

修 士 論 文

撮影条件の違いに頑健な
ドットマトリクス文字の
抽出と認識に関する研究

平成 27 年度修了

三重大学大学院工学研究科

博士前期課程 情報工学専攻

鈴木 鴻志

はじめに

商工業において、大量の製品を管理することは重要な工程であり、多くの製品に管理用の固有番号、製造番号、消費期限等が印字されている。これらの情報は、利用者（顧客、製造所の作業員）が目視により記載された内容を理解できる必要があるため、人間が見て理解できる「文字」で表示される必要がある。特に薬品や食品においては、消費期限や製造所などの重要な情報を製品そのものに直接示すことが法令により定められている。現在、管理工程において文字の目視確認と手動でのデータ入力が行われており、人件費や所要時間等の観点から非効率的であるため、自動化が期待されている。OCR（Optical Character Recognition）によって製品を撮影した画像から文字を自動認識すれば、生産工程に複数の新規要素（コード化、コードの印字、コードの読み取り）のための新たなコストを生じさせることなく、製造段階や流通段階において製品の管理を支援することができる。管理用の情報は印字コストの低さやメンテナンスの容易さから、ドットインパクト方式のプリンタによって印字される。このため、製品管理用 OCR はドットマトリクスによって構成された文字（ドット文字）を認識する必要がある。

一般的に、文字以外の様々な背景を含む画像から文字を認識するためには、画像中から文字が書かれている場所を特定する抽出プロセスと、抽出した文字が何であるかを判定する認識プロセスが必要である。現在、高い精度で画像から文字を抽出する手法は、文字色の連続性を用いた手法（MSER）や、文字幅が一定であることを仮定した手法（SWT）である。しかし、ドット文字は複数のドットマトリクスで構成されるため、同じ手法をそのまま利用することは不可能である。また、ドット文字は印字される面の素材や状態が様々であり反射や射影の条件が異なることや、印字する時に文字の崩れ、かすれ、回転、位置ずれが発生しやすく、自動認識が困難である。これらの課題に対して、ドット文字列の抽出手法やドット文字認識手法が提案されているが、いずれも OCR システムの一部であるため、画像を入力して文字を認識する End-to-End の手法が必要である。さらに、現在普及している文字認識装置の多くは、事前に認識対象の文字形状、印字位置、光源、素材を細かく登録するマッチング方式によるものであり、ユーザビリティを改善する余地がある。

本研究では、ドット文字が表面に印字された製品の画像を入力し、ユーザによる事前の情報登録なしでドット文字のみを抽出して認識する手法を提案する。本手法は以下のステップからなる。

1. ドット文字列の抽出

(a) ドット文字候補点抽出

入力画像から Features from Accelerated Segment Test (FAST) コーナーを抽出し、こう配方向、こう配強度、彩度からドット文字上に存在する FAST コーナーを選別する。

(b) ドット文字列候補領域抽出

ドット文字候補点を、近傍画素の輝度を用いてクラスタリングすることでドット文字列候補領域を抽出する。

2. ドット文字の認識

(a) 前処理（二値化，行間検出，膨張処理）

抽出した文字列候補領域を二値化し、行間の位置を検出する。膨張処理を適用してドットを結合し、連結成分を抽出した後、領域の大きさと位置関係を用いて文字列候補領域から 1 文字の候補領域を抽出する。

(b) ドット文字の抽出と認識

ドット文字の変動モデル学習によって作成された学習辞書と Modified Quadratic Discriminant Function (MQDF) 識別器を用いて、文字候補領域を認識し、文字認識評価値（MQDF 評価値）を算出する。

3. 非ドット文字の棄却

MQDF 評価値と抽出された文字候補領域の位置関係を用いて非ドット文字領域を棄却する。

提案手法の性能を検証するため、画像 250 枚を用いて性能評価実験を行った。実験の結果、文字抽出の再現率は 78.60%，精度は 76.03%，抽出された文字の認識率は 94.43% であり、汎用 OCR システムである Tesseract-OCR と比較して性能が高いことが確認された。250 枚から 100 枚の画像をランダムに選出して評価実験を 50 回繰り返した結果、F 値について 3.63% 以上の性能低下は無く、本手法は様々な撮影対象に適用可能である。さらに、実応用において高速で流れる生産ライン上での使用によるモーションブレイや、撮影時の位置ずれによる文字列の回転が発生することを想定し実験した結果、本手法はそれらの変化に対して性能が安定していた。

本論文では、1 章で研究の背景と目的、2 章でドット文字認識に関する手法、3 章で提案手法、4 章で検証実験、5 章で結果の考察、6 章で全体のまとめについて述べる。

目次

はじめに	i
第 1 章 序論	1
1.1 研究背景	1
1.1.1 産業の支援	1
1.1.2 ドットマトリクス文字	2
1.2 OCR 共通の問題点	3
1.2.1 文字の抽出	3
1.2.2 文字の認識	3
1.3 ドット文字用 OCR の問題点	4
1.3.1 ドット文字の抽出	4
1.3.2 ドット文字の認識	4
1.3.3 製品管理用 OCR	5
1.4 本研究の取り組み	6
1.4.1 先行研究	6
1.4.2 研究目的	6
第 2 章 文字認識手法	7
2.1 学習辞書作成	7
2.1.1 変動モデル学習	7
2.1.2 複数のドット文字フォント	8
2.1.3 3 次元回転文字	10
2.1.4 ドット直径の変更	11
2.1.5 ドット抜け文字	12
2.2 特徴抽出	13
2.2.1 濃度こう配特徴	13
2.2.2 特徴量の変数変換	13

2.3	識別関数	15
2.3.1	Modified Quadratic Discriminant Function	15
2.3.2	MQDF 評価値	15
第 3 章	提案手法	16
3.1	提案手法の概要	16
3.1.1	入出力と目的	16
3.1.2	提案手法の流れ	17
3.2	ドット文字候補点抽出	18
3.2.1	コーナー検出	19
3.2.2	コーナーの選別	20
3.3	ドット文字列候補領域抽出	21
3.3.1	ドット文字候補点のクラスタリング	22
3.4	文字認識の前処理	23
3.4.1	切り出した文字列画像の二値化	24
3.4.2	行間の検出	25
3.4.3	膨張処理	25
3.5	ドット文字の抽出と認識評価	26
3.5.1	分離候補位置設定	26
3.5.2	文字の認識と分離位置決定	27
3.6	非ドット文字領域棄却	29
3.6.1	文字領域の画素値とアスペクト比を用いた棄却	30
3.6.2	MQDF 評価値を用いた棄却	30
3.6.3	文字領域の位置関係を用いた棄却	31
第 4 章	性能評価実験	33
4.1	実験内容	33
4.1.1	実験用画像	33
4.1.2	評価基準	34
4.1.3	画像の回転に対する頑健性検証	35
4.1.4	モーションブラーに対する頑健性検証	36
4.1.5	実行時間測定	37
4.1.6	評価対象を変化させた実験	37
4.1.7	MQDF 評価値の効果検証	37
4.2	結果	38

4.2.1	性能評価結果	38
4.2.2	評価対象を変化させた実験結果	39
4.2.3	MQDF 評価値の効果	40
第 5 章	考察	41
5.1	文字抽出に関する考察	41
5.1.1	FAST アルゴリズムの有効性	41
5.1.2	文字列候補領域抽出アルゴリズムの設定値	43
5.2	文字認識に関する考察	45
5.2.1	変動モデル学習の有効性	45
5.2.2	非文字領域棄却における Precision の向上	46
5.2.3	MQDF 評価値の効果	48
5.3	実行結果に関する考察	51
5.3.1	回転に対する頑健性	51
5.3.2	モーションブラーに対する頑健性	51
5.4	実行時間に関する考察	52
5.4.1	実行時間と性能の関係	52
5.5	産業用 OCR としての実応用に向けて	53
5.5.1	性能向上	53
5.5.2	実行時間改善	53
5.5.3	文字列の位置を設定した場合の性能向上	54
第 6 章	結論	55
6.1	本研究のまとめ	55
6.1.1	提案手法	55
6.1.2	提案手法の有効性	56
6.1.3	ドット文字列抽出	56
6.1.4	ドット文字認識	57
6.1.5	実応用に向けた実験と結果	57
6.1.6	性能向上に関する提案	58
謝辞		59

第 1 章

序論

1.1 研究背景

1.1.1 産業の支援

商工業において、大量の製品を管理することは重要な工程であり、多くの製品に管理用の固有番号、製造番号、消費期限等が印字されている。特に薬品や食品においては、消費期限や製造所といった重要な情報を製品そのものに直接示すことが法令（例として [1]）により定められている。これらの情報は、利用者（顧客、製造所の作業員）が目視により記載された内容を理解できる必要があるため、人間が見て理解できる「文字」で表示される必要がある。現在、流通している様々な製品は表面に印字された文字によって生産から消費まで管理されている。管理工程において文字の目視確認と手動でのデータ入力が行われているが、人件費や所要時間等の観点から非効率的であるため、自動化が期待されている。さらに、定期的に目視による印字状態のメンテナンスが行われているが、自動化することでメンテナンスを実施できる回数が増え、印字不良を検出した場合自動で修復して、安全性と保守性を高めることができる。

これらを自動化するためには、印字されている文字自体を検出して認識する OCR（Optical Character Recognition）が理想である。OCR によって製品を撮影した画像から文字を自動認識すれば、生産工程に複数の新規要素（コード化、コードの印字、コードの読み取り）のための新たなコストを生じさせることなく、製造段階や流通段階において製品の管理や印字状態の自動メンテナンスを支援することができる。

1.1.2 ドットマトリクス文字

管理用情報の多くは印字コストの低さやメンテナンスの容易さから，ドットインパクト方式のプリンタ [2] によって印字される．このため，図 1.1 に示すように，製品管理用の OCR は複数のドットマトリクスによって構成される文字（ドット文字）を認識する必要がある．ドット文字は，整列した点の集合が意味を成す語として知覚される特性 [3] を利用した文字であるため，人間は容易に読解できるが，OCR にとって精度を低下させる様々な問題が存在する．これらの理由から，ドット文字の OCR に効果的な手法の提案が必要である．

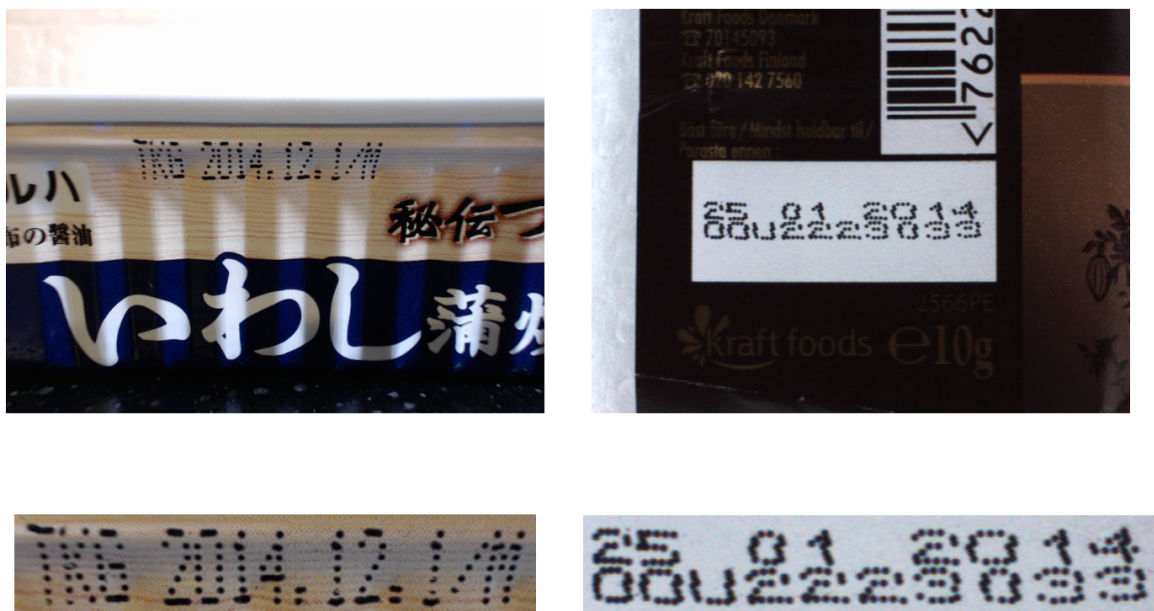


図 1.1: ドットマトリクス文字

1.2 OCR 共通の問題点

一般的に、文字以外の様々な背景を含む画像から文字を認識するためには、画像中から文字が書かれている場所を特定する抽出プロセスと、抽出した文字が何であることを判定する認識プロセスが必要である。OCR は古くから研究されてきた課題であり、様々な取り組みによって高精度化されてきたが、解決されていない課題が多く存在する [4][5][6]。

1.2.1 文字の抽出

高精度な文字の抽出はユーザの介入が必要であり、文字の位置や認識対象の形状を細かく手動設定しない限り利用することが不可能であるか、文字列部分だけが撮影されていると仮定して認識する OCR システムが多い。実際に、ユーザの介入なしで複雑な情景画像中から認識対象となる文字を検出して認識する Robust Reading Competition[7] において、文字検出性能は 80% 以下であり、精度向上が求められている。

1.2.2 文字の認識

コンピュータが認識できる文字の種類は、人間の知覚と比較して限定されている。さらに、システムが認識対象として想定していないフォントで書かれた文字や、回転や照明変化による影響を受けて変形した文字を認識することは困難である。あらゆる文字の変化に対して頑健な認識手法が求められている。

1.3 ドット文字用 OCR の問題点

OCR 共通の課題に加え，製品管理用の OCR としてドット文字を抽出して認識するために，解決すべき様々な課題が存在する．

1.3.1 ドット文字の抽出

現在，画像から文字を抽出するために，文字色の連続性を用いる手法 [8][9] (Maximally stable extremal regions, MSER) や，文字幅が一定であることを用いた手法 [10][11] (Stroke width transform, SWT) が注目されている [12]．しかし，ドット文字は複数のドットマトリクスによって文字が構成されるため，文字を連結成分として抽出したり，文字幅を推定することが困難であり，同様の手法を用いて十分な抽出性能を得ることが不可能である．また，印字される面の素材や状態が様々であり，反射や射影の条件が異なることや，製品の外観に配慮されて目立たないように印字されることから，文字の抽出と認識が困難である．

1.3.2 ドット文字の認識

図 1.2 のように，製品ごとにドット配列が異なることや，曲面印字によって形状が変化することが原因となり認識精度が低下する．また，インクののにじみや詰まりの影響で，図 1.3 のように隣り合うドットが連結する場合や，文字中のドットが部分的に抜けてしまう場合がある [13]．したがって，複数の文字が接触した場合と完全にドット間が分離した場合の両方を認識する必要がある．



図 1.2: ドット配列の違い



図 1.3: ドットが連結している例

1.3.3 製品管理用 OCR

製品管理用 OCR は，文書認識用の OCR 以上に精度と実行速度が求められ，システム操作が簡単である必要がある．図 1.4 のように，英数字を認識する産業用システムが製品化された例^{*1}も存在するが，十分な精度を得るには，事前に認識対象フォントの登録が必要であり，200 万画素以上のカメラと特定の LED 照明下で，文字列の位置を指定して読み取る必要がある．製品管理用 OCR システムの普及には，これらの問題を解決し，ユーザビリティを向上させることが必要である．

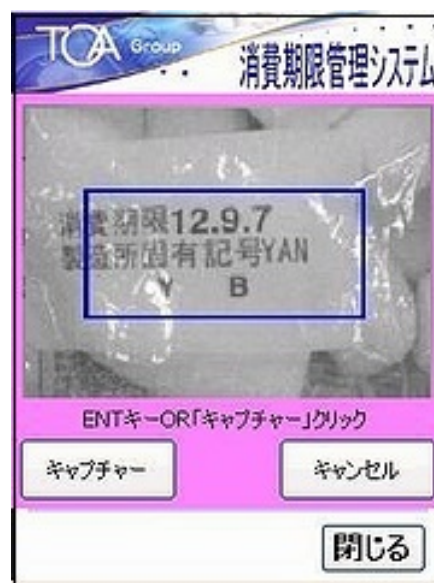


図 1.4: 実用化されている製品の例

^{*1} THC1020, <http://www.h-toa.co.jp/product/ocr/#mobile>, 東亜エレクトロニクス株式会社

1.4 本研究の取り組み

1.4.1 先行研究

ドット文字 OCR を対象とする研究は広く行われている。文字の認識に関して、1 文字が細かく分離している文字への対処法として、認識対象のドット文字に対して隣り合うドットを連結する前処理をした上で、文字を認識する手法 [14] がある。しかし、あらゆるドット文字に対して同一の前処理を用いた正確なドット間の連結は困難である。他に、変形したドット文字を認識するための前処理 [15] や特徴抽出手法 [16] も提案されている。文字の抽出に関しても、画像中のコーナーに注目して文字を抽出する手法 [17] や、輪郭抽出を用いて文字を抽出する手法 [18][19] が存在する。また、産業用に文字認識をするシステムが製品化されている。しかし、先行研究や製品の多くが、文字のフォント、光源の強度や色、印字位置や素材などの条件をあらかじめユーザに指定させ、その環境下での使用に限定することで、ドット文字認識に関する問題を回避している。さらに、上記の先行研究は実際の OCR システムの一部であり、前提とする条件や想定している対象がそれぞれ異なる。そのため、画像を入力して文字を認識する End-to-End の手法が必要である。

1.4.2 研究目的

本研究では、ドット文字が表面に印字された製品の画像を入力し、ユーザによる事前の情報登録なしでドット文字のみを抽出して認識する手法を提案する。本研究は、文字が印字されている素材、光源の色や強度、印字位置、ドット文字配列の種類を制限せず、あらゆる環境下であらゆる製品上のドット文字認識に適用可能な OCR システムの提案を目的としている。

第 2 章

文字認識手法

本章では，本研究で用いる以下の学習辞書作成手法，特徴抽出手法，文字認識手法について記す．

1. 変動モデル学習
2. 濃度こう配特徴抽出
3. Modified Quadratic Discriminant Function (MQDF)

2.1 学習辞書作成

2.1.1 変動モデル学習

変動モデル学習 [20] は，様々な種類のドット文字を字種ごとに同一クラスとして学習させることで，局面への印字やドット構成の違いによって形状が異なる文字を認識するための学習辞書作成手法である．本研究では，以下に示す 4 種類の変動モデルを生成して学習する．

- 複数のドット文字フォント
- 3 次元回転文字
- ドット直径の変更
- ドット抜け文字

2.1.2 複数のドット文字フォント

品質情報の表示に使用されるドット文字の多くは、(縦 × 横) = (5 × 7) または (5 × 5) ドットで構成される英数字であるため、これらのドット文字を人工的に生成する。文字生成処理の流れを図 2.1 に示す。まず生成する文字画像の縦サイズ h 画素と横サイズ w 画素を決定する。次に、画像を 5 × 7 と 5 × 5 のブロックに分割する。最後にドットマトリクスが印字される位置にあるブロックの中心に直径 5[pixel] の塗りつぶされた円を配置することで、ドット文字を生成する。生成する (5 × 7) 配列のドット文字を図 2.2 に、(5 × 5) 配列のドット文字を図 2.3 に示す。また、ドット文字は同じ文字種でも異なるドット配置で印字される場合がある。このようなドット文字を安定して認識させるため、本研究においては「1,2,5,6」を 2 種類、「3」を 3 種類のドット構成で生成する。ドット配置の異なるドット文字を図 2.4 に示す。生成する学習サンプル数を表 2.1 に示す。

表 2.1: 複数のドット文字フォントの変動モデル学習

ドットマトリクスの構成種類	82 種類
横サイズ	11 通り (30~50[pixel], 2 画素間隔で生成)
縦サイズ	9 通り (30~50[pixel], 2 画素間隔で生成)
学習サンプル数合計	8118 画像

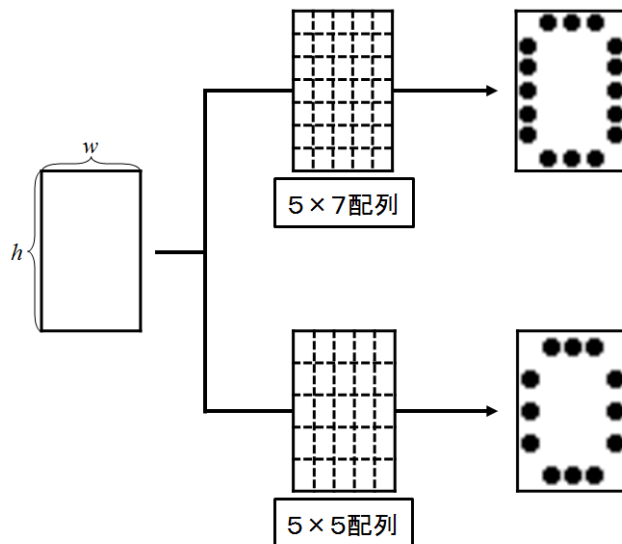


図 2.1: ドット文字の変動モデル生成

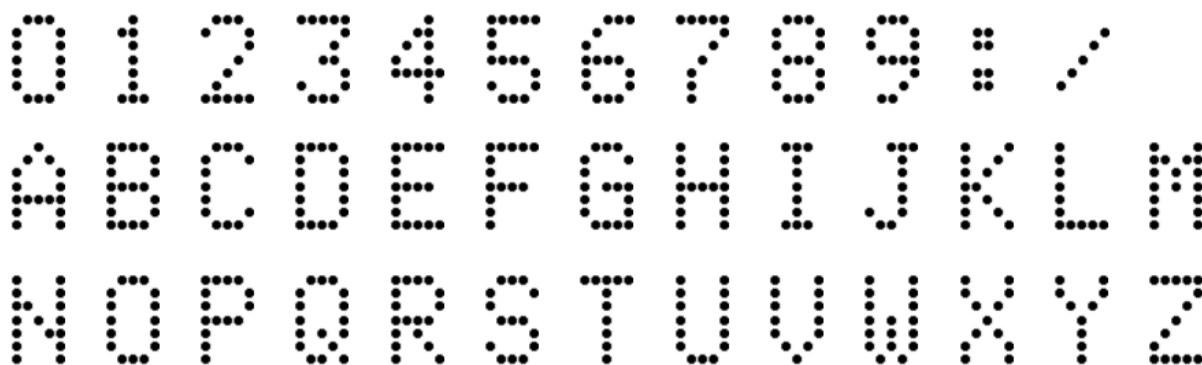


図 2.2: 5 × 7 ドット文字の生成

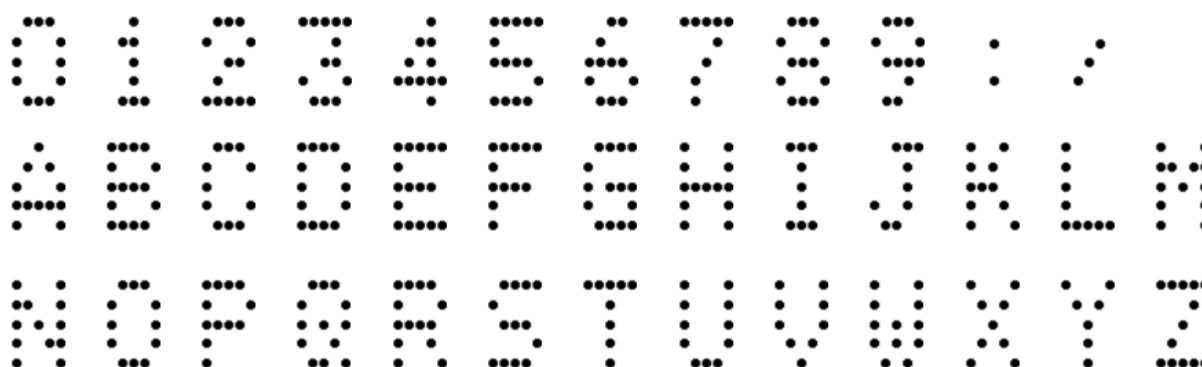


図 2.3: 5 × 5 ドット文字の生成

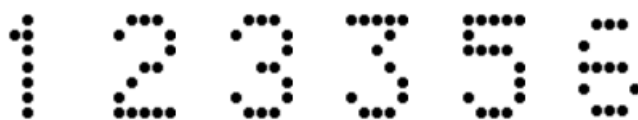


図 2.4: 1,2,3,5,6 の追加生成

2.1.3 3次元回転文字

曲面に印字されたドット文字や，斜め方向から撮影されたり，回転したりしているドット文字を認識するために，3次元回転処理 [21] によって回転したドット文字を生成する．3次元回転処理とは， xy 平面上に存在する文字に対して水平方向を軸に回転する x 軸回転，垂直方向を軸に回転する y 軸回転，文字の中心を軸に二次元平面上の回転を行う z 軸回転を組み合わせて文字を3次元空間上で回転させ， xy 平面に投影させる処理である．本研究では， 5×7 のドット文字に3次元回転の変動モデルを適用し，文字を生成する．ただし，文字列を切り出す処理に含まれるアフィン変換によって z 軸方向の回転は補正されるため， z 軸回転の変動モデルは生成しない．座標系は左手系であり，回転は x 軸回転， y 軸回転の順で，画素値の線形補間によって変換画像を生成する．回転によって生成された文字を図 2.5 に示す．生成する学習サンプル数を表 2.1 に示す．

表 2.2: 3次元回転文字の変動モデル学習

ドットマトリクスの構成種類	38 種類
回転角度	49 通り ($\pm 30^\circ$ 回転, 10° 間隔で生成)
横サイズ	2 通り (30[pixel], 50[pixel])
縦サイズ	2 通り (50[pixel], 90[pixel])
学習サンプル数合計	7448 画像



(a) x 軸方向に 30 度回転した文字



(b) y 軸方向に 30 度回転した文字

図 2.5: 回転した文字の変動モデル

2.1.4 ドット直径の変更

インクのにじみや印字機器の違いによって、隣り合うドット同士が連結したドット文字を認識するために、学習サンプルを生成する際の各文字のドット直径 $d[\text{pixel}]$ を変化させてドット文字を生成する。ドット直径の変更によって生成された文字を図 2.6 に示す。生成する学習サンプル数を表 2.3 に示す。

表 2.3: ドット直径の変更の変動モデル学習

ドットマトリクスの構成種類	38 種類
ドット直径	4 通り ($d = 5, 7, 9, 11$)
横サイズ	6 通り (30~50[pixel], 4 画素間隔で生成)
縦サイズ	9 通り (50~90[pixel], 5 画素間隔で生成)
学習サンプル数合計	8208 画像

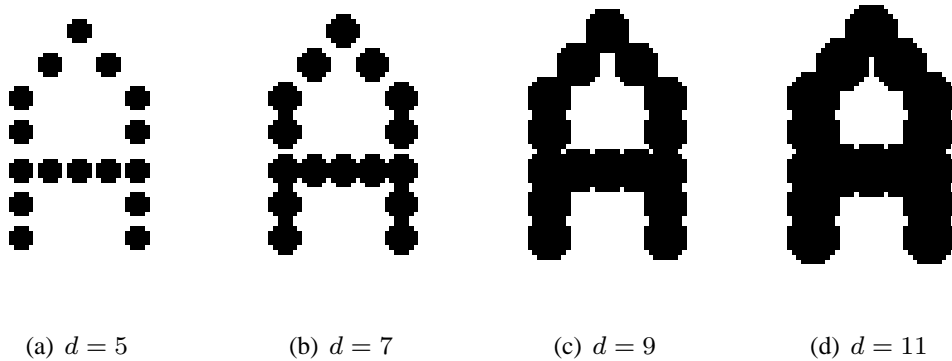


図 2.6: 回転した文字の変動モデル

2.1.5 ドット抜け文字

ドットマトリクスの一部が消失したドット文字を認識するために，ドットマトリクスの点群から1個を消失させたドット文字を生成する．本研究では， 5×7 のドット文字にドット抜けの変動モデルを適用し，文字を生成する．生成文字例を図 2.7 に示す．ただし，ドットマトリクス1個を消失させたことにより，異なる文字種が同じ形状になる場合には，パターン（例として図 2.8）を学習サンプルから除外する．生成する文字数を表 2.4 に示す．

表 2.4: ドット抜け文字の変動モデル学習

ドットマトリクスの構成種類	542 種類
横サイズ	2 通り (30[pixel], 50[pixel])
縦サイズ	2 通り (50[pixel], 90[pixel])
学習サンプル数合計	6766 画像

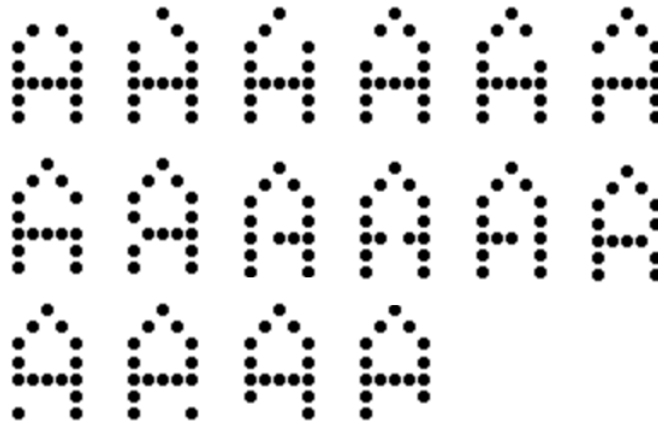


図 2.7: ドット抜け文字



図 2.8: 学習から除外するドット抜け文字の例

2.2 特徴抽出

2.2.1 濃度こう配特徴

特徴ベクトルには、図 2.9 に示す濃度こう配特徴 [22] を用いる。1 画像から $8 \times 6 \times 6$ (方向数 \times 横ブロック数 \times 縦ブロック数) の 288 次元の濃度こう配特徴を抽出する流れを以下に示す。

1. 入力画像のサイズを $56 \times 56[\text{pixel}]$ に正規化する。
2. 2×2 平均値フィルタを 9 回適用して正規化画像を平滑化する。
3. Roberts フィルタにより、濃度値曲面のこう配方向と強度を取得する。
4. こう配の方向を $\pi/16$ 刻みで向きを考慮した 32 方向に量子化する。
5. 量子化した方向別に 32 枚の画像を作成し、各画像を 11×11 のブロックに分割する。
6. 各ブロック内で、こう配強度を加算して $32 \times 11 \times 11$ 次元の局所方向ヒストグラムを作成する。
7. 加重フィルタ $[1\ 4\ 6\ 4\ 1]$ を 1 方向置きに適用することで、ヒストグラムを 32 方向から 16 方向に再標本化する。
8. 加重フィルタ $[1\ 2\ 1]$ を 1 方向置きに適用することで、ヒストグラムを 16 方向から 8 方向に再標本化する。
9. 7×7 の二次元ガウスフィルタを縦横 1 ブロック置きに施し、分割ブロック数を 6×6 に削減する。
10. 抽出した特徴ベクトルの各成分に対して変数変換 [23] を行い、特徴量の分布を正規分布に近づける。

2.2.2 特徴量の変数変換

特徴抽出の手順 10 では、変数を式 2.1 によって変換し、特徴量の分布を正規分布に近づけている。

$$y = x^\nu \quad (2.1)$$

識別に用いる関数が、観測値の分布を多変量正規分布と仮定してベイズ決定測から導かれているためである。観測値が多変量正規分布でない場合は、この最適性が失われるため、変数を変換して観測値の分布を正規分布に近づけることが有効である。一般に、正の数のみを計数して観測された特徴量はガンマ分布になることが多い。一次元の確率変数の場合には、式 2.1 の変換をすることで、図 2.10 のように平均値周辺の対称性と正規性が改善される。本研究では、先行研究 [24] によって濃度こう配特徴を用いる英数字の識別に最適とされる $\nu = 0.4$ を用いる。

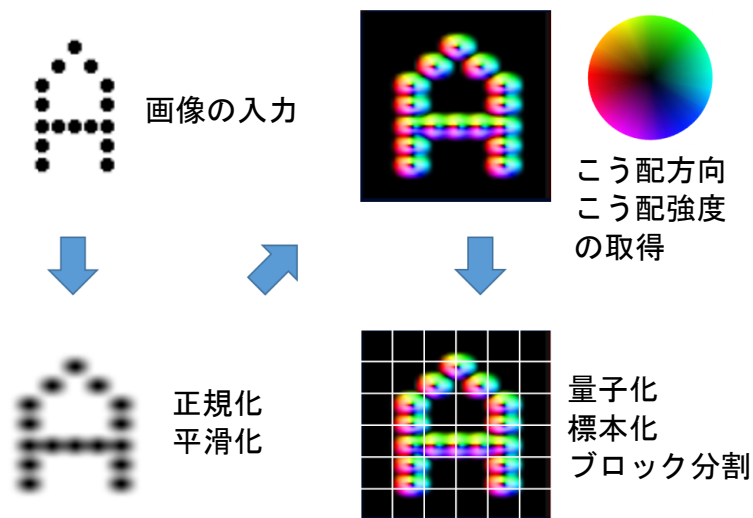


図 2.9: 濃度こう配特徴抽出

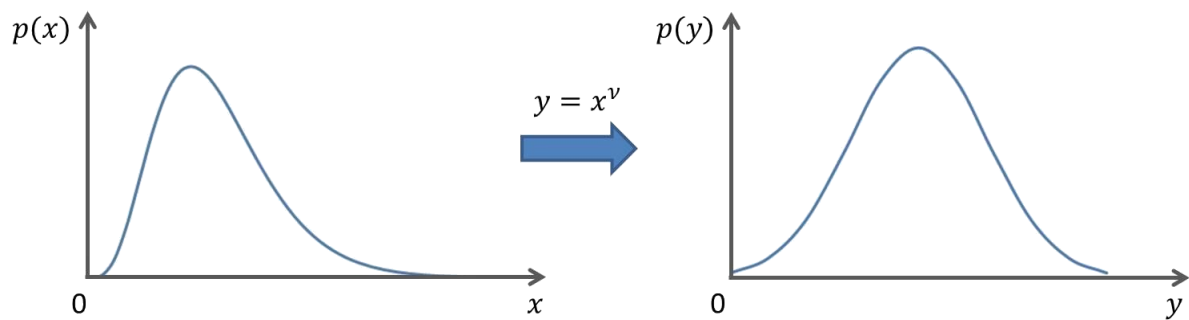


図 2.10: 変数変換

2.3 識別関数

2.3.1 Modified Quadratic Discriminant Function

ドット文字認識の識別器として、式 2.2 であらわされる Modified Quadratic Discriminant Function (MQDF) [25] を使用する. MQDF は分布パラメータのうち母集団の共分散行列が未知の正規分布に対する最適識別関数 [26][27] から導出された近似式で、識別精度を損なうことなく識別時の計算量と記憶容量を $O(n^2)$ から $O(kn)$ に削減できる特徴がある [28][29][30]. 学習サンプル数に対して、特徴ベクトルの次元数を過度に増やすと性能が低下するピーキング現象 [31] を抑える効果があり、共分散行列の推定誤差に起因する性能低下 [32] も少ない. 本研究では、各パラメータは先行研究 [20] においてドット文字の認識率が最も高いとされる $\alpha = 0.1, k = 40$ を用いる.

$$g(X) = \frac{1}{\alpha\sigma^2} \left[\|X - M\|^2 - \sum_{i=1}^k \frac{(1-\alpha)\lambda_i}{(1-\alpha)\lambda_i + \alpha\sigma^2} \{\Phi_i^T(X - M)\}^2 \right] + \sum_{i=1}^k \ln \{(1-\alpha)\lambda_i + \alpha\sigma^2\} \quad (2.2)$$

- X : 入力文字の特徴ベクトル
- M : 母集団の平均ベクトル
- λ_i : 標本共分散行列の第 i 固有値
- Φ_i : 標本共分散行列の第 i 固有ベクトル
- k : 識別に用いる固有ベクトル数
- σ^2 : 特徴ベクトル X の事前確率分布を球状分布と仮定した場合の分散
- α : σ^2 の信頼度を表す定数

2.3.2 MQDF 評価値

MQDF では、 $g(X)$ が最小となるクラスが認識結果となる. 以降、認識結果の $g(X)$ の値を MQDF 評価値と記述する.

第 3 章

提案手法

3.1 提案手法の概要

3.1.1 入出力と目的

本研究では，ドット文字が表面に印字された製品の画像を入力し，ユーザによる事前の情報登録なしでドット文字のみを自動抽出して認識する手法を提案する．本手法の入力と出力の例を図 3.1 に示す．



(a) 入力：ドット文字が印字された製品の画像 (b) 出力：ドット文字の抽出位置と認識結果

図 3.1: 提案手法の入力と出力の例

3.1.2 提案手法の流れ

提案手法は以下のステップによって構成される。

1. ドット文字列の抽出

- (a) ドット文字候補点抽出

入力画像から Features from Accelerated Segment Test (FAST) コーナー [33][34] を抽出し、こう配方向、こう配強度、彩度からドット文字上に存在する FAST コーナーを選別する。

- (b) ドット文字列候補領域抽出

ドット文字候補点を、近傍画素の輝度を用いてクラスタリングすることでドット文字列候補領域を抽出する。

2. ドット文字の認識

- (a) 前処理（二値化，行間検出，膨張処理）

抽出した文字列候補領域を二値化し，行間の位置を検出する．膨張処理を適用してドットを結合し，連結成分を抽出した後，領域の大きさと位置関係を用いて文字列候補領域から 1 文字の候補領域を抽出する．それぞれの文字候補領域から濃度こう配特徴 [22] を抽出する．

- (b) ドット文字の抽出と認識

ドット文字の変動モデル学習 [20] によって作成された学習辞書と Modified Quadratic Discriminant Function (MQDF) 識別器 [25] を用いて，文字候補領域を認識し，文字認識評価値（MQDF 評価値）を算出する．

3. 非ドット文字の棄却

抽出された文字候補領域の MQDF 評価値と位置関係を用いて非ドット文字領域を棄却する．

以降，図 3.1 の例を用いて，それぞれのプロセスの詳細について記述する．

3.2 ドット文字候補点抽出

様々な背景を含む画像から文字を認識するためには，最初に画像から文字列を抽出する必要がある．画像からドット文字列を抽出するために，画像中のコーナーを検出し，ドット文字の特徴を持つコーナーを文字候補点として抽出する．コーナーとは，図 3.2 に示すように，画像中のエッジが垂直に交差する点である．本ステップで誤った候補点を抽出しても後処理でノイズとして除去することはできるが，抽出すべき点を抽出できない場合は以降の処理で文字を認識することができないため，再現性を重視して抽出する必要がある．そのため，False-Positive を含む多めのドット文字候補点を抽出する．ドット文字上で抽出されたコーナーを図 3.3 に，検出されたすべてのコーナーを図 3.4(a) に，ドット文字を構成する候補点として抽出された点を図 3.4(b) に示す．

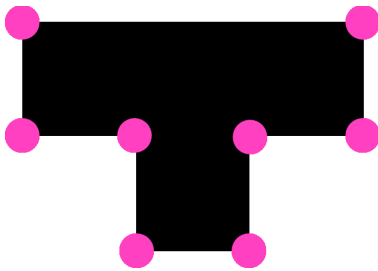


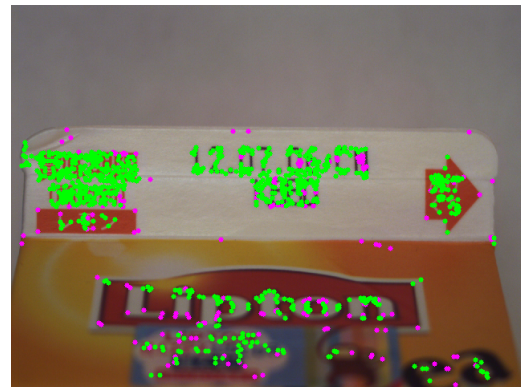
図 3.2: 画像中のコーナー



図 3.3: ドット文字上のコーナー



(a) 紫：検出されたすべてのコーナー



(b) 緑：ドット文字候補点となるコーナー

図 3.4: ドット文字候補点抽出

3.2.1 コーナー検出

ドット文字はコーナー検出によって抽出可能であると想定し，入力画像からコーナー点を検出する．高い再現性で高速にコーナーを検出するために，Features from Accelerated Segment Test (FAST) アルゴリズムを用いる．

以下に FAST アルゴリズムを示す．

1. グレイスケール画像上の点 P を決め，その画素の値を I_P とする．
2. P を囲む 16 画素（図 3.5）について調べ，画素の値を I_q とする． ($1 \leq q \leq 16$)
3. $I_q \leq I_P - T$ の画素が 12 個以上連続している，または $I_P + T \leq I_q$ の画素が 12 個以上連続している場合， P をコーナーとする．
4. すべての画素に対して上記の 1～3 を実行し，画像上のすべてのコーナーを検出する．

本手法では，初期設定値を $T = 5$ とし，検出されたコーナーの数が 500 よりも多い場合は， T を 1 ずつ増加させて再度 FAST コーナーを検出する．

		16	1	2		
	15				3	
14						4
13			P			5
12						6
	11				7	
		10	9	8		

図 3.5: p を囲む画素

3.2.2 コーナーの選別

検出したコーナーから，ドット文字上に存在するコーナーを選別し，ドット文字候補点を抽出する．文字を構成するドットマトリクスには以下の特徴があるため，これらの特徴を持つコーナーを抽出する．

- 製品上の文字を読みやすくするため，背景と明度差があり，こう配強度が大きい．
- 1 個のドットマトリクスは円形であるため，周囲のこう配方向が分散している．
- 製品外観に影響を与えないため，あるいはコスト削減のため，印字の彩度が低い．

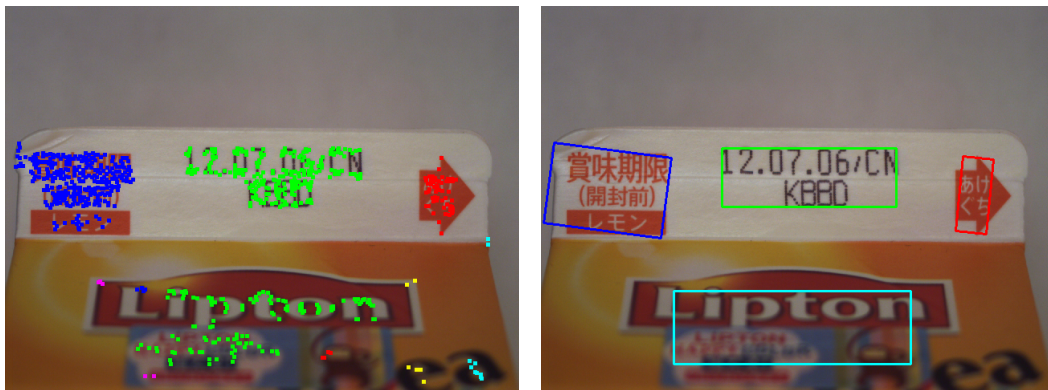
FAST アルゴリズムによって検出された任意のコーナーを $F_i = (x_i, y_i, s_i, m_i, d_i)$ とする．ここで， x_i, y_i, s_i, m_i, d_i はそれぞれ，コーナー画素の座標，正規化された彩度値，画素の周囲 16×16 の局所領域における平均こう配強度値，こう配を 8 方向に量子化して求めた方向ヒストグラム $d_i = \{d_{i,0}, d_{i,1}, \dots, d_{i,7}\}$ である．こう配強度，方向は二次ソーベルオペレータで求めたものである．

抽出された全てのコーナーに対して，以下の 4 条件が全て成り立つか判定する．全ての条件を満たすコーナーをドット文字を構成する候補点 c_j として抽出する ($j \subset i$)． j は候補点の通し番号である．

- $m_i > \bar{m}$, ここで， \bar{m} は画像全体の平均こう配強度である．
- $d_i^{\max} < \bar{d} + \sigma_d$, ここで， $d_i^{\max}, \bar{d}, \sigma_d$ はそれぞれ，値が最大となる d_i の要素，全コーナー点の d_i の要素の平均値及び標準偏差である．
- F_i の周囲 16×16 の局所領域に Canny オペレータ [35] により検出されるエッジ画素が存在する．
- $s_i < 0.8$

3.3 ドット文字列候補領域抽出

抽出したドット文字候補点をクラスタリングし，それぞれにラベルを付与して，同じラベルの候補点を内包する矩形を抽出することで，ドット文字列候補領域 $T_i = (x_i, y_i, w_i, h_i, \theta_i)$ とする．ここで， $x_i, y_i, w_i, h_i, \theta_i$ はそれぞれ，矩形の左上座標，横のサイズ，縦のサイズ，回転角度である．本ステップで誤った候補領域を抽出しても後処理で除去することはできるが，抽出すべき領域を抽出できない場合は以降の処理で文字を認識することができないため，再現性を重視して多めに抽出する必要がある．そのため，False-Positive を含むドット文字列候補領域を抽出する．図 3.4 をクラスタリングした結果を図 3.6(a) に示し，文字列候補領域を抽出した結果を図 3.6(b) に示す．背景を含む画像から切り出されたドット文字列候補領域を図 3.7 に示す．



(a) 文字候補点のクラスタリング

(b) 抽出されたドット文字列候補領域

図 3.6: ドット文字列候補領域抽出



図 3.7: 切り出されたドット文字列候補領域

3.3.1 ドット文字候補点のクラスタリング

ドット文字候補点を以下の手順でクラスタリングし、ドット文字列候補領域を抽出する。同じ文字列周辺は同程度の輝度で書かれていると想定し、候補点周辺の輝度の正規化ヒストグラムインタセクションを用いて候補点にラベルを付与する。

1. 256 階調 RGB である入力画像を 16 階調 YCbCr 空間に変換し、Y 信号のみを取り出すことで、16 階調グレイスケール画像に変換する。
2. 各候補点 c_j の周囲 16×16 画素の輝度ヒストグラム $h(c_j)$ を作成する。
3. 画像をラスタ走査して、 c_j の中から未分類の候補点を見つけ、 $c_a = (x_a, y_a)$ とする。 c_a に新しいラベルを付与して、手順 4 へ進む。ラスタ走査が画像の右下まで到達したら手順 6 へ進む。
4. c_a を起点にラスタ走査を継続し、 c_j の中からラベルが付与されていない候補点を見つけ、 c_b とする。 c_a と c_b について、式 (3.1) の条件が成立する場合、 c_b に c_a と同じラベルを付与し、手順 5 へ進む。成立しない場合、 c_b にはラベルを付与せずラスタ走査を継続する。ラスタ走査が画像の右下に到達するまで繰り返す。

$$mD_h > D_e \quad (3.1)$$

ここで、 D_h は $h(c_a)$ と $h(c_b)$ の正規化ヒストグラムインタセクション、 D_e は c_a, c_b の画像座標間のユークリッド距離であり、それぞれ以下の式により計算される。 m は適当に設定する定数である。

$$\begin{aligned} D_h &= h(c_a) \otimes h(c_b) \\ &= \frac{1}{16 \times 16} \sum_{j=0}^{15} \min(h_j(c_a), h_j(c_b)) \end{aligned} \quad (3.2)$$

$$D_e = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (3.3)$$

5. 候補点 c_b を新しい基準点として c_a とみなし、手順 4 を再帰的に処理する。
6. 同じラベルの候補点を内包する最小矩形をドット文字列候補領域として切り出す。

再現性を高くするため、式 (3.3) の m を 16, 24, 32 と変化させて 3 回実行し、全てのクラスタリング結果を切り出す。

3.4 文字認識の前処理

切り出したドット文字列候補領域の画像について、矩形の長辺の傾きを回転角度として、回転を0度に補正する。また、画像の縦横それぞれの大きさを2倍する。以下の理由から文字認識に必要であり、また有効であるため、画像を二値化する。

- 文字認識のために、文字画像の背景を白、文字を黒と二値化する手法は有効である[48][49]。
- 濃度こう配特徴を用いる場合でも、文字の切り出しや背景ノイズの除去のために、二値化プロセスが有効である。
- 多値画像処理よりも簡単に高速に処理することができる。

文字認識では、文字列全体を1文字単位の複数の領域に分離し、分離した領域それぞれを認識する。一般的に、二値画像中の大文字英字と数字は、連結成分の外接矩形を抽出することで文字列から1文字単位に分離することが可能である。ドット文字は1文字が複数の連結成分で構成されているため、前処理によってドット文字を可能な限り1連結成分が1文字として抽出できる状態にする。ドット文字列は2行で記述されることもあるため、行間を検出し、1文字中のドットマトリクスを接触させるために黒画素を膨張する。それぞれの処理過程を図3.8に示す。

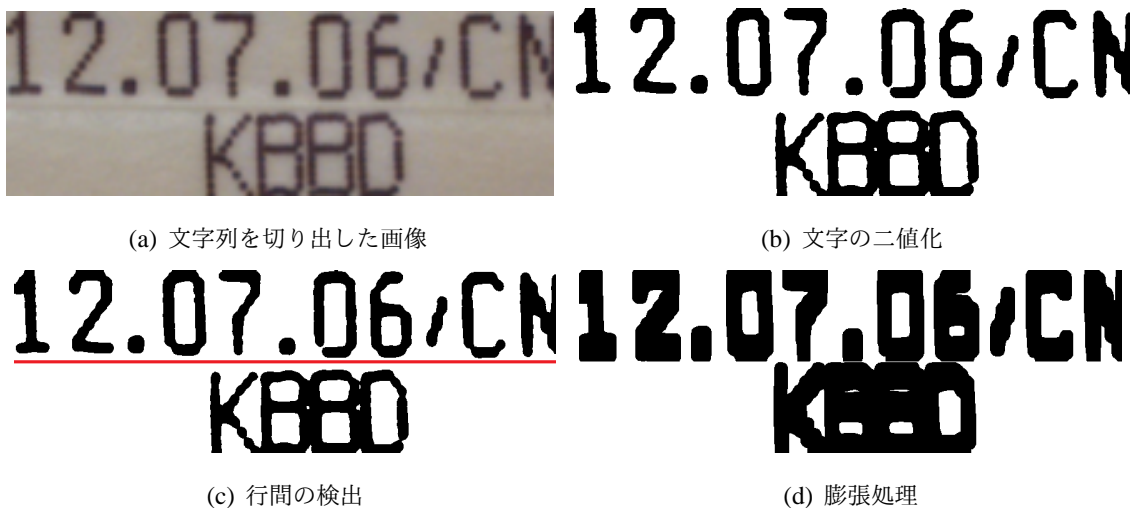


図 3.8: ドット文字の抽出と認識のための前処理

3.4.1 切り出した文字列画像の二値化

文字列領域は複数の文字が領域内にあるため，縦サイズと比較して横サイズが大きくなりやすい．このため照明変化の影響を受けやすく，一定のしきい値で全体を二値化することは困難である．本手法では，文字列画像を局所しきい値法 [36] によって二値化する．局所しきい値法は文字列画像を k 個のブロックに分割し，それぞれのブロックを大津の二値化手法 [37] によって二値化する．ブロック分割数 k は式 3.4 によって定められる．ここで， w と h はそれぞれ，文字列画像の横と縦のサイズである．図 3.9(a) の二値化をするときの画像の分割例を図 3.9(b) に，提案手法によって二値化した画像を図 3.9(c) に，判別分析法を用いて単一のしきい値で二値化した画像を図 3.9(d) に示す．提案手法は局所領域ごとにしきい値を決定することで，水平方向の照明変化による影響を軽減できるため，単一のしきい値を用いる手法よりも正確に文字列を二値化することができる．

$$k = \max \left(\text{round} \left(\frac{w}{h} \right), 1 \right) \quad (3.4)$$

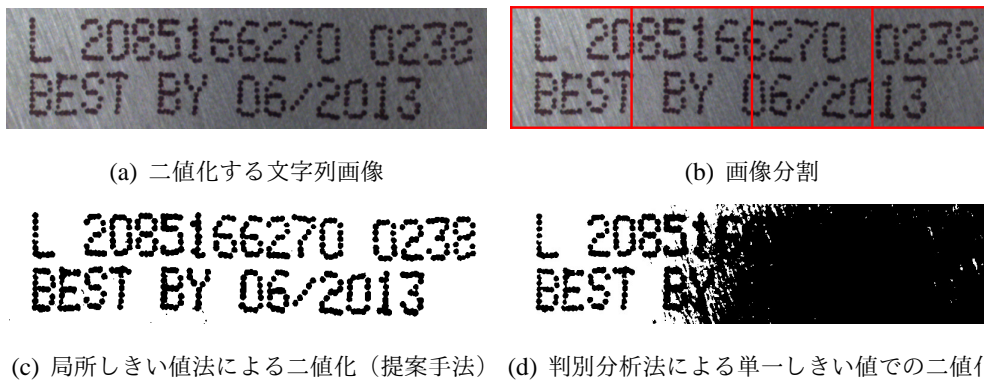


図 3.9: 文字列の二値化

3.4.2 行間の検出

切り出した文字列画像が2行で書かれているかどうかを前もって知ることはできない。最初に、文字列が2行に書かれていると仮定し、行を分けている座標 y_l を検出する。文字列画像の縦のサイズを h として、水平投影ヒストグラム $P = \{p(1), p(2), p(3), \dots, p(h)\}$ を作成し、 P に判別分析法を適用して求められた位置を y_l とする。次に、 $p(i)$ の平均値 $m_P = \Sigma p(i)/h$ と標準偏差 $\sigma_P = \sqrt{\Sigma (p(i) - m_P)^2/h}$ が式 3.5 を満たす場合、検出された位置 y_l を行間とする。満たさない場合、1行で書かれており行間がない文字列とする。

$$p(y_l) < m_P - \sigma_P \quad (3.5)$$

3.4.3 膨張処理

膨張は、二値図形を X ，図 3.10 に示す構造化要素を B とすると、 $X \oplus B$ で表される。膨張を指定回数繰り返すことにより1文字を構成するドットマトリクスを結合することができる。膨張によって上下の文字が接触することを避けるため、複数行に書かれている文字列において、指定回数に満たない状態でも検出した行間 y_l で膨張を停止する。

	1	
1	1	1
	1	

図 3.10: 構造化要素 B

3.5 ドット文字の抽出と認識評価

切り出された文字列画像に書かれた文字を認識する。画素のヒストグラムや輪郭の形状のみで文字列を正しく 1 文字単位に分離することは困難であるため [38], 提案手法では, 前処理後の二値画像を用いて複数の組み合わせパターンで分離候補領域を設定し, 後に MQDF 評価値によって最適な文字領域を決定する手法を用いる。

3.5.1 分離候補位置設定

前処理後の画像について, 連結成分の外接矩形を切り出し, 文字領域 $R_i = (x_i, y_i, w_i, h_i)$ とする。ここで, x_i と y_i はそれぞれ矩形の左上 x 座標と y 座標であり, w_i と h_i は幅と高さである。前処理後の文字列には, 膨張処理によるドットマトリクスの結合に失敗し, 1 文字が複数の連結成分で表現されている文字や, 過剰に膨張したため隣接した他の文字と接触した文字が含まれるため, 文字領域 R_i の抽出位置を以下の手順で修正する。

図 3.11 に示すように, 文字領域の抽出位置修正は領域の結合と分割によって行われる。

1. 2 個の文字領域 R_i, R_j が 20% 以上重なっている場合, 1 文字が 2 個の領域で表されているものとして, 領域を結合する。 R_i と R_j を結合する例を図 3.11(a) に示す。
2. 文字領域 R_i について, $w_i > h_i$ の場合, 複数の文字が 1 個の領域で表されているものとして, 領域を分割する。 w_i/h_i の整数部分を s として, R_i を $1, s+1, \dots, s+5$ 個のバリエーションで等分割する。 R_i を 4 分割する例を図 3.11(b) に示す。 5 種類の分割パターンの内, 1 パターンを除き抽出位置が誤っている。
3. ここでは, 膨張後の文字領域を抽出しているが, 高精度な認識のためには入力画像上に対して文字領域を抽出するべきである。文字領域の分割位置を膨張前の二値画像に適用し, 各小領域中に存在する黒画素の最小包含矩形を求める。
4. 縦または横が 40 画素未満の領域が分割パターンに含まれる場合, 文字認識のために十分な特徴が抽出できないため, パターンから除外する。

これにより, 分離候補領域を含む, 入力画像上の文字領域を抽出することができる。

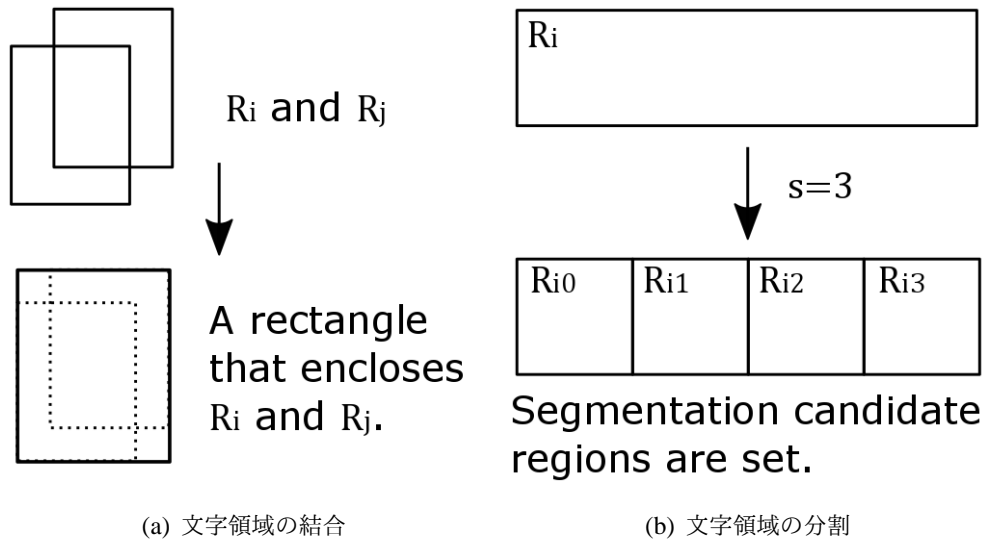














図 3.11: 文字領域の抽出位置修正

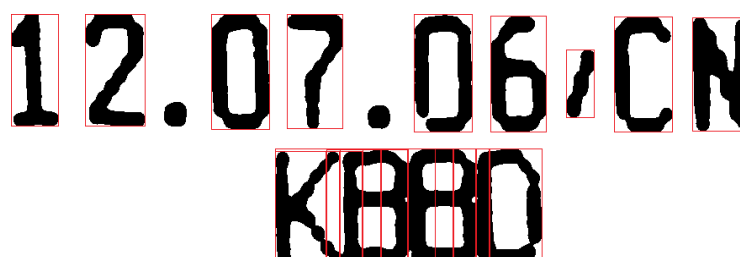
3.5.2 文字の認識と分離位置決定

膨張前の二値画像を読み込み，すべての文字候補領域 R_i を認識する．本研究では，抽出と認識の対象字種は大文字英字 (A から Z)，数字 (0 から 9)，スラッシュ (/) とする．変動モデル学習で学習サンプルを増やし，特徴ベクトルは濃度こう配特徴，識別関数は MQDF を用いる．

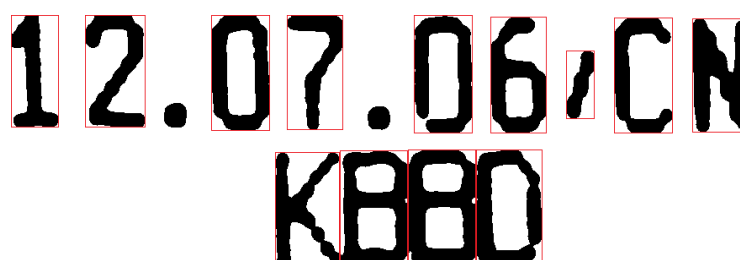
認識時に算出される MQDF 評価値は，その領域の「文字らしさ」を表す．認識した文字領域に複数の分離候補位置がある場合，表 3.1 に示すように，平均の MQDF 評価値が最も低い分割数を正しい分割数とし，認識結果を抽出された文字とする．文字列の分離結果を図 3.12 に示す．

表 3.1: 文字候補領域ごとの MQDF 評価値の平均

S	文字 1	文字 2	文字 3	文字 4	文字 5	平均評価値
2				N/A	N/A	472.453
3					N/A	32.471
4						453.900



(a) 文字候補領域



(b) MQDF 評価値によって決定された文字領域

図 3.12: MQDF 評価値を用いた分離位置決定

3.6 非ドット文字領域棄却

ドット文字候補点抽出とドット文字列候補領域抽出で再現性を重視したため，図 3.7 に示すように，抽出結果は非ドット文字領域を多く含む．本手法では，これら全ての領域を分離し，認識したため，現時点でドット文字の抽出と認識の結果は図 3.13(a) のようになっている．抽出された全領域からドット文字のみを抽出するために，MQDF 評価値，文字領域の画素値とアスペクト比，文字領域同士の位置関係を用いて，非ドット文字領域を False-Positive として棄却する．それぞれの処理過程を図 3.13 に示す．

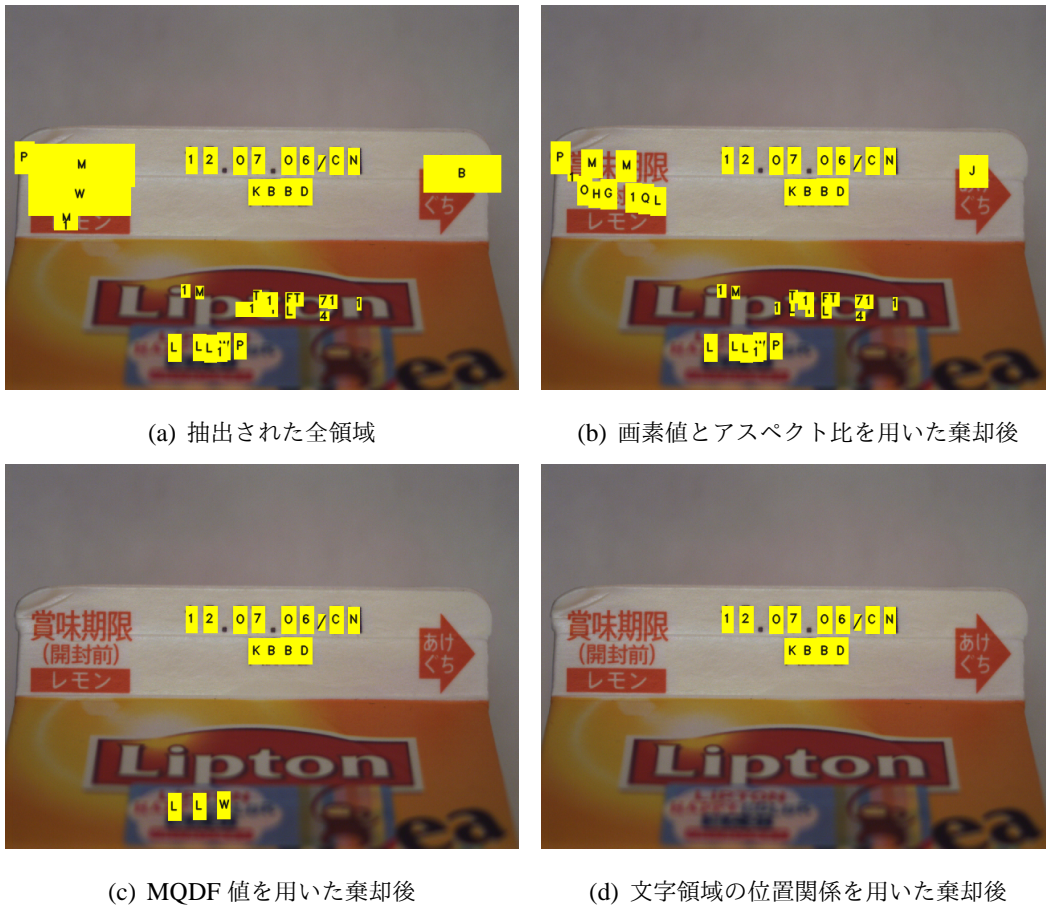


図 3.13: 非ドット文字領域の棄却

3.6.1 文字領域の画素値とアスペクト比を用いた棄却

ドット文字領域は横長の領域にはならない。また、領域の大部分が白画素のみ、黒画素のみで構成されることは無い。このため、文字領域が以下の条件 1,2,3 のいずれかを満たす場合、非ドット文字領域として棄却する。棄却後のドット文字領域を図 3.13(b) に示す。

1. ドット文字領域の 90% 以上が白画素である。
2. ドット文字領域の 90% 以上が黒画素である。
3. ドット文字領域の幅が高さの 2 倍以上大きい。

3.6.2 MQDF 評価値を用いた棄却

非ドット文字を認識した場合、MQDF 評価値は大きな値となる [41]。このため、文字領域の MQDF 評価値が以下の条件 4,5 のいずれかを満たす場合、非ドット文字領域として棄却する。棄却後のドット文字領域を図 3.13(c) に示す。

4. ドット文字領域の MQDF 値が 600 以上である。
5. ドット文字領域が所属するドット文字列候補領域の平均 MQDF 値が 400 以上である。

しきい値 600, 400 は予備実験により定めた。

3.6.3 文字領域の位置関係を用いた棄却

ドット文字は消費期限や製造番号等の目視で理解できる管理情報を表す文字であるため、単独で印字されることや離れた位置に散在することは無い。このため、文字領域やドット文字列候補領域の位置関係を用いて、文字領域が以下の条件 6,7,8 のいずれかを満たす場合、非ドット文字領域として棄却する。文字領域の位置関係について、それぞれの状況を図 3.14 に示す。棄却後のドット文字領域を図 3.13(d) に示す。

6. 複数の文字領域が 20% 以上重なっている場合で、MQDF 評価値が最小ではない。
(A)

図 3.14 の場合、領域 R_1 と R_2 の MQDF 評価値を $g(R_1), g(R_2)$ としたとき、 $g(R_1) > g(R_2)$ であれば、 R_1 が棄却される。

7. 文字領域の幅の 3 倍以内の距離の範囲に他の文字領域が存在しない。(B)

図 3.14 の場合、 R_{10} の幅の 3 倍以内の距離にほかの文字領域が存在しないため、 R_{10} が棄却される。

8. 平均 MQDF 評価値が最も小さい文字列領域 T_{min} と文字領域 R_i の距離 $D_{(T_{min}, R_i)}$ が文字領域の幅の 3 倍以上である。(C)

領域 T_i に含まれる文字の MQDF 評価値の平均を $g(T_i)$ したとき、 $g(T_i)$ が最小となる文字列領域 T_{min} について、領域の中心を通り、回転角度を傾きとする直線を定義する。図 3.14 の場合、最も平均 MQDF 評価値が小さい領域は T_2 であり、 T_2 の中心座標と回転角度により直線が定義される。 $D_{(T_2, R_{13})}$ は文字列領域 T_2 によって定義された直線と文字領域 R_{13} の中心との距離である。 T_2 によって定義される直線と R_{13} の中心との距離 $D_{(T_2, R_{13})}$ が R_{13} の幅の 3 倍以上であるため、 R_{13} が棄却される。 R_{11} と R_{12} も同条件で棄却される。

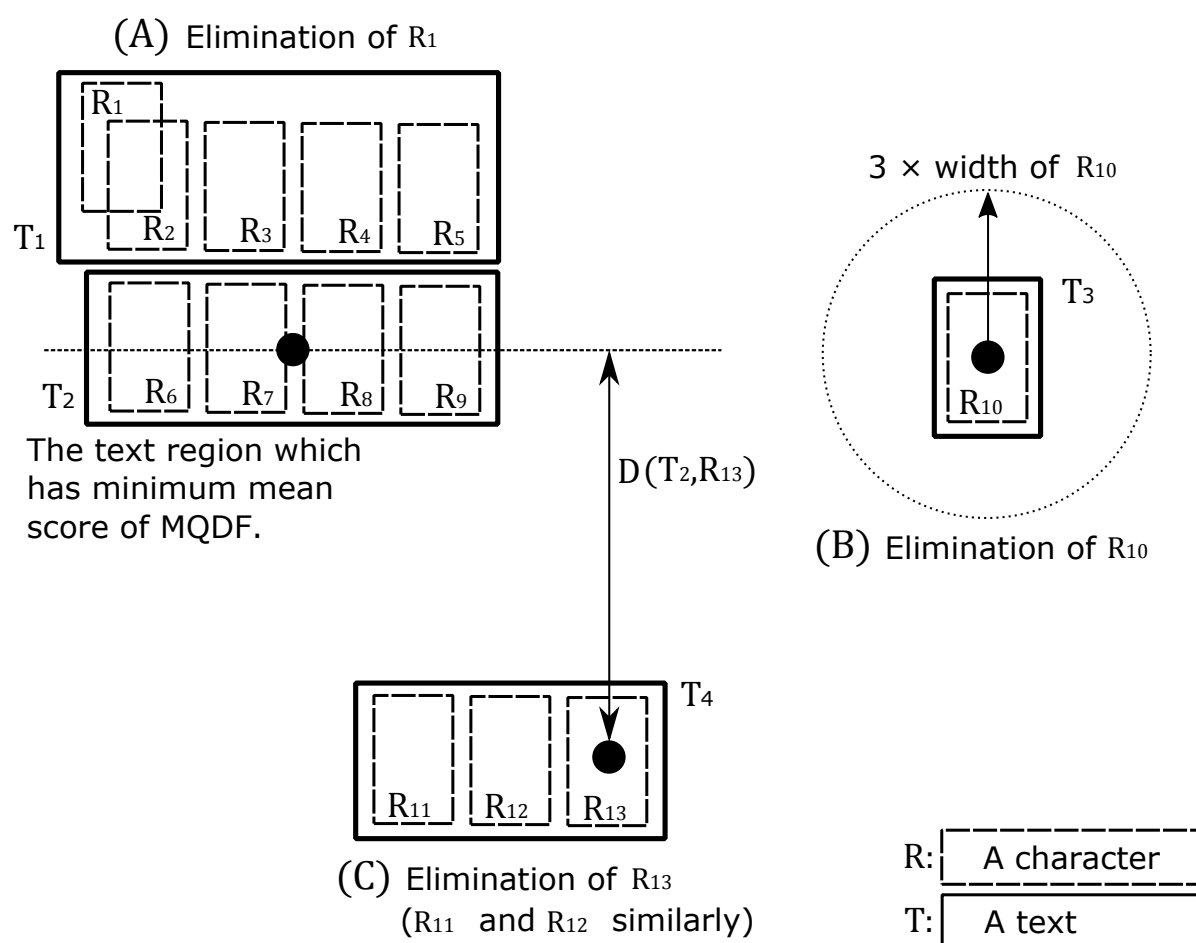


図 3.14: 文字領域の位置関係を用いた棄却

第 4 章

性能評価実験

4.1 実験内容

4.1.1 実験用画像

評価用にはドット文字が印字された製品を撮影した 250 枚の画像を用いる。画像サイズは 640×480 であり，すべての画像に製品情報を表すためのドット文字が必ず印字されている。印字面の素材，印字される位置，撮影した距離は図 4.1 に示すように様々である。これらの画像に対して，提案手法がドット文字のみを抽出して認識する性能を評価する。

また本実験では，これら 250 枚の画像に対して，実応用上発生し，自動認識の精度を下げる原因となる様々な変化を人工的に生成して，提案手法の頑健性を検証する。変化を与えていない画像を原画像と表記する。

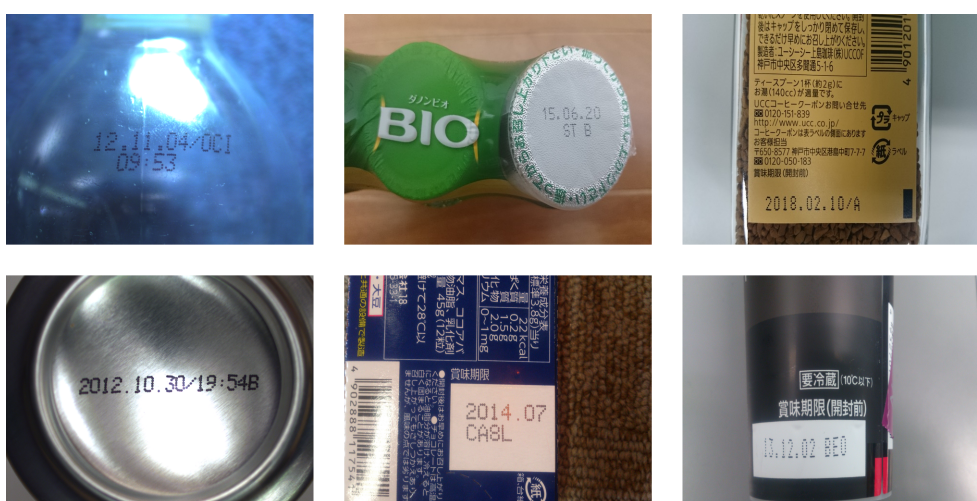


図 4.1: 実験用画像例

4.1.2 評価基準

文字の抽出について、目視で設定した正解座標と比較し、再現率 (Recall) と精度 (Precision) を評価する。また、正しい位置で抽出された文字について、認識率 (Recognition Rate) を評価する。それぞれ式 4.1, 4.2, 4.3 で定義される。「正しい位置」とは、矩形で与えられた正解領域との重なり率が 60%以上となる位置である。F-score は Recall と Precision の調和平均であり、式中の TP, FN, FP, RR は以下のように定義される。

- TP ... 正しい位置で抽出されたドット文字の数
- FN ... 抽出されなかったドット文字の数
- FP ... 抽出された非ドット文字領域の数
- RR ... 正しい位置で正しく認識されたドット文字の数

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.2)$$

$$\text{Recognition Rate} = \frac{\text{RR}}{\text{TP}} \quad (4.3)$$

さらに、再現率と認識率について、汎用文字認識システムとして知られる Tesseract-OCR^{*2} [39] と比較する。Tesseract-OCR はドット文字のみを認識対象とするシステムではないため、Precision に関しては無視する。

^{*2} <https://github.com/tesseract-ocr>, HP Labs. (between 1985 and 1995), Google Inc. (since 1995)

4.1.3 画像の回転に対する頑健性検証

実際の利用では、回転した位置から文字列を撮影する場合がある。このため、ドット文字の OCR は文字列の回転に対する頑健性が必要である。本実験では、回転に対する提案手法の頑健性について、原画像に回転を付与した画像を用いて評価する。回転は r_n で表され、 n は度数法で表現された角度である。本実験では r_{15}, r_{30}, r_{45} を ImageMagick ^{*3} による画像の回転を用いて生成し、性能評価結果を原画像と比較して性能の低下を検証する。生成した回転画像を図 4.2 に示す。



(a) 原画像

(b) 15 度回転 (r_{15})(c) 30 度回転 (r_{30})(d) 45 度回転 (r_{45})

図 4.2: 回転による画像の変化

^{*3} <http://www.imagemagick.org/script/index.php>, ImageMagick Studio LLC

4.1.4 モーションブラーに対する頑健性検証

実際の利用では，高速な生産ライン上の製品をカメラで撮影した時や，手でカメラを持った時のブレが原因となり，自動認識の性能を下げるモーションブラーが発生する．このため，ドット文字の OCR はモーションブラーに対する頑健性が必要である．本実験では，モーションブラーに対する提案手法の頑健性について，原画像にモーションブラーを付与した画像を用いて評価する．モーションブラーを生成するため， 1×3 画素の水平方向の平均移動フィルタを原画像に複数回適用する．モーションブラーの強度は b_n で表され， n は画像中の文字幅の平均ピクセル数 w に対し，フィルタリングの影響を受けて平滑化されるピクセル数の割合 (%) である．モーションブラー生成のためのフィルタリング回数は， $(w \times n/100)$ で求められる．本実験では， b_5, b_{10}, b_{15} の画像を生成し，性能評価結果を原画像と比較して性能の低下を検証する．生成したモーションブラーによる文字列の変化を図 4.3 に示す．

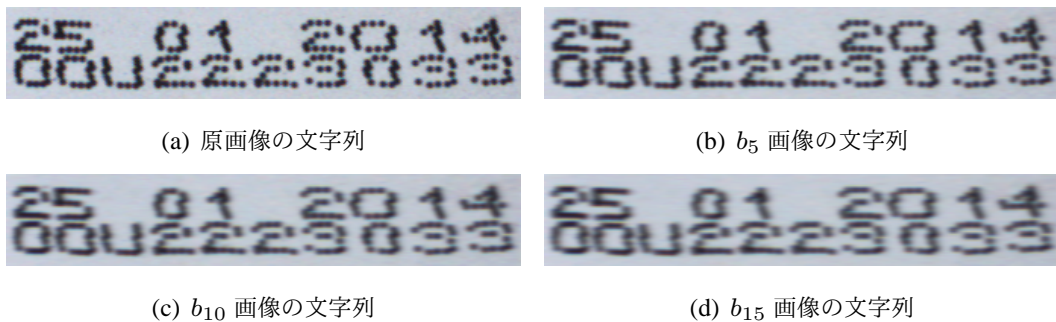


図 4.3: 生成したモーションブラーによる文字列の変化

4.1.5 実行時間測定

OCR にとって実行時間は重要である。本実験では、手法全体について、それを構成する以下のプロセスそれぞれに分けて実行時間を測定する。実行に使用した CPU は、Intel Core i7-2600 CPU (3.40 GHz) である。

処理 1 ドット文字列候補領域抽出 (3.2.1 - 3.3.1)

処理 2 前処理と分離候補位置設定 (3.4.1 - 3.5.1)

処理 3 濃度こう配特徴抽出 (3.5.2)

処理 4 MQDF による文字認識と非文字成分棄却 (3.5.2 - 3.6.3)

実験には 250 枚の原画像、回転を含む画像、モーションブラーを含む画像を用い、それぞれの平均実行時間を測定する。

4.1.6 評価対象を変化させた実験

性能の安定性を検証するため、250 枚あるデータセットからランダムに 100 個を取り出して、Recall, Precision, 実行時間を計測する。本実験を 50 回繰り返し、結果の変化を検証する。また、実行時間と文字の抽出と認識性能の関係性を検証する。

4.1.7 MQDF 評価値の効果検証

従来から、MQDF 評価値を OCR の非文字領域棄却に用いる研究 [40] や、印字不良を検出するために用いる研究 [41] がなされているが、その有効性は様々である。本手法で、3.6 節の非ドット文字成分の棄却条件 4,5,6,8 に MQDF 評価値を用いており、この有効性を確認するため、非ドット文字成分の棄却で MQDF 評価値を使用しなかった場合と性能評価結果を比較する。

4.2 結果

4.2.1 性能評価結果

性能評価実験結果を表 4.1 に示す．実行時間の測定結果を表 4.2 に示す．Tesseract-OCR では，Recall が 9.48%，認識率は 77.01%である．提案手法では，原画像についてドット文字抽出の Recall は 78.60%,Precision は 76.03%であり，ドット文字の抽出と認識に対して有効である．

実行時間は全体で 2397[ms] である．モーションブラーを付与した画像では，処理 1 で抽出するドット文字候補点が少ないため，実行時間が減少する．また，回転を付与した画像では，回転画像の生成によって画像サイズが大きくなるため，実行時間が増加する．

表 4.1: 性能評価実験結果

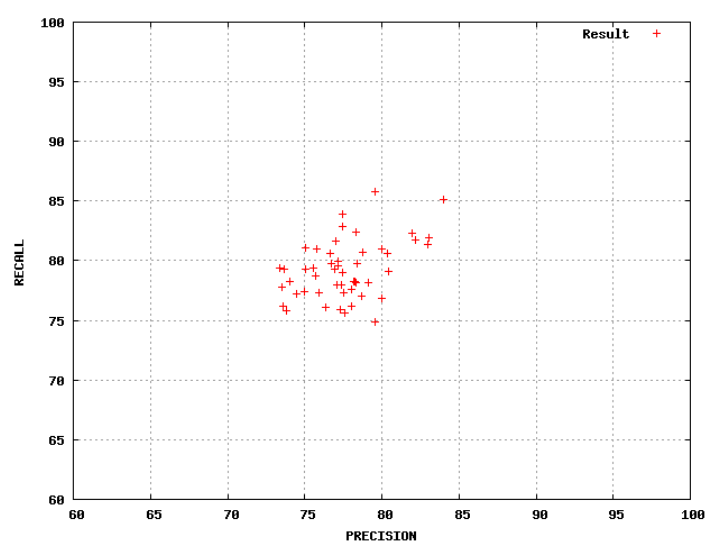
手法	変化	Recall(%)	Precision(%)	Recognition rate(%)
Tesseract-OCR	原画像	9.48	-	77.01
提案手法	原画像	78.60	76.03	94.43
	r_{15}	80.53	73.96	93.64
	r_{30}	77.67	76.03	91.22
	r_{45}	80.26	77.47	93.82
	b_5	78.20	72.54	93.47
	b_{10}	75.27	70.18	92.16
	b_{15}	68.30	63.91	88.04

表 4.2: 実行時間

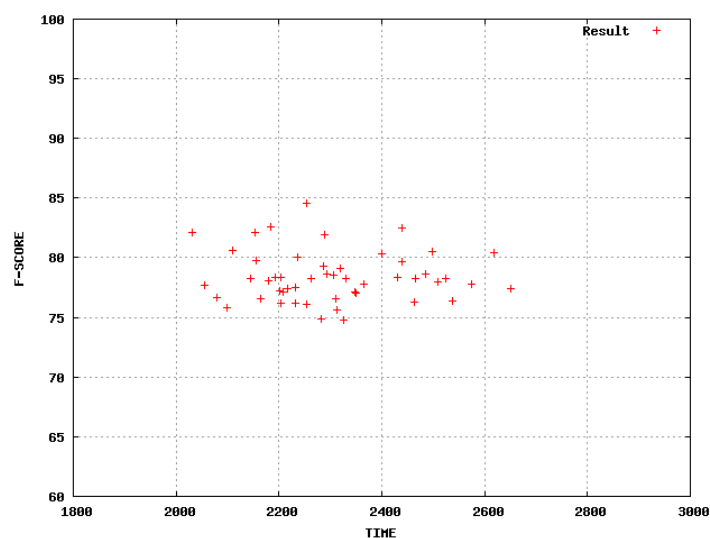
変化	処理 1 [ms]	処理 2 [ms]	処理 3 [ms]	処理 4 [ms]	全体 [ms]
原画像	309	489	1502	97	2397
r_{15}	313	502	1585	114	2514
r_{30}	311	511	1692	109	2623
r_{45}	317	518	1659	109	2603
b_5	300	469	1478	97	2344
b_{10}	294	482	1454	97	2327
b_{15}	284	474	1438	90	2286

4.2.2 評価対象を変化させた実験結果

ランダムに選ばれた 100 枚の画像による実験を 50 回繰り返した結果を図 4.4(a), 図 4.4(b) に示す. Recall の平均は 79.25%, 分散は 2.45, Precision の平均は 77.56%, 標準偏差は 2.56 である. F-score が最低となる結果でも Recall は 75.77%, Precision は 73.79% である. 撮影対象の製品が変わったり, 印字位置ずれ, 角度の変化によって性能が低下する従来のドット文字 OCR と比較して, 本手法は様々な製品上, 様々な撮影条件でのドット文字 OCR において有効である.



(a) Recall と Precision



(b) F-score と実行時間

図 4.4: 評価対象を変化させた実験結果

4.2.3 MQDF 評価値の効果

非ドット文字領域の棄却で MQDF 評価値を用いない場合の実験結果を表 4.3 に示す. 表 4.1, 表 4.3 を見ると, MQDF 評価値を用いることで, 原画像に対する実験において, Recall は 6.52% 低下したが, Precision が 49.89% 向上していることがわかる.

表 4.3: MQDF 評価値を用いない場合の実験結果

変化	Recall(%)	Precision(%)	Recognition rate(%)
原画像	85.12	26.14	94.43
r_{15}	86.09	23.44	93.58
r_{30}	83.95	24.56	90.95
r_{45}	85.83	27.18	92.80
b_5	83.37	25.29	93.64
b_{10}	82.80	22.90	91.22
b_{15}	76.50	22.52	88.02

第 5 章

考察

5.1 文字抽出に関する考察

5.1.1 FAST アルゴリズムの有効性

近年、情景内文字抽出において、領域分割手法である **MSER** を用いる手法 [8][9] が有効とされている。図 5.1(a) は **MSER** による領域分割結果である。異なるクラスタに分割された領域が異なる色で示されており、それぞれの領域を抽出することで、文字を抽出することが可能である。しかし、ドット文字はドットマトリクスで構成されるため、文字部分が正しく分割されておらず、**MSER** によってドット文字を抽出することは困難である。様々な字種に対して適用可能な手法として、コーナー検出による文字抽出手法 [12] が提案されており、本手法では文字列の検出にコーナーを用いた。図 5.1(b), 図 5.1(c), 図 5.1(d) に示すとおり、コーナーはドット文字上に存在するため、ドット文字はコーナー検出によって抽出可能である。

代表的なコーナー検出手法として、最小固有値法 [42], Harris 検出器 [43], FAST アルゴリズムが存在する。ここで、画像 250 枚を用いた性能評価実験の結果から、ドット文字候補点の抽出に最適なコーナー検出手法を検討する。画像から、それぞれの手法を用いて 500 個のコーナーをドット文字候補点として抽出した場合の OCR 性能を比較する。ただし、最小固有値法と Harris 検出器はコーナーの検出個数を指定できるが、検出個数を指定できない FAST アルゴリズムは、3.2.1 節で記した手法で設定値を調整し、500 個に最も近く、かつ 500 個以下の数だけコーナーを検出する。

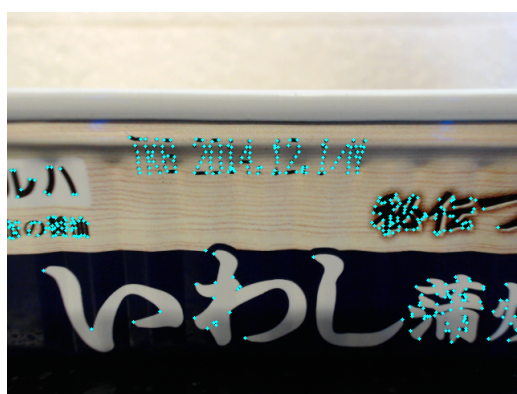
評価結果を表 5.1 に示す。FAST アルゴリズムは比較したコーナー検出手法で、最もドット文字候補点抽出に有効である。また、コーナー検出に要する時間に関しても、これら 3 手法の中で最も高速である。



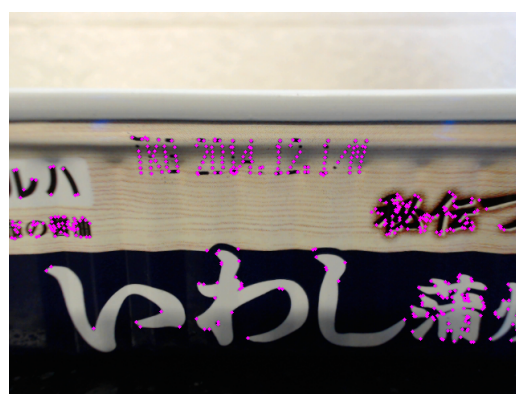
(a) MSER による候補領域抽出



(b) 最小固有値法によるコーナー検出



(c) Harris 検出器によるコーナー検出



(d) FAST アルゴリズムによるコーナー検出

図 5.1: ドット文字抽出のための手法

表 5.1: OCR 性能に対するコーナー検出手法の比較

手法	Recall(%)	Precision(%)	実行時間 [ms]
最小固有値法	69.86	69.56	15.44
Harris 検出器	77.37	73.45	14.14
FAST アルゴリズム	78.60	76.03	6.76

5.1.2 文字列候補領域抽出アルゴリズムの設定値

提案手法は，文字候補点をクラスタリングすることで文字列候補領域を抽出する．本クラスタリング手法は，分類する候補点の数が n 個の時，計算量 $O(n^2)$ で実行できるため階層的クラスタリングよりも高速であり，分類先クラス数を指定する必要が無いため，候補点からの文字列抽出に適している．候補点を分類するための評価式に，任意に設定した値 m が存在し， m を変化させてクラスタリングを複数回繰り返し，全ての結果を組み合わせることで，文字認識のために必要となる，文字列抽出の再現性を向上させている． m は離れた位置にある 2 個の候補点が，同一クラスへ分類される程度を示すパラメータであり，適切な m の値は製品を撮影した距離によって変化するため，画像によって様々である．ここで，250 枚の評価用データセットを 50 枚ずつ DATA1 から DATA5 の 5 組に分け，それぞれ m を表 5.2 のように変化させたときの OCR 性能を 15 パターン検証し，適切な m の値と組み合わせを検証する．表中の ○ で分類を実行する．

表 5.2: クラスタリングの設定値 m の変化

テスト番号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$m = 8$	○					○				○			○		○
$m = 16$		○				○	○			○	○		○	○	○
$m = 24$			○				○	○		○	○	○	○	○	○
$m = 32$				○				○	○		○	○	○	○	○
$m = 40$					○				○			○		○	○

DATA1 から DATA5 の実験結果と、その平均値を図 5.2 に示す．組み合わせ 1 個では，OCR 性能が低く，評価用データセットの違いによって性能が様々に変化する．3 通りまでは，異なる m を組み合わせて使用することで，文字列抽出の再現性が上がり，OCR 性能の F-score が上がる．それ以上に組み合わせを増加させても文字列抽出の再現性は向上せず，分類回数が多いことが原因となり抽出精度が低下するため，OCR 性能が低下する．比較した中では $m = (16, 24, 32)$ の 3 回が文字列抽出用クラスタリングのために最適なパラメータの組み合わせである．

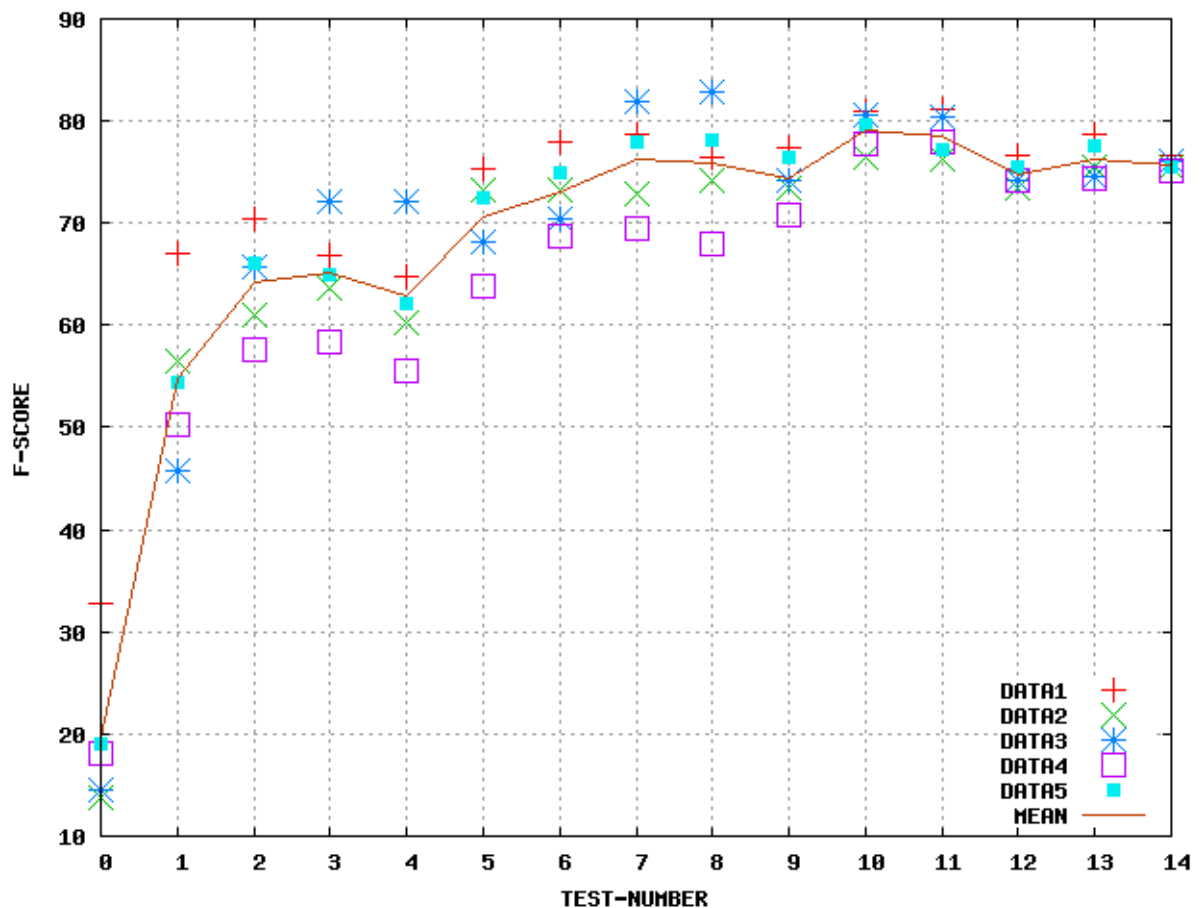


図 5.2: 文字列抽出のパラメータ m の値と組み合わせを変化させた実験の結果

5.2 文字認識に関する考察

5.2.1 変動モデル学習の有効性

変動モデル学習について，文字を手動で切り出した画像 2030 サンプルを用いた先行研究 [20] での実験により，本稿で提案した各変動モデル (a) から (d) の適用前と適用後で認識率の向上が確認されている．認識率が上がるにつれて，MQDF 評価値であらわされる「文字らしさ」も正確に算出されることが確認されている [45]．

- (a) 複数のドット文字フォント
- (b) 3 次元回転処理
- (c) ドット直径の変更
- (d) ドット抜け

変動モデルの組み合わせによる認識率の変化を図 5.3 に示す．図中の ○ で示された変動モデルを適用した結果である．

変動モデルを全く適用しない状態とは， 5×7 ドット構成のみを学習した辞書である．本手法はドット文字認識のための学習辞書作成に有効な手段であり，さらにモーションブレンダーなど新たな変動モデルを組み合わせることで更なる性能向上が期待できる．また，学習サンプルを増加すると学習にかかる時間も増加するが，文字認識の実行時間は特徴ベクトルの次元数と認識対象クラス数のみに依存するため変化しない．

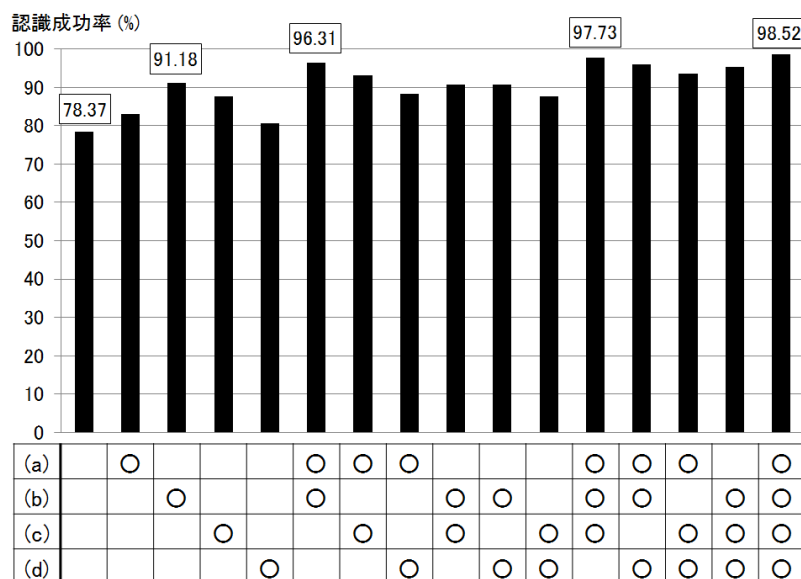


図 5.3: 変動モデルの追加による認識率の向上

5.2.2 非文字領域棄却における Precision の向上

非文字領域の棄却は，提案手法の Precision に最も影響し，本手法では，文字領域の画素値とアスペクト比，MQDF 評価値，文字領域の位置関係を用い，以下の 8 つの条件によって非文字領域を棄却した．棄却能力の低い条件を複数連結することで非文字領域の棄却を実現している．条件 6,7,8 の状況を図 5.4 に示す．また，非文字領域棄却手法の詳細は 3.6 節に記述されている．

1. ドット文字領域の 90% 以上が白画素である．
2. ドット文字領域の 90% 以上が黒画素である．
3. ドット文字領域の幅が高さの 2 倍以上大きい．
4. ドット文字領域の MQDF 評価値が 600 以上である．
5. ドット文字領域が所属するドット文字列候補領域の平均 MQDF 評価値が 400 以上である．
6. 複数の文字領域の面積重なり率が 20% 以上の場合で，MQDF 評価値が最小ではない領域．(A)
7. 文字領域の幅の 3 倍以内の距離に他の文字領域が存在しない．(B)
8. $D(T_{min}, R_i)$ が文字領域の幅の 3 倍以上である．(C)

ここで，それぞれの条件について，Recall と Precision の変化を確認し，有効性を考察する．変化を表 5.3 に示す．

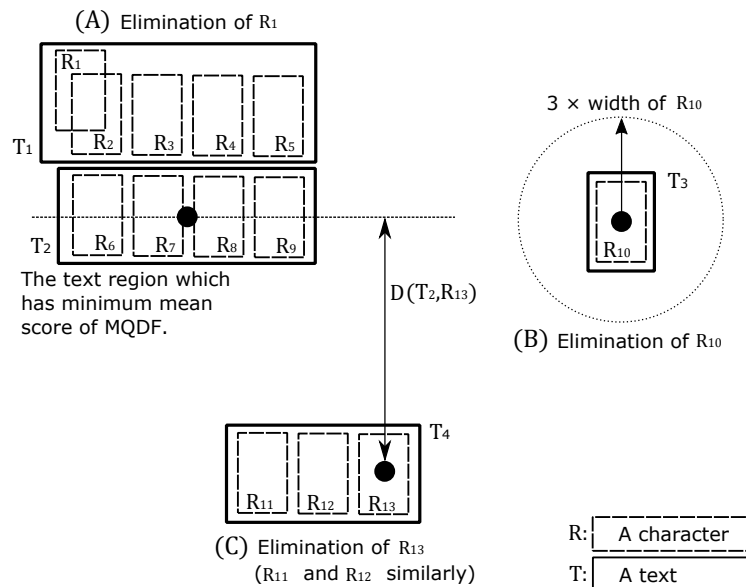


図 5.4: 文字領域の位置関係を用いた棄却

表 5.3: 非文字領域棄却による Recall と Precision の変化

条件	Recall(%)	Precision(%)	Recall 変化 (%)	Precision 変化 (%)
棄却前	86.83	15.41	N/A	N/A
1 (適用後)	86.83	18.15	0.0	+ 2.74
2	86.83	18.36	0.0	+ 0.21
3	86.83	23.75	0.0	+ 5.39
4	85.81	35.59	- 1.02	+ 11.84
5	83.80	49.09	- 2.01	+ 13.50
6	81.92	58.54	- 1.88	+ 9.45
7	81.55	62.00	- 0.37	+ 3.46
8	78.60	76.03	- 2.95	+ 14.03

1. 文字領域の画素値とアスペクト比

本条件を満たす文字領域の多くは、MQDF 評価値を用いた条件によって棄却される。しかし、領域の大部分が黒画素、白画素で占められていても、MQDF 評価値が小さく、1, L, T として認識される場合がある。これらの領域は、抽出した非文字領域に対しても、文字列領域と同じように、文字列から 1 文字単位に分離処理をすることが原因となり発生する。本条件は、このようにして発生した、小さな MQDF 評価値を持つ非文字領域の棄却に有効である。

2. MQDF 評価値

Recall の低下に影響したドット文字領域は、二値化後の画像の状態が悪く、適切な MQDF 評価値を取得できなかった領域である。非文字領域 1 個が学習した文字に偶然類似していることもあるため、Precision の向上には条件 5 の寄与が大きく、複数の領域の平均 MQDF 評価値の利用が有効である。

3. 文字領域の位置関係

すべての条件の中で、条件 8 は最も Precision の向上率が高い。複数の領域の平均 MQDF 評価値と他の文字領域との距離を同時に利用することで、最も多くの非文字領域を棄却することができる。ただし、非文字領域棄却に条件 8 のみを用いた場合、Recall は 79.14%、Precision は 63.14%である。抽出された文字領域が極端に多い場合、良好な MQDF 評価値を持つ非文字領域が多く存在し、位置関係を適切に推定できないため、性能が大きく向上しない。条件 8 で最良の MQDF 評価値を持つ文字列領域 T_{min} を決定するために、他の棄却条件によって十分に Precision を上げておくことが重要である。

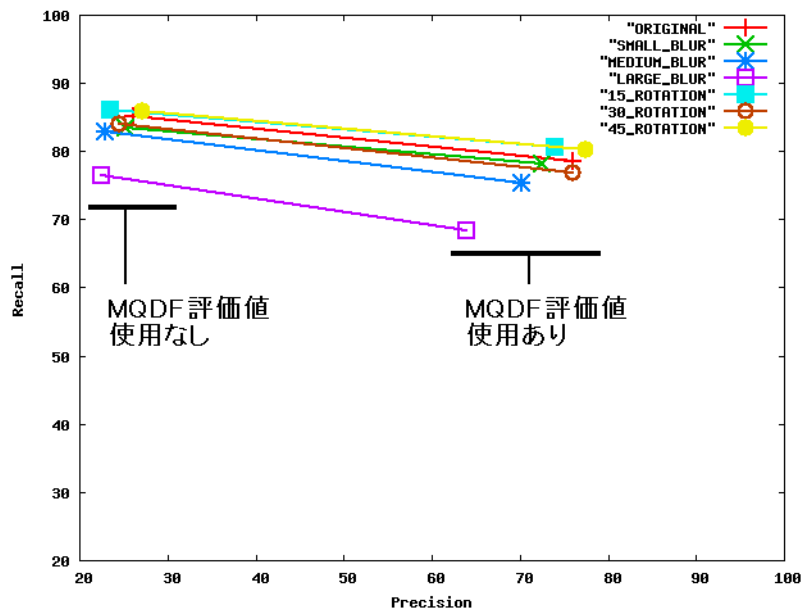
5.2.3 MQDF 評価値の効果

非文字領域棄却に MQDF 評価値を用いた場合と用いなかった場合の, Recall と Precision の変化を図 5.5(a) に示す. 認識率と F-score の変化を図 5.5(b) に示す. 図中で, 原画像, モーションブラーを含む画像, 回転を含む画像に対する実験の結果は, それぞれ以下のとおり示されている.

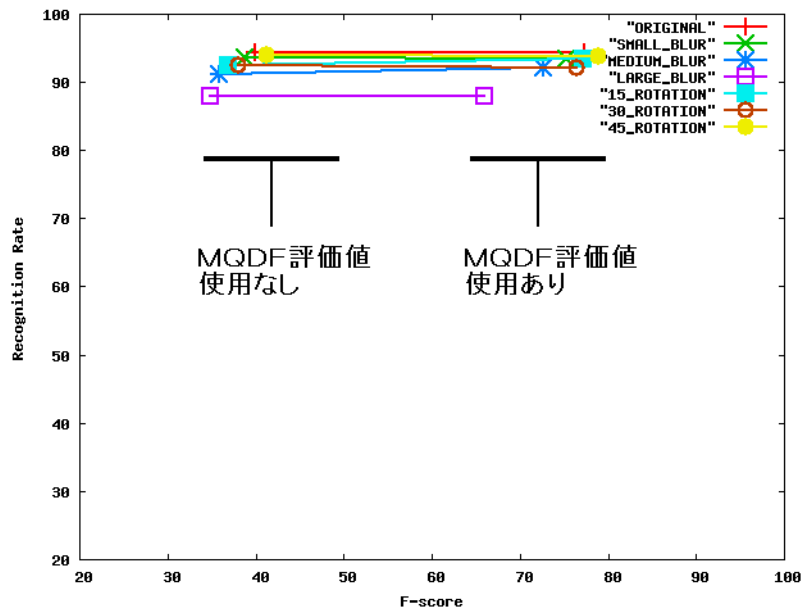
原画像	:	ORIGINAL	(赤)
b_5 画像	:	SMALL BLUR	(緑)
b_{10} 画像	:	MEDIUM BLUR	(青)
b_{15} 画像	:	LARGE BLUR	(紫)
r_{15} 画像	:	15 ROTATION	(水)
r_{30} 画像	:	30 ROTATION	(茶)
r_{45} 画像	:	45 ROTATION	(黄)

図より, MQDF 評価値を用いない場合, Precision が大きく低下する. 非ドット文字領域の棄却に MQDF 評価値を利用することは, ドット文字の抽出に有効である. さらに, 非文字領域棄却に MQDF 評価値を用いた場合, 抽出した文字の認識率が向上することから, MQDF 評価値を用いた非文字領域棄却によって, 正しく認識された文字のみを抽出できることがわかる.

正しい位置で抽出された文字について, MQDF 評価値と認識率の関係を図 5.6, 詳細を表 5.4 に示す. MQDF 評価値が高い領域は認識率が低く, MQDF 評価値が低い領域は認識率が高い. このことから, MQDF 評価値は正しく文字を認識する指標となるため, 学習辞書に登録された「文字らしさ」としての利用に適し, 非文字領域の棄却に有効である.



(a) Recall と Precision の変化



(b) 認識率と F-score の変化

図 5.5: 文字認識に MQDF 評価値を用いたときの性能変化

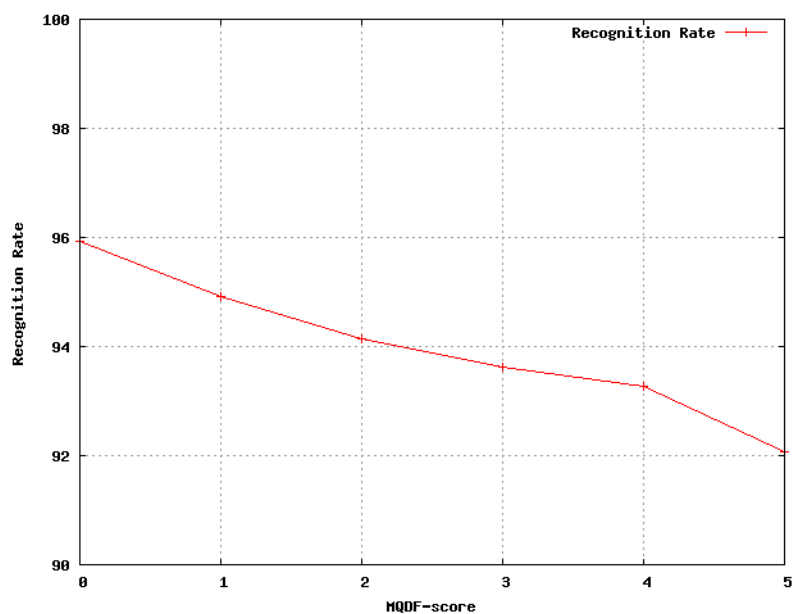


図 5.6: MQDF 評価値と認識率の関係

表 5.4: MQDF 評価値と認識率の詳細

MQDF 評価値	100 未満	200 未満	300 未満	400 未満	500 未満	500 以上
文字認識率 (%)	95.92	94.91	94.15	93.63	93.26	92.07
抽出された文字数	3140	3476	3743	3925	4048	4277
認識成功文字数	3012	3299	3524	3675	3775	3938

5.3 実行結果に関する考察

5.3.1 回転に対する頑健性

画像を回転させたときについて、どの角度でも性能は大きく低下しない。これは提案手法が、ドット文字候補点の抽出で回転の影響を受けにくいコーナー検出を用いていること、ドット文字列候補領域の抽出後、文字列の回転角度を推定し、アフィン変換を用いて0度に修正して認識すること、文字列の回転角度を非ドット文字領域の棄却を行っていることなどから、回転に対して頑健であることを示している。

5.3.2 モーションブラーに対する頑健性

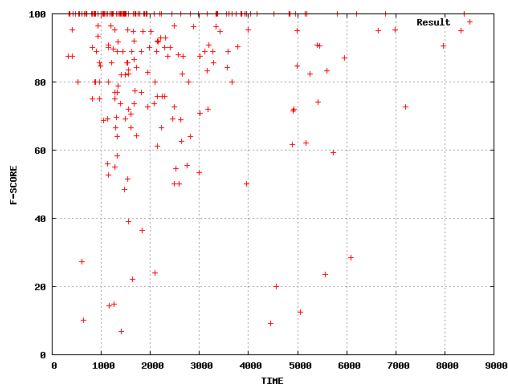
わずかなモーションブラーによって、抽出性能や認識率が急激に低下することは無い。しかし、 b_{10} までの変化と比較して、 b_{10} と b_{15} の間でドット文字抽出性能低下が顕著である。モーションブラーによる性能低下の原因は、コーナーの検出性能によるものである。モーションブラーの影響が大きいときは、ドット文字候補点を抽出する時にコーナーが検出できなくなるためである。また、モーションブラーによって文字中のドットマトリクスの消失やドットマトリクスの横方向の結合によって字形が変化するため、性能が低下する。変動モデル学習によって学習された以上の変化が発生した場合、学習辞書作成の再検討、または二値化性能の向上が課題となる。

5.4 実行時間に関する考察

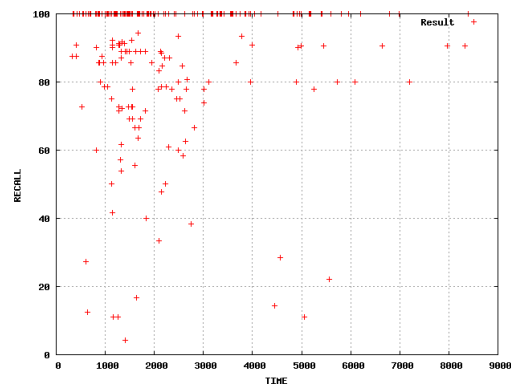
5.4.1 実行時間と性能の関係

画像 250 枚を用いた実験結果から，実行時間と性能の関係を検証する．実行時間と F-score の関係を図 5.7(a) に，Recall との関係を図 5.7(b) に，Precision との関係を図 5.7(c) に示す．実行時間と各スコアの相関係数は，F-score の場合は 0.06，Recall の場合は -0.09，Precision の場合は -0.04 であり，OCR の性能との関連性は低い．

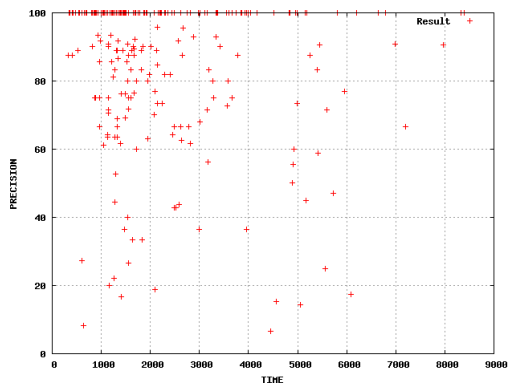
非文字領域棄却前の Precision と実行時間の関係を図 5.7(d) に示す．このとき，実行時間と Precision の相関係数は -0.53 であり負の相関関係にある．実験の結果，実行時間の多くが特徴抽出であったが，非文字領域棄却前に高い Precision でドット文字を抽出できる画像は実行時間が短いことから，実行時間の多くは非文字領域の特徴抽出によるものであることがわかる．



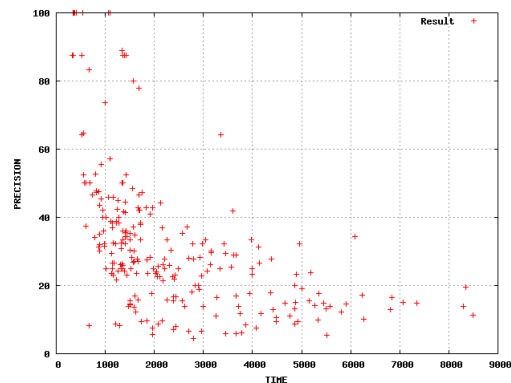
(a) F-score と実行時間の関係



(b) Recall と実行時間の関係



(c) Precision と実行時間の関係



(d) 非文字領域棄却前の Precision と実行時間の関係

図 5.7: 実行時間と性能の関係

5.5 産業用 OCR としての実応用に向けて

5.5.1 性能向上

強いモーションブラーを受けた場合の性能低下に対して，さらなる頑健性向上のためには，十分な数のコーナーが検出できなかった場合，モーションブラーを補正する前処理によって検出性能を向上させる必要がある。

文字が抽出できない原因の多くは，二値化によって文字と背景が分離されないことや，行の検出に失敗することである．これらの課題に対して，文書用の二値化手法 [46][47]，文字列の二値化手法 [48][49]，二値化を用いない文字認識手法 [50] など様々な手法が提案されているが，様々な製品上に印字されたドット文字の抽出と認識に対して提案された手法ではないため，手法の改善が必要である。

5.5.2 実行時間改善

実行時間の大部分は，非文字領域の特徴抽出で占められているため，LBP 特徴 [51] など，濃度勾こう配特徴以外的高速な特徴抽出手法を用いて短縮することが可能である．ただし，文字列を 1 文字単位で分離し，非文字領域の棄却をするために，抽出した特徴を用いて高い認識率と良好な MQDF 評価値を得る必要がある。

5.5.3 文字列の位置を設定した場合の性能向上

現在、高い性能を示している OCR ソフトウェアでは、認識対象フォント、文字の印字位置、光源や材質の指定など、様々な詳細情報を登録することで性能を向上させている。本手法は、これらの情報を事前に登録することなく Recall 78.60%, Precision 76.03% の文字抽出率と認識率 94.43% の文字認識が得られた。本手法に、以下のような簡単な情報を登録することによる、さらなる性能向上の可能性を検証する。

1. ドット文字列が印字される領域として、1 箇所の座標、幅、高さ (x, y, w, h) を事前に与え、文字列領域として抽出する。
2. ドット文字が印字される領域として、文字数分の座標、幅、高さ (x_i, y_i, w_i, h_i) を事前に与え、文字領域として抽出する。

結果を表 5.5 に示す。文字抽出の改善によって、文字認識以降の処理を改善することなく最大で Recall 99.94%, Precision 100.0% の性能を引き出すことが可能であった。文字認識率は先行研究において 98.52% であるが、本実験では最大 96.89% である。本手法では二値化を自動化したため、強い照明変化のある環境下で文字と背景を二値に分離できなかったことが性能低下の原因である。

表 5.5: 事前情報の登録による性能向上

登録情報	Recall(%)	Precision(%)	認識率 (%)	実行時間 [ms]
登録なし	78.60	76.03	94.43	2397
文字列の位置	90.01	90.07	96.89	463
文字の位置	99.94	100.0	96.41	350

第 6 章

結論

6.1 本研究のまとめ

6.1.1 提案手法

本論文では，以下のプロセスによって，ユーザーがシステムに介入することなく，カメラで撮影された画像からドット文字のみを抽出して認識する手法を提案した．ドット文字列は FAST アルゴリズムによって検出されたコーナ点のクラスタリングによって抽出される．ドット文字の認識には，変動モデル学習によって作成された学習辞書，特徴量として文字の濃度こう配特徴，識別関数として Modified Quadratic Discriminant Function (MQDF) を用いた．

1. ドット文字候補点抽出
2. ドット文字列候補領域抽出
3. 前処理（二値化，行間検出，膨張処理）
4. ドット文字の抽出と認識評価
5. 非ドット文字領域棄却

6.1.2 提案手法の有効性

ドット文字が印字された製品を撮影した画像 250 枚を用いて、提案手法の文字抽出と文字認識の性能を評価した。汎用文字認識システムとして知られる Tesseract-OCR では、ドット文字抽出の Recall は 9.48%，抽出した文字の認識率は 77.01% であり、ドット文字の OCR 性能として不十分である。

提案手法では、画像から文字抽出する Recall は 78.60%，Precision は 76.03%，正しい位置で抽出された文字の認識率は 94.43% である。250 枚の中からランダムで選ばれた 100 枚の画像による実験を 50 回繰り返した結果、F-score が最低となる結果でも Recall が 75.77%，Precision が 73.79% であり、様々な製品上に印字されたドット文字の抽出と認識に有効であることがわかった。

6.1.3 ドット文字列抽出

ドットマトリクスはコーナー検出によって検出可能であり、検出したコーナーをドット文字候補点として抽出する。ドット文字候補点を近傍画素の輝度値を用いてクラスタリングすることで、ドット文字列を抽出する。

ドット文字候補点の抽出に使用するコーナー検出手法について、最小固有値法，Harris 検出器，FAST アルゴリズムによるコーナー検出を比較すると，FAST アルゴリズムを用いた場合の OCR 性能が最も高く，実行が高速である。

ドット文字候補点からドット文字列候補領域を抽出するため，クラスタリング手法を提案した。提案手法は分類する候補点数が n 個のとき，計算量 $O(n^2)$ で計算可能であるため，階層的クラスタリングよりも高速であり，分類先クラスタ数を指定する必要が無いため，候補点からの文字列抽出に適している。文字認識のために必要である文字列抽出の再現性を向上させるために，評価式中の任意に設定する値 m を適した組み合わせで変化させ，クラスタリングを複数回繰り返し，全ての結果を文字列として抽出する。 m の値の組み合わせを変化させた実験を行った結果， $m = (16, 24, 32)$ の組み合わせによる手法が最適であることが確認された。

6.1.4 ドット文字認識

文字認識のための特徴抽出には濃度こう配ヒストグラムを用い、学習辞書作成には変動モデル学習を用いた。抽出されたドット文字列候補領域を1文字ずつに分離してMQDF識別器を用いて認識した。さらに、MQDF評価値と文字の位置関係により非ドット文字領域を棄却した。

変動モデル学習では、複数のフォントで印字されたドット文字、3次元回転した文字、ドット直径が変更された文字、ドット抜けを含む文字を生成し、それぞれを追加することに文字認識率が向上した。

全てのドット文字候補領域に8つ棄却条件で判定を行い、非ドット文字領域を棄却する。棄却能力の低い条件を複数連結させることで、Recallは86.83%から78.60%に、8.23%低下したが、Precisionは15.41%から76.03%へと、60.62%向上した。MQDFの出力結果（MQDF評価値）を非文字領域棄却に用いない場合と比較してMQDF評価値を棄却に用いる場合はRecallは6.52%低いがPrecisionは49.89%高く、MQDF評価値は非文字成分棄却に有効であることを示した。

MQDF値が低い文字領域は文字認識成功率が高いことを確認し、MQDF評価値は学習辞書に登録された「文字らしさ」としての利用に適し、非文字成分の棄却と文字の分離位置決定に有効であることを示した。

6.1.5 実応用に向けた実験と結果

実際にOCRシステムを利用する環境下で発生し、自動認識の性能を下げる原因となる文字列の回転とモーションブラーを画像に与えて実験した結果、提案手法では、文字列の回転と文字の10%が影響を受けるモーションブラーに対して頑健性があることを確認した。Intel Core i7-2600 CPUを使用した実行環境下で、実行時間の全体平均は画像一枚あたり2397[ms]であった。このうち特徴抽出に要する時間は1502[ms]であり、その多くが非文字成分の特徴抽出に要する時間であった。

6.1.6 性能向上に関する提案

ユーザが文字列の座標のみを情報登録し、本手法で文字列の座標情報を用いると、Recall 90.01%, Precision 90.07%での文字抽出が可能であることから、今後、文字認識手法を変更することなく、性能と実行速度を向上させることができる。抽出や認識に失敗する原因の多くは、文字認識の前処理での二値化と行間検出の性能が十分でないことであり、前処理を改善することが課題である。実行時間の短縮には、濃度こう配ヒストグラム以外の高速な特徴抽出手法を検討する必要がある。

謝辞

本論文は様々な方のご支援をいただき執筆することができました。本研究に関し、研究に対する基本姿勢，様々な専門知識と技術，熱心なご指導，ご指摘をしてくださいました木村文隆教授，若林哲史准教授，大山航助教，三宅康二名誉教授に深く感謝申し上げます。また，短期留学中に研究の助言をしてくださった Yi Lu Murphey 教授，Malayappan Shridhar 教授に心より感謝いたします。日頃お世話になった田中みゆき事務官，研究室の先輩方，特に研究を多方面から支えてくださった遠藤浩士先輩にお礼申し上げます。そして，大学院生として充実した研究期間を共に過ごした三重大学大学院情報工学専攻の皆様感謝します。最後に，私の学生生活を最後まで支えていただいた家族に感謝を表し，本論文の結びと致します。

参考文献

- [1] “農林物資の規格化及び品質表示の適正化に関する法律（昭和二十五年五月十一日法律第百七十五号）”
- [2] “インパクトプリンタ技術”，小森 智裕，日本画像学会誌 51(2)，pp.191-198，2012
- [3] 中島義明他編，「心理学辞典」，有斐閣，p.760，1999
- [4] Jian Liang, David Doermann, and Huiping Li, “Camera-based analysis of text and documents: a survey”, International Journal of Document Analysis and Recognition (IJ-DAR), Vol.7, No.2-3, pp.84-104, 2005
- [5] 黄瀬 浩一，大町 真一郎，内田 誠一，岩村 雅一，“カメラを用いた文字認識・文書画像解析の現状と課題”，IEICE technical report, PRMU2004-246, March 2005
- [6] 津雲淳，“文字認識研究の新たな展開に向けて”，信学技報，PRMU2007-226, 2008
- [7] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Sergi RoblesMestre, Joan Mas, David Fernandez Mota, Jon Almazan, Lluís Pere de las Heras, et al. ICDAR 2013 robust reading competition in 12th International Conference on Document Analysis and Recognition (ICDAR), pp.1484-1493, IEEE, 2013
- [8] Yin, X.C., Yin, X., Huang, K., Hao, H.W. “Robust text detection in natural scene images”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.36, Issue.5, pp.970-983, September 2013
- [9] Neumann, L., Matas, K. “Real-time scene text localization and recognition”, In: CVPR (2012)
- [10] B.Epshtein, E.Ofek, Y.Wexler, “Detecting text in natural scenes with stroke width transform”, CVPR 2010
- [11] W.Huang, Z.Lin, J.Yang, J.Wang, “Text localization in natural images using”, stroke feature transform and text covariance descriptors. In: ICCV (2013)
- [12] Q.Ye, D.Doermann, “Text Detection and Recognition in Imagery: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Volume:37, Issue:7, pp.1480-1500, 2014

- [13] 遠藤 浩士, 大山 航, 若林 哲史, 木村文隆 “ドット文字の生成と認識に関する考察”, IEICE technical report, 113(346), 133-137, 2013-12-12
- [14] Khader Mohammad, Sos Agaian “Practical Recognition System for Text Printed on Clear Reflected Material”, ISRN Machine Vision, Volume 2012, Article ID 253863, 2012
- [15] Berrin A. Yanikoglu, “Pitch-based segmentation and recognition of dot-matrix text”, Sabanci University, IJDAR, October 1999
- [16] Abderrahmane Namane, El Houssine Soubari, Patrick Meyrueis, “Degraded dot matrix character recognition using CSM-based feature extraction”, In proceedings of the 10th ACM symposium on Document engineering, pp.207-210, 2010
- [17] Yuning Du, Haizon Ai, Shihong Lao, “Dot Text Detection Based on FAST Points”, International Conference on Document Analysis and Recognition, 435-439, 2011
- [18] P.Shivakumara, D.D.Hong, D.Zhao, C.L. Tan and U.Pal, “A New Iterative-Midpoint-Method for Video Character Gap Filling”, ICPR2012, pp.673-676, 2012
- [19] 夏目祐輔, 菅谷至寛, 大町真一郎, “エッジ情報を用いた全方位カメラの映像中の文字列検出”, 信学技報, PRMU2011-106, 2011
- [20] K.Endo, W.Ohyama, T.Wakabayashi, F.Kimura, “Performance Improvement of Dot-Matrix Character Recognition by Variation Model based Learning” Proceedings of First International Workshop on Robust Reading (IWRR2014), 2014-11-02
- [21] R.Narita, W.Ohyama, T.Wakabayashi, F.Kimura “Three Dimentional Rotation-Free Camera-based Character Recognition” IEEJ Transactions on Electronics, Information and Systems 133(4), 876-882, 2013
- [22] 澤 和宏, 若林哲史, 鶴岡信治, 木村文隆, 三宅康二, “こう配特徴ベクトルと変動吸収共分散行列による手書き漢字認識の高精度化”, 信学論 Vol.J82-D-2, No.11, pp.1-12, 2001
- [23] K.Fukunaga, “Introduction to Statistical Pattern Recognition, Second Edition”, Academic Press, New York, 1990
- [24] 若林哲史, 鶴岡信治, 木村文隆, 三宅康二, “手書き文字認識における特徴量の次元数と変数変換に関する考察”, 信学論 (D-2), vol.J77-D-2, no.12, pp.2495-2503, 1993
- [25] F.Kimura, S.Tsuruoka, Y.Miyake, “Modified quadratic discriminant functions and the application to Chinese character recognition”, IEEE transaction on Pattern Analysis and Machine Intelligence, PAMI-9(1), pp.149-153, 1987
- [26] D.G. Keehn, “A note on learning for Gaussian properties”, IEEE Trans. Inform. Theory, vol.IT-11, no.1, pp.126-132, 1965
- [27] R.O. Duda and P.E. Hart, “Pattern Classification and Scene Analysis”, John Wiley &

- Sons Inc., 1973
- [28] Z. Yan, Z. Li, Z. Cairong, Y. Yinhua, H. Chengwei and W. Qingyun, "Modified quadratic discrimination function for non-normal distribution and its application in speech emotion recognition", Proc. APCCAS2008, pp.213-216, 2008
- [29] Y. Liu, T. Zhao and J. Zhang, "Learning multispectral texture features for cervical cancer detection", Proc. ISBI2002, pp.169-172, 2002
- [30] T. Kato and Y. Ninomiya, "An approach to vehicle recognition using supervised learning", IEICE Trans. Inform. SYST. vol.E83-D, no.7, pp.1475-1479, 2000
- [31] G. F. Hugs, "On the mean accuracy of statistical pattern recognizers", IEEE Trans. Info. Theory, vol.IT-14, no1, pp53-63, 1968
- [32] F. Kimura, S. Tsuruoka and Y. Miyake "On avoiding peaking phenomenon of the quadratic discriminant functions", Proc. ICPR1986, pp.275-278, 1986
- [33] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection", Computer Vision-ECCV 2006 pp.430-443, 2006
- [34] 福山翔平, 白川透, 上瀧剛, 内村圭一, "AR のための遺伝的アルゴリズムによる特徴点の自動選択", 火の国情報シンポジウム 2013 論文集, C-2-3, 2013
- [35] J. Canny, "A Computational Approach To Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-714, 1986
- [36] 加藤裕康, 大山航, 若林哲史, 木村文隆, "ナンバープレート認識を目的とした文字抽出のための文字列分割", 電気関連学会東海支部連合大会, 2008 号, O-357, 2008
- [37] Nobuyuki Otsu, "A threshold selection method from gray-level histograms", Systems, Man and Cybernetics, IEEE Transactions on Volume 9, Issue 1, 62-66, Jan 1979
- [38] 馬場口登, 塚本正敏, 相原恒博, "手書き日本文字列からの文字切り出しの基礎的考察", 電子通信学会論文誌, D 68 (12), No.4, pp.2123-2131, December 1985
- [39] Kay. Anthony "Tesseract: an Open-Source Optical Character Recognition Engine", Linux Journal. Retrieved 28 September 2011.
- [40] K. Kuramoto, W. Ohyama, T. Wakabayashi, F. Kimura, "Accuracy Improvement of Viewpoint-Free Scene Character Recognition by Rotation Angle Estimation", Camera-Based Document Analysis and Recognition, 60-70, 2014
- [41] S. Matsui, K. Endo, W. Ohyama, T. Wakabayashi, F. Kimura, "Dot-Matrix Character Recognition and Printing Defect Detection using Variation Model Based Learning", IEICE-114, no.454(PRMU), no.455(CNR), pp.89-90, Feb 2015
- [42] J. Shi and C. Tomasi, "Good Features to Track", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.593-600, June 1994
- [43] C. Harris and M. Stephens, "A Combined Corner and Edge Detector", Proceedings of

- the 4th Alvey Vision Conference, pp. 147-151, August 1988
- [44] B.S.Everitt “Cluster Analysis”, Edward Arnold third edition, 1993
- [45] D.Yimei, M.Okada, F.Kimura, Y.Miyake, “Application of slant correction to handwritten Japanese address recognition”, Document Analysis and Recognition, pp.670-674, 2001
- [46] J. Kittler and J. Illingworth, “Minimim error thresholding” Pattern Recognition, vol.19, pp.41-47, 1986
- [47] J.Sauvola, M. Pietikinen, “ Adaptive document image binarization”, Pattern Recognition, vol.33, pp.225-236, 2000
- [48] 藤本克仁, 鎌田洋, “低解像度カラー文書画像から高品質な文字画像を抽出する二値化方式” 電子情報通信学会技術研究報告, PRMU, パターン認識・メディア理解 99(380), 7-14, 1999-10-21
- [49] 田中 宏, 藤井 勇作, 堀田 悦伸, “二値化閾値の補正による低解像度画像に頑強な文字抽出用二値化” 電子情報通信学会技術研究報告, PRMU, パターン認識・メディア理解 110(467), 99-104, 2011-03-03
- [50] 後藤 英昭, 平山 理継, 阿曾 弘具 “局所多値しきい値処理による濃淡文書画像からの文字パターンの抽出”, 電子情報通信学会論文誌. D-2, 情報・システム, パターン処理 J82-D-2(11), 2188-2192, 1999-11-25
- [51] T Ojala, M Pietikainen, D Harwood, “A comparative study of texture measures with classification based on feature distributions” Pattern Recognition 29 (1), vol.29, pp.51-59, 1996