{w4} , Σ_{w5} とする。そのためICP アルゴリズムを用いて初期位置あわせで出た誤差を修正するために用いる。式 (2) を用いて得られた座標系 Σ_{w1} がワールド座標系 Σ_w と一致していると仮定

する。ICP アルゴリズムは式 (2) を用いて得られた座標系 Σ_{w1} を基準とし、式 (7) の f を最小にする回転行列 $\mathbf{R}$ 、並進ベクトル $\mathbf{t}$ を求める。この行列およびベクトルから Σ_{w2} から Σ_{w5} を Σ_{w1} に位置合わせするための同次変換行列 ${}^{w1}T_{Vi}$ が以下のように表される。

$${}^{w1}T_{wi} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix} \quad (8)$$

Fig.9 に示したサンプルデータに ICP アルゴリズムを用いて Σ_{w2} から Σ_{w5} を Σ_{w1} に位置合わせするための同次変換行列を求める。ICP により得られた同次変換行列を用いて座標系 Σ_w から各撮影姿勢までの同次変換行列の推定値 ${}^w\hat{T}_{Vi}$ を式 (9) を用いて求める。式 (9) から得られる Σ_{w1} 座標系から各撮影位置姿勢を表す同次変換行列は式 (10) から (14) のようになる。

$$\begin{aligned} {}^w\hat{T}_{Vi} &= {}^wT_{wi} {}^{wi}T_{Vi} \\ &\approx {}^{w1}T_{wi} {}^{wi}T_{Vi} \end{aligned} \quad (9)$$

$${}^w\hat{T}_{V1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 450 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$${}^w\hat{T}_{V2} = \begin{bmatrix} -0.26 & 7.27 \times 10^{-9} & 0.97 & -116.5 \\ 1.44 \times 10^{-8} & 1 & 1.16 \times 10^{-8} & -6.5 \times 10^{-6} \\ -0.97 & -1.29 \times 10^{-8} & -0.26 & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$${}^w\hat{T}_{V3} = \begin{bmatrix} 0.26 & -3.57 \times 10^{-8} & 0.97 & 116.5 \\ 2.57 \times 10^{-8} & 1 & 2.98 \times 10^{-8} & -1.2 \times 10^{-5} \\ -0.97 & -2.01 \times 10^{-8} & 0.26 & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$${}^w\hat{T}_{V4} = \begin{bmatrix} -4.04 \times 10^{-10} & -1.38 \times 10^{-8} & 1 & 1.85 \times 10^{-7} \\ 0.26 & 0.97 & -3.5 \times 10^{-8} & -116.5 \\ -0.97 & 0.26 & -8.1 \times 10^{-9} & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$${}^w\hat{T}_{V5} = \begin{bmatrix} -4.70 \times 10^{-6} & -6.3 \times 10^{-7} & 1 & 2.11 \times 10^{-3} \\ -0.26 & 0.97 & -7.2 \times 10^{-7} & 116.5 \\ -0.97 & -0.26 & -4.7 \times 10^{-6} & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

ここで式 (3) から (6) を式 (10) から (13) と比較すると、姿勢行列の最大偏差は 0.004、並進行列の最大偏差は 0.03 であり、ICP による補正は小さい。

ここで ICP アルゴリズムを用いて位置あわせを行った点群と初期位置あわせのみを行った点群を Fig.10 に示す。また、これらの3次元点群データをもとに生成された点連鎖モデルは Fig.11 に示す。Fig.10(a) をもとに生成された点連鎖モデルは Fig.11(a) である。このとき点連鎖モデルは途中で途切れており、正確な形状を認識することが出来ていない。Fig.10(b) をもとに生成された点連鎖モデルは Fig.11(b) である。このときの点連鎖モデルは途中で途切れることなく、全体の形状を認識することが出来ている。また、点群のばらつきを評価するためにひもの半径を求める。Fig.12 に今回サ

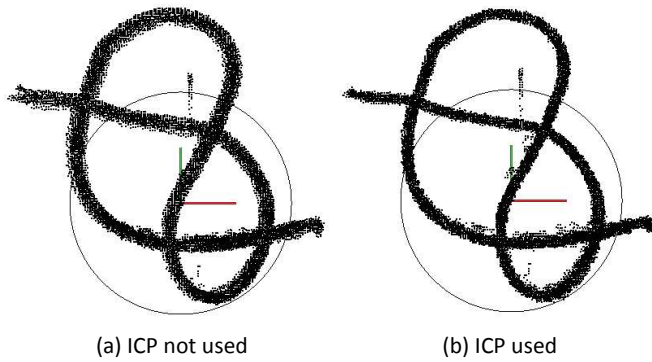


Fig. 10: 点群の比較

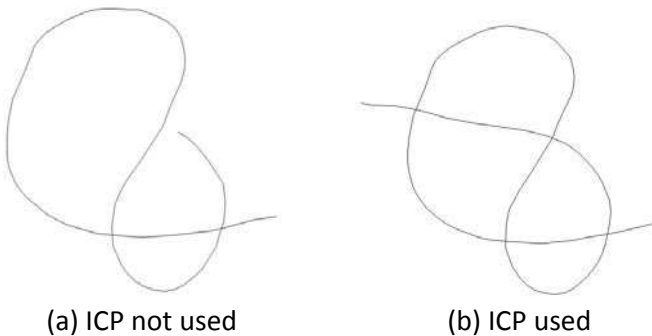


Fig. 11: 点連鎖モデルの比較

ンプリングを行った点を赤い点として示す. それぞれのサンプリング点について, Fig.12 を Fig.10(a),(b) に投影した際の近傍点における最大半径を求めた結果を Table3 に示す. サンプリングした 10 点中 9 個の点において, ICP を用いた点群におけるひもの半径が ICP が行わなかった場合より小さい. 点連鎖モデルを生成する際に, 点群を細くすることでより精度が高くなるため, ICP アルゴリズムを用いれば補正量は小さいが, 点連鎖モデルを精度よく生成できる.

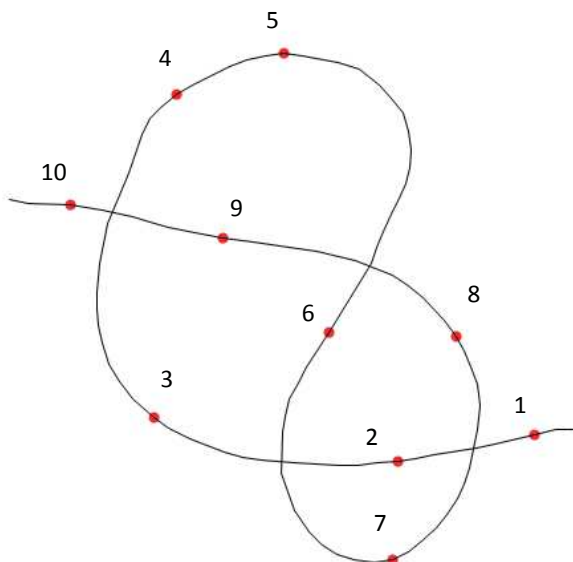


Fig. 12: 半径のサンプリング点

Table 3: 各点における最大半径

Point	maximum radius[mm]	
	ICP	without ICP
No.1	6.27	8.39
No.2	7.52	8.38
No.3	6.39	7.10
No.4	6.60	9.43
No.5	5.53	8.86
No.6	8.69	8.83
No.7	8.96	8.54
No.8	5.44	7.44
No.9	5.77	8.14
No.10	6.32	8.24
平均	6.75	8.34
標準偏差	0.00125	0.00068



(a) Sample 1

(b) Sample 2

Fig. 13: 実験で用いたひもの形状

5 ひも形状操作実験

今回はひも解き操作の実験について述べる. ひもの交点が 0 個であるとき, ひもが結ばれていないことは明らかであるため, 本実験における目標形状の P-data を空行列とする. 今回実験を行ったひもの形状を Fig.13 に示す. このひもの長さはおおよそ 1[m] で, 直径は 8.4[mm] である. なお, 実験結果についてはひも解き操作の成功ケースのみを報告する.

5.1 実験の流れ

実験の流れを Fig.14 に示す. はじめに Fig.14(a) において異なる 5 つの位置姿勢から距離カメラを用いて対象となるひもを計測する. その後 Fig.14(b) において形状認識を行い, 点連鎖モデルおよび P-data が生成される. 生成された P-data が目標とする形状の P-data と一致しているか判別をする. 一致していない場合は現在および目標形状の P-data をもとに現在のひも形状を目標形状に近づけるための操作計画を生成する. さらに, 実際にひもに行う操作の軌道を計算する. Fig.14(c) において実際に操作を行い, Fig.14(d) において操作を実行完了する. これらの処理をひもの形状が目標形状と一致するまで繰り返し行う.

5.2 実験結果

5.2.1 サンプル 1

Fig.13(a) に示した形状のひもに対して実験を行った結果を Fig.15 に示す. また, この形状における操作計画とコストを Fig.16 に示す.

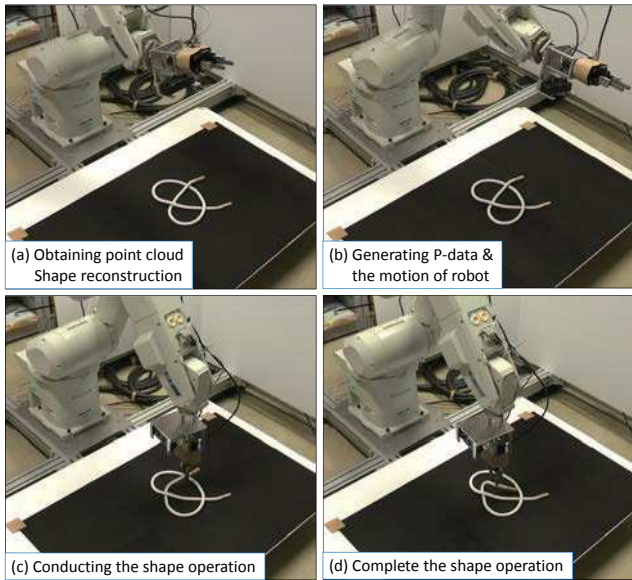


Fig. 14: 実験の流れ

Fig.15(a) に示した初期形状のひもに対して形状認識および操作軌道生成の結果を Fig.15(e) に示す. このとき選択された操作は CR(U,L) である. このとき赤い矢印はロボットによる把持操作の軌道である. 操作後のひもの形状は Fig.15(b) である. このとき 2 番の交点においてセグメント同士の干渉が起きたが, 交点を取り除くことが出来ている. 再度認識および操作軌道の生成を行った結果は Fig.15(f) に示す. このとき選択された操作は CR(L, U) である. 操作後のひもの形状は Fig.15(c) である. このときの操作はひも同士の干渉も少なく形状操作は停滞することなく行われた. 再度認識および操作軌道の生成を行った結果は Fig.15(g) に示す. このとき交点が 1 個であるため, 選択された操作は CR(U,L) であった. 操作後の形状は Fig.15(h) となった. このときひもの交点が 0 個となり, ひも解き実験は終了した.

5.2.2 サンプル 2

Fig.13(b) に示した形状のひもに対して実験を行った結果を Fig.17 に示す. Fig.17(a) に示した初期形状のひもに対して形状認識および操作軌道生成の結果を Fig.17(e) に示す. このとき選択された操作は CR(L,U) である. 操作後のひもの形状は Fig.17(b) であり, 再度認識および操作軌道の生成を行った結果は Fig.17(f) に示す. このとき選択された操作は CR(U,L) である. 操作後のひもの形状は Fig.17(c) であり, 再度認識および操作軌道の生成を行った結果は Fig.17(g) に示す. このとき交点が 1 個であるため, 選択された操作は CR(U,L) であった. 操作後の形状は Fig.17(h) となった. このときひもの交点が 0 個となり, ひも解き実験は終了した. サンプル 2 の実験においては全ての動作が停滞することなく行われた.

6 おわりに

本論文では実装されているハンドアイシステムについて述べ, それを用いたひも解き操作実験の結果についても報告した. ひもの形状操作の結果, ひもが距離カメラの撮影範囲外に出てしまうことがあった. 現在は

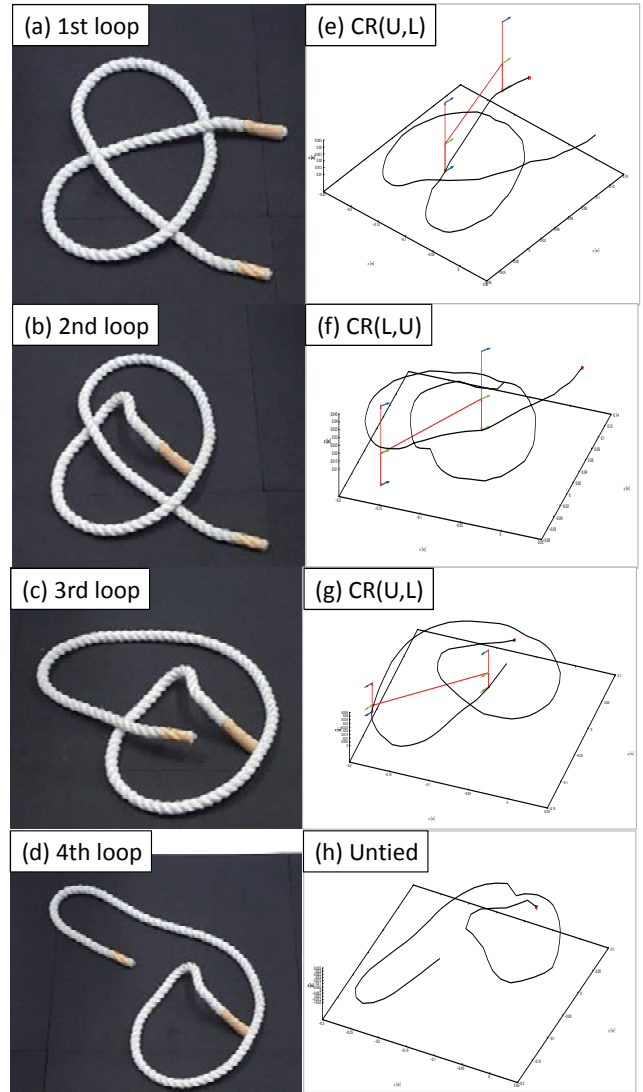


Fig. 15: 実験結果 1

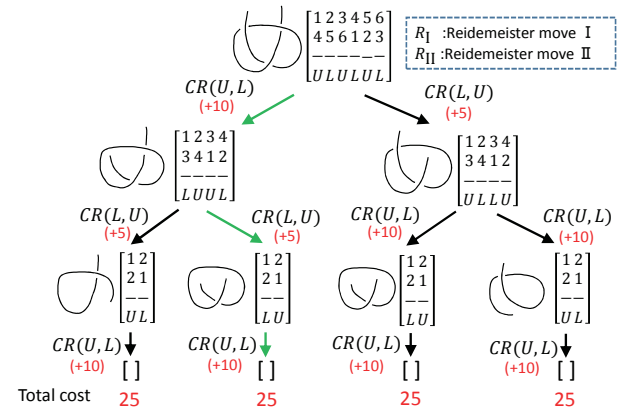


Fig. 16: 実験結果 1-1

あらかじめ指定した 5 つの位置姿勢からのみで距離カメラによる撮影をおこなっており, このような状況ではひもの全体形状を復元することは不可能である. そのためアルゴリズムの改良を行い, 任意の位置姿勢から撮影を行い点群データを補完する必要があると考えられる.

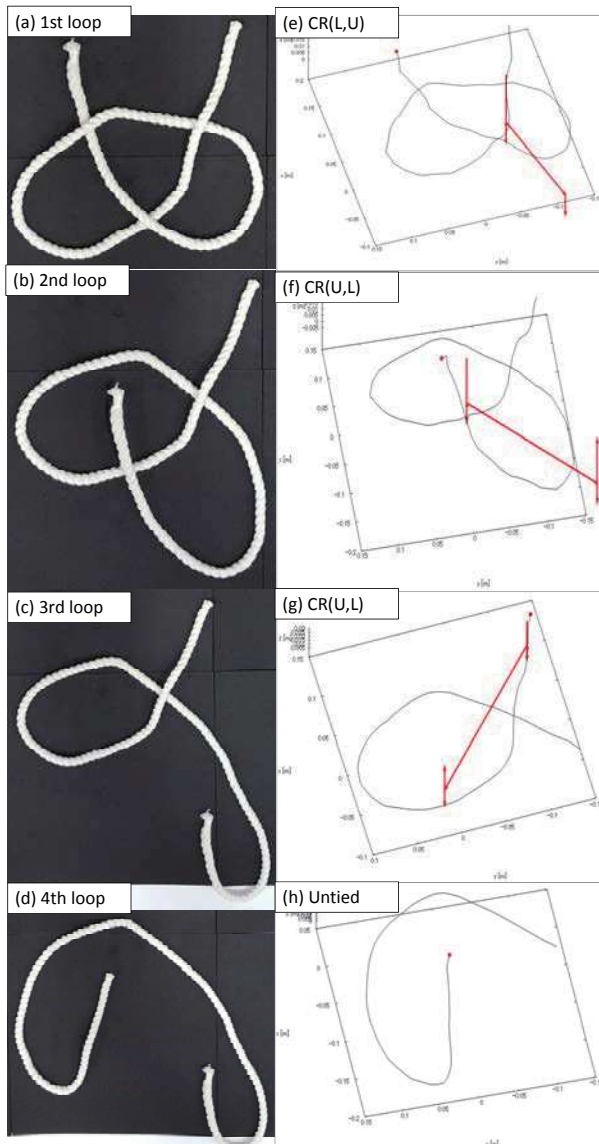


Fig. 17: 実験結果 2

のためのロープの形状認識”, 計測自動制御学会論文集, Vol.41, No.4, p.366~p.372, 2005.

- 7) Keisuke Mukai, Takayuki Matsuno, Akira Yanou, Mamoru Minami, "Shape Modeling of A String And Recognition Using Distance Sensor", IEEE 24th International Symposium on Robot and Human Interactive Communication, Aug.31-Sep.4, 2015.
- 8) 中井康博, 美濃導彦, "複数の点群に対する位置合わせ手法の性能比較", 特別研究報告書, 2012.

謝辞

本研究は MEXT 科研費 JP26820088 の助成を受けたものです。

参考文献

- 1) 鷺見和彦, “解説 柔軟物も取り扱える生産用ロボットシステムの開発,” 日本ロボット学会誌, Vol.27, No.10, pp1082-pp1085, Jul.2009.
- 2) 金子学, 柿倉正義, “家庭用サービスロボットののための柔軟物体のハンドリングに関する研究-布地の画像処理について-”, コンピュータビジョンとイメージメディア, 135-2, 2002.
- 3) 高松淳, 森田拓磨, 小川原光一, 木村浩, 池内克史, “ロボットによる実行を目的とした結び作業の記述”, 日本ロボット学会誌 Vol.23No.5, pp.572~582, 2005.
- 4) 森田拓磨, 高松淳, 小川原光一, 木村浩, 池内克史, “観察によるひも結び動作の学習”, コンピュータビジョンとイメージメディア, 135-12, 2002.
- 5) 若松栄史, 妻屋彰, 荒井栄司, 平井慎一, “結び/解き操作を含めた線状物体のマニピュレーション”, 日本ロボット学会誌, Vol.23, No.3, pp.344~351, 2005.
- 6) 松野隆幸, 玉置大地, 新井史人, 福田敏男, “トポロジカルモデルと結び目不変量を用いたマニピュレーション