

# 古い機械をIoT化する光学的読み取り技術の開発

新田優樹\* 米沢裕司\* 田村陽一\*

電氣的に機械の状態を取得するのが難しい機械をIoT化するため、カメラで機械の表示を撮影して光学的に情報を読み取る技術が求められている。本研究では、アナログメータの指示値、7セグメント表示器の数値、ランプ表示の点灯状況の3種の表示を対象とし、それらをカメラで撮影してデジタルデータ化する技術の開発を目指した。アナログメータは針の角度から指示値を読み取り、7セグメント表示器はOCRによって数値を読み取り、ランプ表示は輝度を閾値処理することで点灯と消灯を読み取り、それぞれデジタルデータ化できることを確認した。結果、電氣的に状態を取得できない機械も表示を読み取ってデジタルデータ化することで容易にIoT化する技術を開発できた。

キーワード: IoT, カメラ, アナログメータ, OCR

## Development of Optical Reading Technology to Convert Old Machines into IoT Devices

Yuki NITTA, Yuji YONEZAWA and Youichi TAMURA

In order to transform machines whose state is difficult to obtain electrically into IoT devices, there is a need for a technology to read information optically by capturing the display of the machine using a camera. In this research, we aimed to develop a technology to photograph such data using a camera and convert it into digital data, for three types of display: an analog meter (needle), a 7-segment numerical display, and a lamp (illumination status). The analog meter value was read from the angle of its needle, the 7-segment numerical display was read by OCR, and the lamp ON/OFF state via a luminance threshold. It was confirmed that each can be converted to digital data. As a result, we were able to develop a technology that allows the easy transformation of machines whose state is difficult to obtain electrically into IoT devices by reading their displays and converting the output to digital data.

Keywords: IoT, camera, analog meter, OCR

### 1. 緒言

製造業においては、IoTによる見える化を行いたいというニーズが高まっている。産業機械にもIoT対応を謳うものが出てきているほか、既存の機械にセンサを設置することでIoT化するものも製品化されている。しかし、取り付けたセンサによって機械の不具合を誘発するリスクや、そもそもセンサの取り付けが困難な機械も存在する。そこで機械に手を加えることなくIoT化する技術として、メータ等の表示器をカメラで撮影し、その表示内容を読み取ってデジタルデータに変換する技術の開発を行った。同様の技術を用いたものが既に製品化されているが、いずれも高額な初期費用やライセンス費用が必要となる。そこで本研究では、低コストでカスタマイズ性の高い読み取り技術の開発

を目指し、安価なハードウェアとオープンソースソフトウェア(OSS)を用いて、アナログメータの指示値、7セグメント表示器の数値、ランプ表示の点灯状況の3種の表示を読み取る技術を開発した。

### 2. 読み取りアルゴリズム

#### 2.1 アナログメータの読み取り

##### 2.1.1 読み取り手法の概要

本研究では図1に示すような目盛りが等間隔のアナログメータを対象に針の指示値を数値化する。具体的にはメータの画像に扇形を描画することで針の回転軸を設定する。これにより図中の針の角度(回転角) $a$ を求めれば、指示値 $n_{\text{indicate}}$ は式(1)により計算できる。

$$n_{\text{indicate}} = \frac{a}{a_{\text{max}}} \times (n_{\text{max}} - n_{\text{min}}) + n_{\text{min}} \quad (1)$$

ここで、 $n_{\text{min}}$ 、 $n_{\text{max}}$ は目盛りの最小値、最大値、 $a_{\text{max}}$

\*電子情報部

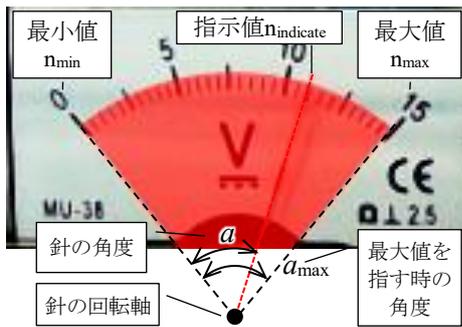


図1 角度と回転軸の設定

は最大値を指す時の角度とする。

### 2. 1. 2 針の検出

この項では、画像から針を検出してその角度を求める方法を述べる。メータ画像はあらかじめ背景を黒、針を白として二値化してあるものとする(図2)。

一般的なメータの針は直線であるため、ハフ変換によって画像中の直線を検出する。この時、針の長さに対して著しく短い線を除外することで、目盛り線等のノイズを除外する。メータ枠など線の長さでは除外できない線については、針の回転軸からの距離を用いて針と識別する。針はその回転軸から伸びているため、検出された線を伸ばした時に回転軸の最も近くを通る線が針であると推測できる。

### 2. 1. 3 アナログメータ読み取りの評価

このアルゴリズムの読み取りの正確性を確認するため、当試験場の冷熱衝撃試験機に設置された温度計(図3)で実験を行った。この温度計は試験機に供給される冷却水の経路上に設置されている。冷却水の温度は試験機内部のセンサでもモニタリングされており、試験機よりリアルタイムのデジタルデータを取得できる。1417枚の撮影画像から読み取った結果を同時刻の試験機のセンサ値と比較したところ、その差は1℃以下であった。これは針の角度に直すと2.7°以下となる。

## 2. 2 7セグメント表示器の読み取り

### 2. 2. 1 読み取り手法の概要

7セグメント表示器の読み取りには、OSSのOCR(光学的文字認識)ソフトウェアである Tesselact-OCR<sup>1)</sup>を採用した。OSSの特徴として無料でソフトウェアが使用できるほか、改良も自由に行うことができる。通常のOCRソフトウェアでは Tesselact-OCRも含め、7セグメント表示器のような特殊な書体を認識できない。その

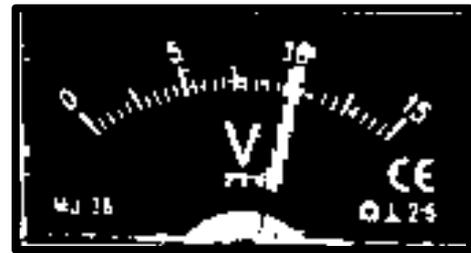


図2 メータの2値化画像

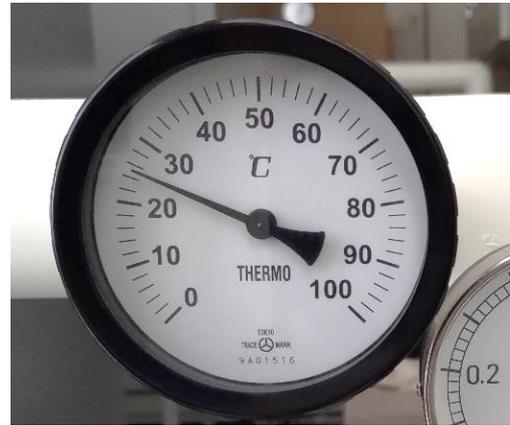


図3 冷熱衝撃試験機の温度計

ため7セグメント形式の数字を学習させることで、7セグメント表示器の読み取りに特化したOCRソフトウェアにした。

### 2. 2. 2 7セグメント表示の学習

Tesselact-OCRの文字認識には機械学習の技術が使われており、学習済みの言語データによって文字の認識が行われる。7セグメント表示器の数値を読み取るためには、7セグメント表示の数字を学習した言語データが必要である。そこで学習用の訓練画像として、最大3桁の7セグ表示画像を103680枚用意した。この画像は7セグフォントのDSEG7<sup>2)</sup>で描画した文字に、ノイズやボカシを加えて作成した。言語データの作成にはゼロから学習を行う方法と、既存の言語データを基にした転移学習によって新しい書体を学習する方法がある。本研究では、新規に作成した言語データと英語の言語データをベースにした言語データを作成し、訓練画像の一部を用いてその正答率を確認した。その結果、英語の言語データをベースに作成したものでは98%の正答率となったが、ゼロから学習した言語データでは正答率が0%となった。この結果から、既存の言語データをベースにする場合には10万枚程度の画像で学習できるが、ゼロから学習させる場合には全く足りないことが明らかとなった。

### 2. 2. 3 7セグ表示読み取りの評価

上記では訓練画像の一部を用いて正答率を求めたが、実環境での正確性を確認するために温度ロガー(図4)を用いた読み取り実験を行った。温度ロガーの液晶表示を1分間隔で24時間撮影して得た1440枚の画像を読み取って正答率を求めたところ、正答率は93%であった。

### 2. 3 ランプ点灯状況の読み取り

ランプの点灯状況は、ランプ部分の輝度を閾値処理することによって点灯の有無を判断する。ランプの輝度に加えて色相及びその彩度にも閾値を設定したプログラムも試作したが、点灯を正しく検知できなかった。これは点灯しているランプをカメラで撮影した場合、ランプの中央付近では赤、緑、青の各値が飽和し、白色となる場合があるためである。一方で、輝度のみで判定した場合は点灯と消灯を正しく判定できた。

## 3. 撮影画像の前処理

### 3. 1 読み取り対象領域の切り出し

撮影画像に読み取りたい表示以外のものが映っていると、それらがノイズとなり認識精度が悪化する。そのため、撮影画像からメータや数値、ランプなど、読み取りの対象となる領域を正確に切り出すプログラムを開発した。単純に画像の座標で切り出す位置を指定すると、カメラや読み取り対象の位置にズレが生じた場合に正しく切り出すことができない。そこで、切り出す領域を正確に特定するために、特徴点マッチングを用いた画像の補正を行った。これは画像から特徴的な場所(=特徴点)を検出し、あらかじめ撮影した画像の特徴点(図5)と突き合わせることで画像の位置ずれや傾きを補正する手法である。補正を行うことで、カメラ等の位置ずれが生じても正確に表示部分のみを切り出すことが可能となった。

### 3. 2 色に関する前処理

色に関しては、最終的に読み取りたい形状を持った白黒の二値画像を得ることを目的とする。アナログメータの読み取りであれば、黒地にメータの針のみが白色で描画されている画像が理想である。そのため、下記の処理プログラムを開発した。

- ・ 特定の色をマスク
- ・ グレースケール画像に変換
- ・ 二値化



図4 読み取り対象の温度ロガー



図5 撮影画像内の特徴点

特定の色をマスクする処理は、あらかじめ指定した色範囲に該当する画素を、黒くマスキングする処理である。例えば赤色の針を読み取りたい場合、それ以外の色を黒く塗りつぶすことで、読み取る対象のみを画像内に残すことができ、認識精度の向上につながる。

グレースケール画像は、色の情報を持たない明るさのみの画像である。グレースケール画像を閾値処理することで、白黒の二値画像が得られる。この時の閾値については、全ての画素に同じ閾値を適用する単純な処理と、周囲の画素から最適な閾値を計算する適応的閾値処理のアルゴリズムに対応している。

### 3. 4 ノイズ除去に関する前処理

画像内にノイズが存在すると、読み取りを阻害する要因になる。そこで、ノイズを除去する方法として、ぼかし処理と、膨張・収縮処理のプログラムを開発した。ぼかし処理は、二値化前のグレースケール画像に対して適用することで、ノイズの少ない二値化画像を得ることができる。また膨張・収縮処理は、二値化後に残る細かいノイズの除去に効果を発揮する。収縮処理では黒い画素に隣接する画素を黒くし、膨張処理では白い画素に隣接する画素を白くする。膨張と収縮を

セットで行うことにより、元の図形を保ったまま細かなノイズを除去することができる（図6）。

## 4. 設定方法

### 4.1 設定ソフトウェアについて

前述の前処理や読み取り処理は、それぞれが独立したクラスとしてカプセル化されている。これらのクラスを組み合わせることで読み取り処理を定義する。その際、読み取る対象によって、その組み合わせやそれぞれの処理のパラメータをカスタマイズする必要がある。このカスタマイズを行うための設定ソフトウェアを開発し、プログラミングや画像処理に精通していなくても、容易にカスタマイズできるようにした。またWeb技術で開発することにより、専用の機材やソフトウェアを使わずに、Webブラウザを持つタブレット端末やPCで容易に設定できるようにした（図7）。

### 4.2 設定ファイル

設定ファイルはJSON形式で保存する。JSONはプログラミング言語に依存しないテキストベースのデータ交換フォーマットである(JIS X 3061)。この規格では有効なJSONテキストの構文のみが定義されているため、設定ファイルのデータ構造については、表1のように定めた。設定ファイルはその名称と処理の配列を持ち、配列の先頭にある処理から順に実行される。処理は、その処理を行うクラス名と、クラスをインスタンス化する際のパラメータで構成される。

## 5. 結 言

電氣的に状態を取得できない機械も、表示を読み取ってデジタルデータ化することで、容易にIoT化できる技術を開発した。機械に直接手を加えないため、不具合を誘発するリスクも少ない。必要となるハードウェアも、安価なカメラと小型コンピュータだけであり、コストをかけずにIoT化を進めることができる。

### 参考文献

- 1) Ray Smith (lead developer). “GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository)”. GitHub <https://github.com/tesseract-ocr/tesseract>, (参照 2022-08-01).
- 2) けしかん. “7セグ・14セグフォント「DSEG」”. けしのみ工房-keshikan.net. <https://www.keshikan.net/fonts.html>, (参照 2022-08-01).

表1 設定ファイルのデータ構造

項目	データ型	概要
name	文字列	設定の名称
process	配列	実行する処理を順に記載する
<<処理>>	オブジェクト	
class	文字列	処理を実行するクラスのクラス名
setting	オブジェクト	インスタンス生成時のパラメータ

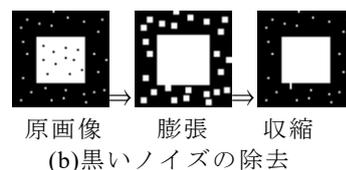
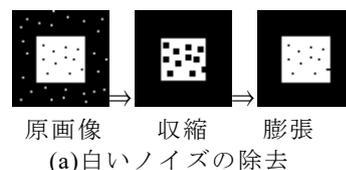


図6 膨張・収縮によるノイズ除去



図7 設定ソフトウェアの概要