

2Y-09

折り紙技法を活用したフレキシブルディスプレイにおける形状認識 およびジェスチャ推定

小宮山 憂† 木下 雄一朗† 郷 健太郎†

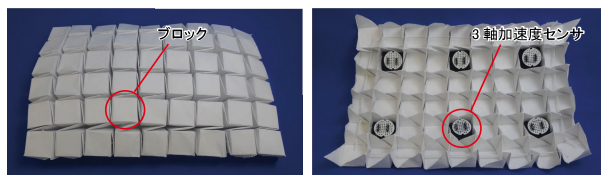
山梨大学工学部コンピュータ理工学科†

1 はじめに

近年HCIの分野では変形可能なユーザインタフェースの研究が盛んに行われている。その一つであるフレキシブルディスプレイは柔軟な特性を生かした新しい入出力装置として注目を集めている。Kinoshita [1] らは、平織りを利用した Origami Tessellation Display (OTD) を提案した。平織りとは、特殊な折り方で紙を折ることで幾何学的な基本パターンを平面に充填させ、柔軟な平面を作り出すものである。先行研究 [1] では OTD によって地図アプリケーションを操作する際のジェスチャを導出し、アプリケーションを実装することで導出されたジェスチャの有効性を示した。しかし、ディスプレイの形状認識およびジェスチャ推定は外部からのカメラトラッキングによって実現しており、持ち運びが想定されるディスプレイのジェスチャ認識方法として現実的ではない。Rendら [2] の研究では外部装置に頼ることなく平面上に張られた圧電センサによってシートの形状認識を行っている。そこで本研究では外部装置に頼ることなく OTD の形状認識とジェスチャ推定を行うシステムを開発することを目的とする。具体的には OTD にセンシング機構を搭載し、操作時および非操作時における形状認識とジェスチャの推定を行うことによって、OTD の持ち運びを実現し利用の幅を広げる。

2 ハードウェア構成

本研究の入力デバイスとして、先行研究 [1] と同様に Water Bomb と呼ばれる平織りを採用した OTD を利用する。本稿では、図 1(a) に示す OTD の平面を構成する平織りの最小単位をブロックと呼ぶ。今回は図に示すよう縦に 6 つ、横に 9 つ並ぶブロックによって構成される OTD モックアップを使用する。

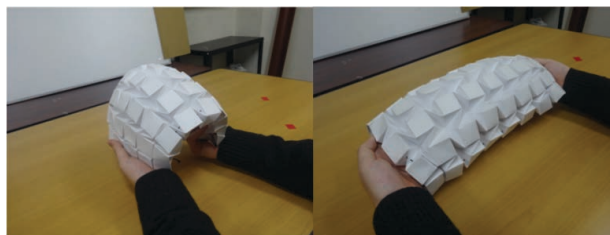


(a) 表面 (b) 裏面
図 1: 使用した OTD モックアップ

Shape Recognition and Gesture Estimation for Origami-based Deformable Surfaces

†Yu Komiya, †Yuichiro Kinoshita, †Kentaro Go

†Department of Computer Science and Engineering, University of Yamanashi



(a) 曲げる (b) 伸ばす

図 2: OTD の変形の様子

この平織りでは、図 2 に示すようブロック間の隙間が広がったり狭まったりすることによってデバイスの曲げや伸ばしを実現される。この曲げや伸ばしには各ブロックに傾きが伴うためこれをセンシングすることで、ディスプレイの形状認識およびジェスチャの推定を実現する。センシングには、図 1(b) に示すように 3 軸加速度センサモジュール (kxr94-2050) を 9 cm 間隔に 6 基配置した。3 軸加速度センサモジュールはマイコンボード (Arduino Uno) に接続され、マイコンボードによって統括された情報を形状認識およびジェスチャ推定システムへ入力情報として送信する。

2 形状認識システム

2.1 形状認識システムの構築

本研究の目指す形状認識システムでは、6 基の 3 軸加速度センサからの値を入力情報として扱う。図 3 に示すように受け取った値をもとに、各ブロックの位置座標とオイラー角を出力する。そして、出力された結果を 54 ブロック分統合することで OTD 全体の形状を認識する。システムの構築において、3 軸加速度センサからの入力と、ブ

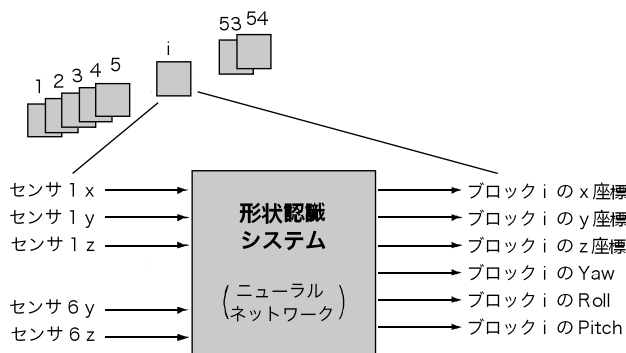


図 3: 形状認識システムの処理

ロックの位置座標およびオイラー角との関係を学習させたモデルを各ブロック毎にニューラルネットワークを用いて作成する。学習には、様々な形状のOTDにおける6基の3軸加速度センサの値18と、その時のブロックの位置座標(x, y, z)とオイラー角(Yaw, Roll, Pitch)の値6との組み合わせを教師データとして利用した。ブロックの位置座標とオイラー角は3次元位置測定装置であるPatriot(Polhemus社)によって測定した。30種類のOTDの形状についてこれらの値を計測し、合計で30のデータを収集した。

2.2 形状認識システムの検証

形状認識システムの検証には、leave-one-out cross validationを用いた。本稿では例として、図2(a)に示す曲げる形状のデータをテストデータとし、それ以外の29のデータを学習データとして作成したモデルの検証結果を示す。図4は、OTDの各ブロックにおけるモデル出力値と実測値との誤差を表し、色が濃いほど誤差は大きい。図4(a)は3次元位置座標(x, y, z)における誤差(単位: cm)であり、図4(b)はオイラー角(Yaw, Roll, Pitch)の誤差の平均(単位: deg.)を表す。なお、3軸加速度センサが配置されているブロックのオイラー角は3軸加速度センサの値により既知であるため空白としてある。検証の結果より、精度は高くないものの、本システムを用いた形状認識は可能であることが確認できた。精度が高くない原因として、学習に使用する教師データの数が少ないということや限られたバリエーションのジェスチャデータで学習を行ったことなどが考えられる。今後学習に使用するデータの数を増やし、幅広いバリエーションのデータで学習を行うことによってシステムの精度を高めることができると考える。

3 ジェスチャ推定システム

3.1 ジェスチャ推定システムの構築

本研究の目指すジェスチャ推定システムでは、形状認識システムと同様に6基の3軸加速度センサからの値を入力情報として扱う。図5に示すよう受け取った値をもとに、現在行われているジェスチャを推定し出力を行う。システムの構築にあたりRandom Forestによる機械学習を利用し、センサ値からジェスチャ推定を行う決定木を作成した。学習に使用するデータとして、ジェスチャとして認識する6つのジェスチャを実際に行い、その時の3軸加速度センサの値を各10回ずつ測定し、合計60のデータを収集した。

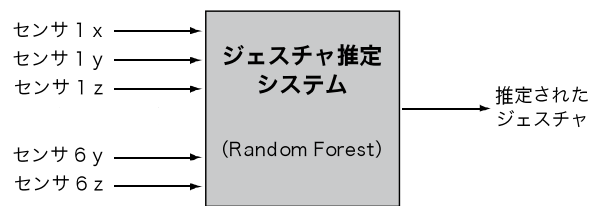


図5: ジェスチャ推定システムの処理

またそれ以外にジェスチャとして認識しない様々なOTDの形状のデータを240収集し、合計で300のデータによって学習を行った。学習の結果得られた決定木は6つのジェスチャとそれ以外の合計7つのクラスに分類を行う。本システムでは10個の決定木を作成した。

3.2 ジェスチャ推定システムの検証

本ジェスチャ推定システムの検証には、10-fold cross validationを用いた。検証の結果、正答率は93%(標準偏差0.04%)であった。これらの結果から本システムを使用した6基の3軸加速度センサによるジェスチャの推定は十分に可能であることがいえる。また不正解となったデータを分析してみた結果、認識させたい6つのジェスチャを、ジェスチャとして認識しないデータであると判断してしまう場合が多くみられた。これは6つのジェスチャのデータ数がそれ以外のデータの数に比べて少なかったことで、cross validationの分割の際にデータの偏りが生じやすくなり、うまく学習ができていなかったことが原因と考えられる。今後は6つのジェスチャデータの数を増やしそれ以外のデータとのバランスをとることでデータの偏りを防ぎ、本システムの精度をさらに向上させることができると考える。

4 おわりに

本研究では新しいフレキシブルディスプレイである折り紙テセレーションディスプレイ(OTD)における形状認識およびジェスチャ推定システムを構築した。検証の結果から本システムを用いたOTDの形状認識とジェスチャ推定が十分に行えることが示された。今後の課題として、開発コストを抑えるために、センサの数を減らしても十分な認識・推定精度を持つシステムを実現する点があげられる。

参考文献

[1] Kinoshita, Y., Go, K., Kozono, R., Kaneko, K., Origami Tessellation Display: Interaction Techniques Using Origami-based Deformable Surfaces. ACM CHI EA '14, 1837-1842 (2014)
 [2] Rend, C., Kim, D., Fanello, S., Parzer, P., Rhemann, C., Taylor, J., Zirkl, J., Scheipl, G., Rothländer, T., Haller, M., Izadi, S., FlexSense: A Transparent Self-Sensing Deformable Surface. In Proc. ACM UIST '14, 129-138 (2014)

3.7	2.6	2.7	1.1	1.9	2.2	1.8	3.9	4.5	8.1	4.2	8.0	8.0	2.8	3.7	6.3	6.6	3.9
5.8	1.6	1.7	1.8	1.1	1.7	2.1	2.6	2.3	4.3	6.3	2.3	3.9	8.4	10.0			
2.8	2.5	1.8	0.8	0.3	1.5	1.5	2.8	2.2	5.7	2.4	8.7	6.2	3.1	5.2	12.6	5.6	11.0
4.8	2.4	1.7	0.9	1.0	2.4	1.7	1.9	2.3	16.7	2.1	7.3	8.0	3.9	3.2	7.5	9.2	8.4
3.8	0.7	1.0	2.8	1.5	2.1	2.3	1.2	1.8	4.0	6.5	2.5	5.7	7.9	13.8			
4.5	1.4	2.7	0.9	2.0	2.8	2.2	1.2	1.9	5.4	1.8	7.9	10.6	4.8	2.3	9.0	8.1	2.8

(a) 座標 (b) オイラー角
 図4: 各ブロックにおける誤差