

単腕ロボットによる折り重ねられたタオルの分離と掴み上げ

○佐野 知樹, 出村 聡規, 中島 渉, 長濱 虎太郎 (信州大学),
竹下 佳佑 (トヨタ自動車㈱), 山崎 公俊 (信州大学)

Separation and Lifting up One of the Stacked Towels by a Single Arm Robot

○Kazuki SANO, Satonori DEMURA, Wataru NAKAJIMA, Kotaro NAGAHAMA(Shinshu Univ.),
Keisuke TAKESHITA(Toyota Motor Corp.), Kimitoshi YAMAZAKI(Shinshu Univ.)

Abstract: In this paper, we focus on the task of grasping one piece from towels folded and stacked. The important thing is to pick up it by grabbing the appropriate place without losing the shape state as much as possible. In order to realize this work using the human support robot HSR, we proposed a recognition method and operation procedure and verified the effectiveness through experiment.

1. はじめに

生活支援をおこなう自律型ロボットには、布製品を扱える能力を持たせることが望ましい。それにより、衣服を畳んだり、アイロンをかけたたり、台を拭くなどの支援作業を自動化することが期待できる。従来から、そのようなタスクの実現に向けた研究が進められており[1][2], 筆者らも、同様のモチベーションを持って研究を進めている。これまでの試みとして、一枚の折りたたまれた布製品を掴み上げる作業に着目し、その解決策を示してきた[3].

本稿では、折り畳まれて積み重ねられたタオルから1枚を掴み上げるタスクに着目する。特に、一番上に置かれたタオルの形状状態をなるべく崩さずに、適切な場所を掴んで持ち上げることを目標にする。このタスクを、トヨタ自動車㈱で製作されたヒューマンサポートロボット HSR[4]を用いておこなうための方法を検討する。HSR は、単腕型移動マニピュレータであり、一軸二指のエンドエフェクタを有するロボットである。このロボットにより掴み上げを実現するため、認識手法と動作手順を提案し、実機実験によって効果を検証する。

2. 折り重ねられたタオルの掴み上げ

2.1 掴み上げにおける課題

折り重ねられたタオルのうち一番上の一枚を崩さずに持ち上げるには、大きく二つの課題がある。1つ目は、布の重なりを考慮しなければならないことである、布製品を掴み上げる際、適切な位置を掴まなければ、重

ねられた布の形状状態は容易に崩れてしまう。よって、掴み上げた後に布製品の形状状態が大きく崩れないような把持箇所を選定が重要である。2つ目は、一枚のタオルだけを他のタオルから分離する動作についてである。例えば、上から2枚目のタオルも同時に掴んでしまうことや、逆に空掴みが生じること可能性がある。また、1枚のタオルを掴んで持ち上げる際に、その下のタオル群が動いてしまうことも、防ぐことが望ましい。以上より、把持したのちの掴み上げ軌道も考慮すべきである。

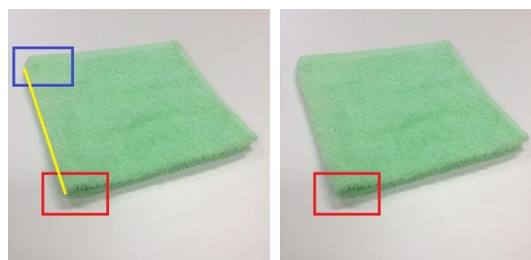


Fig. 1 目標把持箇所

2.2 掴み上げのアプローチ

前節の課題を踏まえて、タオルの掴み上げにおけるアプローチを以下に説明する。まずは、把持箇所を選定についてである。布製品は柔軟物であるため、作業中に形状状態が変わりうる。とはいえ、形状状態が大きく変わる事態は避けるべきである。掴み上げタスクにおいては、例えば、もともとの折り目が解消されて布がだらりと垂れ下がる事態は避けたい。そこで、掴み上げても形状状態が変わりにくい部位を探すと、Fig. 1左に示す赤四角と青四角の角や、黄線のふちが把持

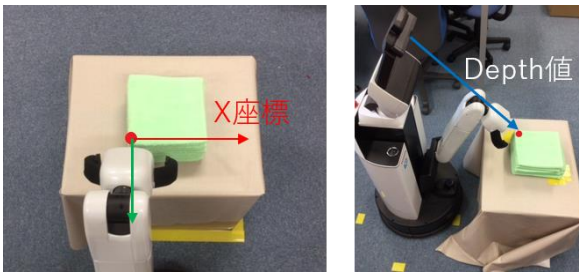


Fig. 2 ハンド座標系を作成する手法

位置として適切である。さらに、認識のしやすさや把持のしやすさの両観点から考えると、Fig. 1 右に示す赤四角が目標把持箇所として適切である。ここで、布製品のふちを把持箇所を選ばない理由は、把持動作の際に HSR のハンドを布製品の間に入り込ませることが困難だからである。一方、折り畳まれたタオルの角部分は、そのような滑り込ませる動作が比較的少ない。

次に、重ねて置かれたタオルのセンシングから掴み上げに至るまでの流れを説明する。把持位置の検出方法については次章で、掴み上げ方法については 4 章で詳細で述べるので、ここでは掴み上げに必要な認識処理について述べる。この処理の目的は、タオルのどこに把持すべき角が存在するかを見つけることである。まず、テーブル上に畳まれて重ねられたタオル群を、HSR 頭部に設置された三次元距離画像センサで計測する。そこで得られた三次元点群に対し、平面検出処理および平面除去処理を施し、タオル群の部分のみを検出する。次に、Fig. 2 内の赤丸に示すような角部の三次元位置を取得する。その後、Fig. 2 の赤丸に示した位置にタオル群の Z 軸まわりの回転を考慮したロボットハンドの目標座標系を作成し、その座標系を基準にして、把持位置の検出のための動作や掴み上げ動作を作成・実施することとした。また、把持位置の検出に失敗することも考えられるため、その対策として、Fig. 3 のような把持位置を検出しやすくなる動作などを取り入れる。



Fig. 3 把持位置の認識のための補助動作

3. 把持位置の検出

3.1 YOLO[5]

YOLO(You Only Look Once)とは、Deep learning を用いた一般物体検出アルゴリズムの一つである。現在、一般的に利用されている Faster R-CNN[6]などに比べ高速であり、かつ同等の精度を得ることができる。また、

構造が比較的単純であり、ImageNet を用いた事前学習結果が公開されているため、利用しやすいというのも特徴の一つである。

物体検出のニューラルネットワークとして、VGG16[7]がよく知られている。一方で YOLO では、Darknet19 と呼ばれる独自の特徴抽出器を利用している。Darknet19 は主に 22 層の Convolution 層と 5 層の Maxpool 層からできている。VGG16 と大きく異なるのは Fully Connected 層が存在しないため、特徴マップの正確な位置情報を保持したまま伝搬することが可能という点である。Darknet19 のおおまかな構造を Table. 1 に示す。

検出方法は、まず入力画像を縦横同数のセルに分割する。次に分割したセル毎にバウンディングボックスを予測する。その後バウンディングボックス毎に中心座標や大きさ、信頼度をそれぞれ保持し、それと同時にバウンディングボックスに含まれているものは何かも予測する。それらの結果をふまえ物体の検出や場所の特定をすることができる。

Table 1 Darknet19

| layer | Filters | Size / Stride | Output | layer | Filters | Size / Stride | Output |
|---------------|---------|---------------|---------|---------------|---------|---------------|--------|
| Convolutional | 32 | 3×3 | 224×224 | Maxpool | | 2×2 / 2 | 7×7 |
| Maxpool | | 2×2 / 2 | 112×112 | Convolutional | 1024 | 3×3 | 7×7 |
| Convolutional | 64 | 3×3 | 112×112 | Convolutional | 512 | 1×1 | 7×7 |
| Maxpool | | 2×2 / 2 | 56×56 | Convolutional | 1024 | 3×3 | 7×7 |
| Convolutional | 128 | 3×3 | 56×56 | Convolutional | 512 | 1×1 | 7×7 |
| Convolutional | 64 | 1×1 | 56×56 | Convolutional | 1024 | 3×3 | 7×7 |
| Convolutional | 128 | 3×3 | 56×56 | Convolutional | 1024 | 3×3 | 7×7 |
| Maxpool | | 2×2 / 2 | 28×28 | Convolutional | 1024 | 3×3 | 7×7 |
| Convolutional | 256 | 3×3 | 28×28 | Convolutional | 1024 | 3×3 | 7×7 |
| Convolutional | 128 | 1×1 | 28×28 | Convolutional | 30 | 1×1 | 7×7 |
| Convolutional | 256 | 3×3 | 28×28 | | | | |
| Maxpool | | 2×2 / 2 | 14×14 | | | | |
| Convolutional | 512 | 3×3 | 14×14 | | | | |
| Convolutional | 256 | 1×1 | 14×14 | | | | |
| Convolutional | 512 | 3×3 | 14×14 | | | | |
| Convolutional | 256 | 1×1 | 14×14 | | | | |
| Convolutional | 512 | 3×3 | 14×14 | | | | |

3.2 折り目の認識

本研究において検出すべき角部の例を Fig. 4 左の赤枠に示す。これは、折り畳まれた布の最も厚い折り目部分であり、ここを掴むことができれば、布製品の形状状態は大きく崩れることはない。そこで、この部位を把持対象角、その他の布の角部を非対象角として、入力画像からこれら 2 種類を検出することとした。YOLO のネットワークの重みを学習するため、把持対象角、非対象角の画像データをそれぞれ 130 枚、60 枚取得し、ImageNet の画像データで事前に学習された重みを初期値として、転移学習を行った。Fig. 5 は、HSR の手先についているカメラからの撮影画像に対して、認識をおこなった結果である。2 段目以降のタオルがどのような向きで積まれていても、おおそ正しく認識できることがわかった。ただし、把持対象角であっても、折り目部分にある丸みははっきり出ていない場

合に、正しく認識できないことがあった。この場合は、Fig. 3に示すように、ロボットが角部を押す操作を導入して、検出しやすい角形状を作ることにした。

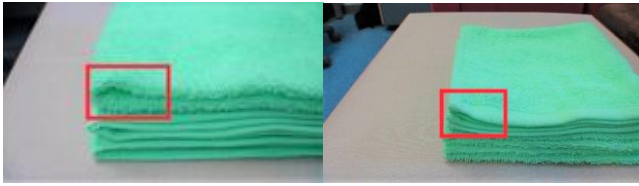


Fig. 4 学習データ (左：成功箇所, 右：失敗箇所)

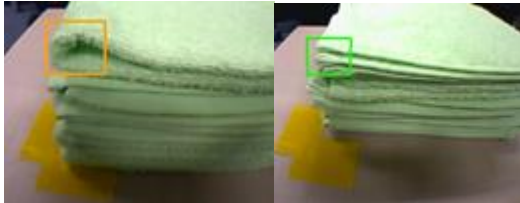


Fig. 5 認識結果 (左：成功箇所, 右：失敗箇所)

4. 掴みかたの検討

検出した角部を二指ハンドで掴むにあたり、二種類の掴み方を検討した。1つ目は、角から少し離れた場所を上から指先で押しつけ、把持箇所を浮かせてから把持する手法である。以後、これを押しつけ法と呼ぶ。Fig. 6に示すように、押しつけ法は柔軟物の変形しやすい特徴を生かした手法であり、浮かせることさえできれば把持が容易になるといったメリットがある。ただし、力の入れ具合や押しつける位置によって把持箇所の浮き方に違いがあるといったデメリットもある。2つ目の掴み方は、一番上のタオルと二番目のタオルの間に指先を入れ、タオルをすくい上げて把持する手法である。以後、この手法をすくい上げ法と呼ぶ。Fig. 7に示すように、すくい上げ法は人がモノを持つ動作に近い掴み方であり、比較的オーソドックスな手法である。押しつけ法と違い、力加減を考えなくてよいので動作が単純になると考えられる。ただし、タオルの下にハンドを入れる際、ハンドの位置が少しずれると把持したタオルの形が崩れてしまったり、二枚のタオルを同時に把持してしまう可能性がある。



Fig. 6 押しつけ法



Fig. 7 すくい上げ法

5. 実験

5.1 実験条件

Fig. 8に示すような矩形のタオル 6 枚を、Fig. 1のように 4 つ折りにして水平な台の上に重ねて置いた。タオルの重ねる際、4 つ折りにしたときにできる Fig. 1 赤四角、青四角および黄線の部分が、同じ面に来るようにした。そのタオル群が HSR 頭部に設置された三次元距離画像センサに映るように、HSR の首を予め傾けた。

タオル群の置き方は、Fig. 9を 0 deg として ± 40 deg の範囲内で、ランダムに置いた。HSR のアームの可動範囲の制約から、把持対象角の位置はロボットから向かって左側にあるものとした。



Fig. 8 使用したタオル (35cm x 35cm)

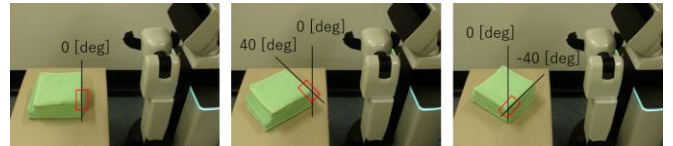


Fig. 9 タオル群の置き方

5.2 実験結果

認識処理や動きの流れは次のようであった。まず、台の上に置かれた布製品を HSR 頭部に設置された三次元距離画像センサで観測し、Fig. 9のようなタオル群の Z 軸まわりの回転量を測定する。次に、Fig. 2に示した赤丸の位置を三次元点群から取得し、HSR の台車に固定された座標系からの座標に変換した。そして、その位置にロボットハンドの目標座標系を作成し、その座標系を基準にロボットハンドを Fig. 2のように近づけ、手首に埋め込まれたハンドカメラを用いて Fig. 5のように把持対象角を検出した。検出できた場合は、4章に示した把持動作を行い、検出できなかった場合は、Fig. 3のような補助動作を加えたのちに再び検出処理をおこない、検出できれば把持動作へ移行した。

Table.2 は、角を生成するための補助動作を入れた場合の、二種類の掴みかたによる実験結果である。掴み

上げの成功率は、押しつけ法が 86.7%，すくい上げ法が 60.0%であった。

Table.2 2種類の掴みかたによる実験結果

| | 実験回数 | 成功回数 | 成功率 |
|--------|------|------|-------|
| 押しつけ法 | 15 | 13 | 86.7% |
| すくい上げ法 | 15 | 9 | 60.0% |

Table.3 押しつけ法での実験結果と手法改善

| | 角の作成動作 | 実験回数 | 成功回数 | 成功率 |
|--------|--------|------|------|-------|
| 角の検出可 | | 15 | 11 | 73.3% |
| 角の検出不可 | 改善前 | 15 | 4 | 26.7% |
| | 改善後 | 15 | 13 | 86.7% |

5.3 考察

Table. 2 に示したように、掴みあげ方法に関しては、押しつけ法の方が高い成功率が得られた。すくい上げ法は、把持後にタオルの形状が崩れてしまうことが多かった。

Table. 3 には、押しつけ法での把持実験の結果と、角を生成するための補助動作の改善の効果を示した。まず、角の検出が成功した場合の掴み上げの成功率は 73.3% であった。また、角の検出ができなかった場合は、角を認識しやすくするための補助動作をおこなったが、Fig. 1 0 に示した当初の動作では、把持の成功率が低かった。そこで、Fig. 1 1 のような動作へ変更した。Fig. 1 0 の改善前はタオルを押す際にハンドの形状を考慮せずに角を押していたため、タオルの間にハンドは滑り込んでしまい、適切に角を作れないことが多かった。しかし、Fig. 1 1 のように、ハンドの傾きをおさえてタオルのふちを押すようにした結果、角の作成の成功率が向上し、その後の把持動作の成功率の向上にもつながった。

その他の失敗要因としては、角の認識の際に、失敗箇所と認識されてしまったり、Fig. 1 2 の右に示すように、把持後にタオルの形状が崩れてしまうことがあった。

6. おわりに

本稿では、折り畳まれて積み重ねられたタオル群から 1 枚を掴み上げるタスクに着目し、単腕移動マニピュレータによってその作業を実現する方法について述べた。今後は、より高速・高精度な掴み上げを目指して、改良をおこなっていく。

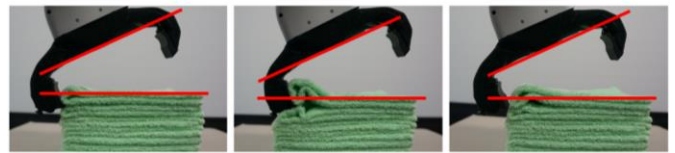


Fig. 1 0 改善前の角を作る動作

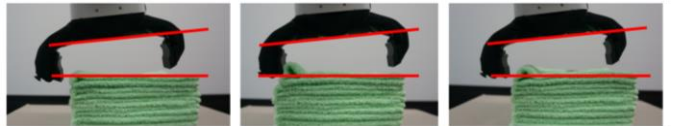


Fig. 1 1 改善後の角を作る動作



Fig. 1 2 把持後のタオルの形状 (左: 成功, 右: 失敗)

参考文献

- [1] J. Maitin-Sp Shepard et al.: "Cloth Grasp Point Detection based on Multiple-View Geometric Cues with Application to Robotic Towel Folding," Int'l. Conf. on Robotics and Automation, pp.2308 – 2315, 2010.
- [2] A. Doumanoglou, A. Kargakos, T-K. Kim, S. Malassiotis: "Autonomous Active Recognition and Unfolding of Clothes using Random Decision Forests and Probabilistic Planning," in Proc. of IEEE ICRA, 2014.
- [3] 守屋, 田中, 山崎, 竹下: 「単腕マニピュレータによる折りたたまれた布製品の適切な掴み上げ」, ロボティクス・メカトロニクス講演会 2017, 講演番号 2P2-G08, 2017.
- [4] トヨタ自動車 ヒューマンサポートロボット HSR: http://www.toyota.co.jp/jpn/tech/partner_robot/
- [5] Joseph Redmon, Ali Farhadi: "YOLO9000: Better, Faster, Stronger," arXiv:1612.08242, 2016.
- [6] R. Shaoqing et al.: "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems, pp. 91-99, 2015.
- [7] K. Simonyan, A. Zisserman: "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv technical report, 2014.