

家具製造現場におけるトレーサビリティ基盤技術の開発（第2報） OpenCV.jsを用いた画像認識による棚卸支援Webアプリケーションの開発

森茂智彦*, 藤巻吾朗*

Basic Technology Development of Traceability in the Furniture Manufacturing Field (II) Development of Inventory Support Web Application by Image Recognition Using OpenCV.js

MORIMO Tomohiko*, FUJIMAKI Goroh*

木製家具製造業における部材の在庫数のカウントを支援するWebアプリケーションを開発した。スマートフォンやタブレット上で部材棚の写真を撮影し、画像上の部材をタップすることで個数をカウントする。印が表示されるため、カウントミスを防止できる。また、タップの労力を減らすため、OpenCV.jsを用いて、画像認識により部材を検出し自動でカウントする機能を追加した。結果、一部の部材で労力の低減が図れたが、部材の径によっては、誤検出の修正による労力の方が大きい場合があるため、パラメータを調整する必要がある。

1. 緒言

棚卸は、会社の利益の把握や在庫管理に必要な作業である。製造業の棚卸では、組立後の製品だけでなく、組立前の部材の個数を把握する必要がある。中でも木製家具製造業は、多品種少量生産であり、さらに自社製品、OEM製品など多くの種類の製品を製造しているため、棚卸作業に多くの時間を費やしている。特に、丸棒のような部材は、図1のように、棚の中に規則的に並んでいるとは限らないため、数えている最中に、数え終えたものとまだ数えていないものの区別が分からなくなったり、丸棒が転がり配置が変わることでやり直しとなったりすることがあるため、効率良く在庫数をカウントできることが求められている。

在庫管理に関する研究として、Webアプリケーションを用いた例がある¹⁾。Webアプリケーションは、端末に依存せず利用可能であり、運用後のソフトウェアの変更が容易であるという特長がある。特に、棚卸のような複数人で行う作業の場合、各自のスマートフォンやタブレットで利用できることは、コストをかけず、使い慣れた端末で操作できるため、その効果は大きい。そこで本研究では、棚卸時の労力を減らすため、部材の在庫数のカウントを支援するWebアプリケーションを開発したので結果を報告

する。



図1 不規則に並ぶ丸棒の例

2. カウント支援アプリケーション

2.1 概要

開発したWebアプリケーションの画面を図2に示す。スマートフォンやタブレットで部材棚の写真を撮影し、写真上の部材をタップする。タップしたところは印が表示される。同時に、タップした回数がカウントされるため、全ての部材をタップすることにより、在庫数を把握することができる。

予め撮影した写真を読み込んでカウントすることもできる。この場合は、部材棚の写真をもとめて撮影しておき、後からカウント作業のみを行うといった使い方が想定される。

* 試験研究部

間違えてタップした場合は、戻るボタンを押して、タップする前の状態に戻すか、消したい印をタップした後、消すボタンを押す。同時に、カウントした個数は、1減算される。または、印をタップすると動かせるため、正しい位置に移動させる。

本アプリケーションを使用することで、部材に印が表示されることによりカウントミスを防ぐことができる。また、カウント中に部材の配置が変わることがなく、作業者は部材をタップすることに意識を集中すれば良いため、労力を減らすことができる。

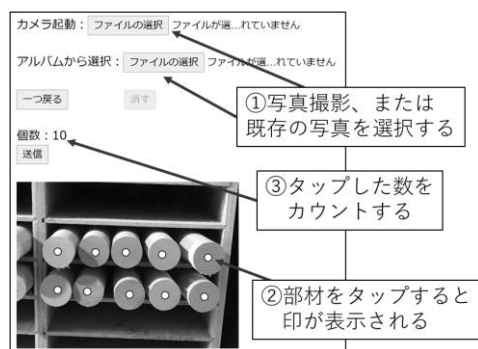


図2 Webアプリケーション画面

2.2 Webアプリケーションの実装

Webアプリケーションは、HTMLとJavaScriptを用いて作成した。

画像の読み込みと印の表示には、Fabric.js²⁾を用いた。Fabric.jsは、HTML5のCanvas要素上で図形の描画・サイズ変更・削除などの操作を行うことができるJavaScriptライブラリである。読み込んだ画像はCanvas要素の背景画像に設定し、タップ時に追記する印を描画するようにした。印の表示には、印の中心がタップした位置となるよう、下記位置を印の中心とした。

```
left:event.pageX - $("#canvas").offset().left-radius
top:event.pageY - $("#canvas").offset().top-radius
#canvas : canvas要素のid
radius : 赤丸の半径
```

2.3 評価と考察

作成したアプリケーションの評価を実施した。企業の棚卸担当者にアプリケーションを用いて部材数のカウントを行ってもらい、意見を聞いた。結果を下記に示す。

- ・部材数が多くなるとカウントに労力を要する
- ・個数が少ないものは、手で数えた方が早い
- ・写真を撮影しておけば別室でカウント作業ができる点は良い

これらの結果から、個数が多いものを対象に、さらに労力を減らすため、カウント時のタップ数を減らすことが必要と考える。

3. 画像認識機能

前項の評価結果を受け、画像認識技術により、自動で部材を認識しカウントする機能の追加を行った。

3.1 方針

家具製造業では、椅子の貫部材などのように、1脚で複数本使用する部材は、相対的に在庫数が多くなる。木製家具製造業者から聞き取りしたところ、中でもろくろで加工した部材が特に多く、まずはこの部材を対象とする。

図1はろくろ加工品である。家具用途において、ろくろで加工する部材は端までろくろ加工することがほとんどであり、それにより端面が丸型となる。そこで、本研究では、この端面の丸形状を認識し、カウントすることとする。

3.2 画像認識機能の実装

画像認識には、OpenCV.js³⁾を用いる。OpenCV (Open Source Computer Vision Library) は、オープンソースの画像処理ライブラリであり、OpenCV.jsは、OpenCVのJavaScript版である。

本研究では、部材端面の丸形状を認識するため、Hough変換を用いて円を検出するHoughCircles関数を用いる⁴⁾。この関数はグレースケール画像から円を検出するため、撮影後の画像をグレースケール画像に変換する必要がある。なお、OpenCV.jsのバージョンは、4.5.1を用いた。

3.3 評価と考察

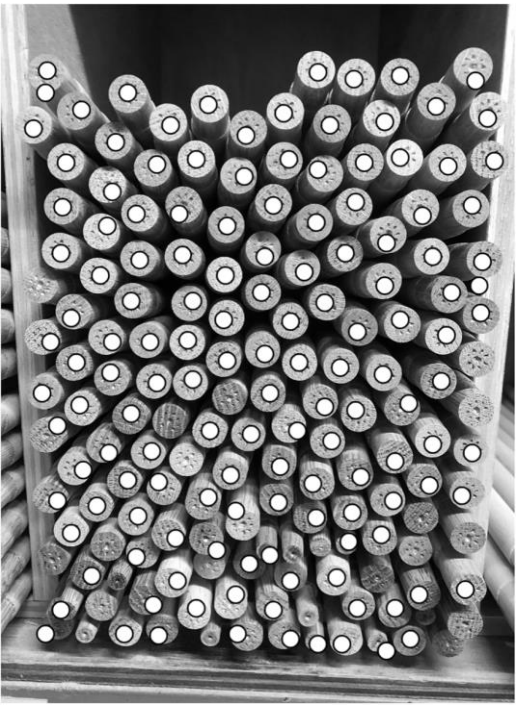
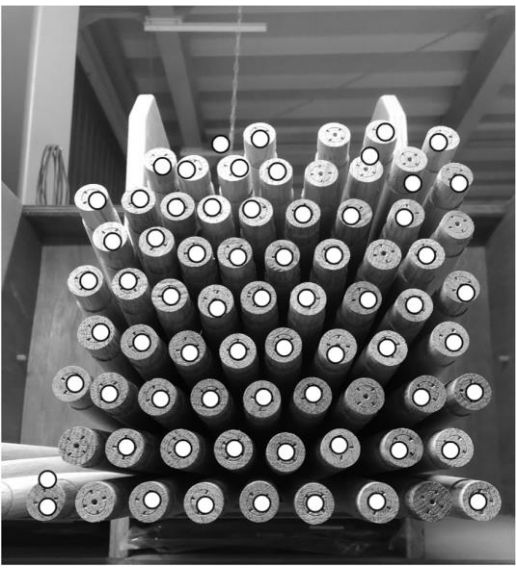
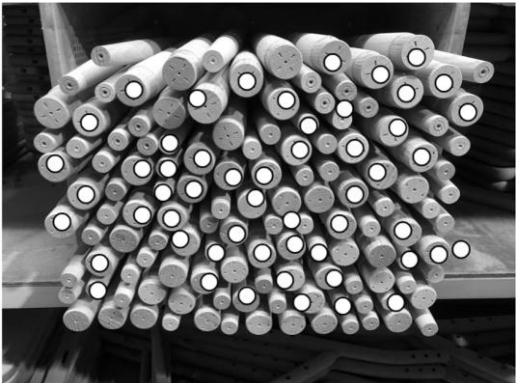
3.3.1 検出条件と結果

6種類の部材棚の写真撮影し、画像認識を用いて部材の検出を行った。検出に用いるHoughCircles関数の引数（パラメータ設定）を表1に、部材棚の写真とその検出結果を表2に示す。

表1 引数の設定値

引数名	設定値
method	CV_HOUGH_GRADIENT
dp	1
minDist	13
param1	80
param2	20
minRadius	5
maxRadius	20

表2 部材棚の写真と検出結果

No	写真	検出結果
1		
2		
3		



3.3.2 精度評価

表2の結果から、カウントの精度を式 (1) (2) によって評価する。

$$\text{再現率} = \frac{\text{検出正解数}}{\text{正解総数 (部材数)}} = 1 - \text{未検出率} \quad (1)$$

$$\text{適合率} = \frac{\text{検出正解数}}{\text{検出総数}} = 1 - \text{誤検出率} \quad (2)$$

精度評価結果を表3に示す。No.1,2,4については、再現率・適合率ともに85%以上となり、高い検出精度が得られた。No.3については、径が小さい部材が多く、撮影角度が部材に対し水平でないため、丸と判断されず、再現率が低くなったと考える。No.5,6については、径が大きいため、撮影角度が傾いていても再現率は高くなったが、1個の部材につき2個検出されることがあった。このことから、撮影角度をできるだけ水平にし、径によってHoughCircles関数のパラメータを変える必要があると考える。

表3 精度評価結果

No	1	2	3	4	5	6
正解総数	164	72	130	96	22	24
検出総数	143	66	54	89	42	40
検出正解数	141	63	52	84	22	21
再現率	86.0%	87.5%	40.0%	87.5%	100.0%	87.5%
適合率	98.6%	95.5%	96.3%	94.4%	52.4%	52.5%

3.3.3 考察

画像認識機能を用いることにより、労力がどの程度低減したかを考える。

画像認識によるカウント後、正しい部材数を得るため、未検出の部材に印を付ける作業と、誤検出した印を消す作業が必要となる。この修正作業による労力と、画像認識機能を使用せずカウントした場合の労力を比較する。今回は、画面をタップする回数が多いほど労力が大きいため、それぞれの場合での総タップ数を比較する。

画像認識機能ありの場合は、未検出の部材に印を付ける場合を1タップ、誤検出した印を消す場合を2タップ（選択、消去）とし、修正に必要な総タップ数を計算する。

画像認識機能なしの場合は、全ての部材をタップして印を付けるため、カウントに必要な総タップ数は、部材数と同じである。

総タップ数を計算し比較した結果を表4に示す。No.1,2,4については、総タップ数の減少率が高く、労力の低減効果が高い結果となった。No.3については、効果は低いが労力の低減は見られた。No.5,6に

ついては、画像認識なしの場合よりも画像認識ありの場合の方が総タップ数は多い結果となり、労力の低減は見られなかった。つまり、表3で再現率が最も低いNo.3よりも、No.5,6の方が労力の低減効果は低い。これは、未検出した場合より誤検出した場合の方が修正のためのタップ数が多いためである。よって、さらなる労力の低減のためには、まず誤検出数を減らすこと、つまり適合率を高めることが求められると考える。

表4 総タップ数の比較結果

No		1	2	3	4	5	6
画像認識 なし	総タップ数 (正解総数)	164	72	130	96	22	24
画像認識 あり	未検出数	23	9	78	12	0	3
	誤検出数	2	3	2	5	20	19
	総タップ数	27	15	82	22	40	41
総タップ数の差		137	57	48	74	-18	-17
総タップ数の減少率		83.5%	79.2%	36.9%	77.1%	-81.8%	-70.8%

4. その他の追加した機能

今回開発したWebアプリケーションで、実装した他の機能を紹介する。

4.1 サーバーへ送信する機能

送信ボタンを押すと、部品番号や型式とカウントした個数をサーバーへ送信する機能である。画面に部品番号・型式入力欄と送信ボタンを追加し、送信部分はPHPで作成した。用途として、サーバー側で在庫数を自動で取り纏める場合が想定される。

4.2 QRコードを読み込む機能

部材の型式をQRコードで読み込む機能である。部材棚に予め部材の部品番号や型式を記録したQRコードを貼り付けておく。棚卸時に、このQRコードを端末のカメラで読み込むことで、4.1項の部品番号欄・型式欄に値が自動で入力される。

QRコードの読み込みには、jsQR.jsを用いた⁵⁾。このライブラリでは、カメラでQRコードを読み込む際に、サーバーがSSL化されている必要がある。

5. まとめ

木製家具製造業における在庫数のカウントを支援するWebアプリケーションを開発した。部材棚の写真撮影し、画像上の部材を順にタップして印を付けることで、そのタップした回数をカウントする。

部材数が多い場合のタップの労力を低減するため、OpenCV.jsによるHough変換を用いた画像認識により、端面が丸形状の部材の個数を自動でカウントする機能を実装した。結果、部材の径によっては良好な結果が得られた。

今後は、部材の径によって画像認識のパラメータを調整する必要がある。また、使いやすさの評価を行っていきたい。

謝辞

研究のきっかけを作ってください、同時に多くのご意見をいただきました飛騨産業株式会社の関係者の方々、画像認識に関する有益なご助言をいただきました岐阜県産業技術総合センター研究員の生駒晃大氏にこの場を借りて厚く御礼申し上げます。

参考文献

- 1) 菅野幸貴ら：産地直売所における在庫管理支援のためのwebアプリケーションの開発，情報処理学会第 70 回全国大会講演論文集，Vol.70, No.3, pp.151-152, 2008.
- 2) Fabric.js Javascript Canvas Library：
<http://fabricjs.com/>, 2021 年 2 月 4 日参照.
- 3) OpenCV：OpenCV.js Tutorials,
https://docs.opencv.org/4.5.1/d5/d10/tutorial_js_rot.html, 2021 年 2 月 10 日参照.
- 4) OpenCV：Hough Circle Transform,
https://docs.opencv.org/4.5.1/d3/de5/tutorial_js_houghcircles.html, 2021 年 2 月 10 日参照.
- 5) jsQR.js：<https://github.com/cozmo/jsQR>, 2021 年 2 月 10 日参照.