シーンテキスト位置の高速検出手法の提案
－日本語と英語を対象として－

馬屋原 昂[†1]　篠原 正太[†2]　山名 早人[†3†4]

†1 早稲田大学基幹理工学部 〒169-8555 東京都新宿区大久保 3-4-1
†2 早稲田大学基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1
†3 早稲田大学理工学術院〒169-8555 東京都新宿区大久保 3-4-1
†4 国立情報学研究科〒101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: †{silversoul, shinohara, yamana}@yama.info.waseda.ac.jp

あらまし 近年，スマートフォンの普及により，いつでもカメラで撮影できる環境にあり，撮影した画像からテキストなどの情報を得ることは有用である．特に，看板やポスターなどを撮影した情景画像内に含まれるテキストをシーンテキストと呼び，他言語のテキストが一緒に撮影される場合がある．シーンテキストの位置検出では，周囲の光や撮影時の角度による陰影，ノイズ，歪みなどの影響が検出精度に大きな影響を与える．従来のシーンテキスト位置の検出手法では，大量の文字候補領域を検出し，その候補領域の中で文字領域と判断された領域のみを結合してテキスト領域を検出することから計算量が膨大となっている．本稿では，PC に比較して処理能力の低いスマートフォンでのシーンテキスト位置の検出を可能とすることを目指し，多段階クラスタリングによる文字候補領域の結合手法を提案する．文字候補領域の中心座標に着目したクラスタリングにより非文字領域を削除し，特徴量の計算コストを抑え，クラスタリングを行う範囲を一定の範囲に限定することで高速化を図る．評価実験では，英語のみのシーンテキストのデータセットである ICDAR2011 と，日本語と英語が含まれるシーンテキストを用いた．評価実験の結果，提案手法は既存手法と比べ F 値を ICDAR データセットでは 0.028，著者が作成した JEST データセットでは 0.202 向上させ，実行時間はそれぞれ約 6.9，10.1 倍の高速化に成功した．

キーワード シーンテキスト，位置検出

1. はじめに

現在，スマートフォンの普及により人々は手軽に画像の撮影をすることができる．日常の風景を撮影した画像にはテキストが含まれることがあり，これらのテキストはシーンテキストと呼ばれる．例えば，街中の店名の看板や壁に貼られているポスター，本のタイトル，商品のラベルなどが挙げられる．また，日本においては，日本語以外にも英語のテキストが含まれることが多い．また，シーンテキストを読み取ることにより，アプリケーションへの応用として，道案内やテキストの翻訳，未知のキーワードの即時検索などが挙げられる．さらに，ドローンに代表される自律移動型ロボット制御への応用もできると考えられている．

シーンテキストの位置検出方法はテクスチャベース[1]と文字領域ベース[2][3][4]の 2 つに分類できる．さらに，文字領域ベースの手法はエッジベース[4]，連結成分ベース[2]，ストロークベース[3]の 3 つの手法に分類できる．近年のシーンテキストの位置検出では文字領域ベースの手法に分類される連結成分ベースの手法が用いられる傾向にある．これは，テキストのサイズや方向，フォントなどに依存しにくいためである．しかし，1 文字単位での文字の領域検出では，候補領域が画像内の文字数と比較して過剰に検出される傾向にある．したがって，非テキスト領域を削除する際に，過剰に検出された領域の特徴量を効率良く計算する必要があり，計算量が膨大となり，精度と速度はトレードオフの関係にある．また，リアルタイムにシーンテキストの位置を検出する手法[2]は，文字が 1 つの連結成分で構成されていることを前提とし，英語に適した手法である．しかしながら，日本語の漢字はアルファベットとは異なり複数の部首によって構成される．

日本語を対象とした手法 [5][6]も存在するが，これらには前提条件や問題が存在する．例えば，看板の背景色情報を利用した手法[5]では，前提条件として，看板の背景色を既知としている．また，カラー情報および明度情報を利用して作成した 2 値画像からテキストの文字の連結性を利用する手法[6]では実用的な計算時間が得られなかったと報告されている．

上記の問題を解決するために，本稿では，文字領域に対して多段階クラスタリングを行うことで，日本語と英語を対象としたシーンテキストの位置を高速に検出する手法を提案する．まず，文字候補領域を検出する．このとき，文字候補領域は先に述べた通り文字の一部を構成する可能性があるため，1 段階目のクラスタリングによって，文字の一部を構成する領域をまとめることで文字候補領域を得る．次に，2 段階目のク

ラスタリングによって，複数の文字候補領域を結合することでテキスト領域を検出する．このとき，文字候補領域の中心座標に着目したクラスタリングにより，非文字領域を削除し，特徴量の計算コストを抑える．さらに，クラスタリングを行う範囲を一定の範囲に限定することで高速化を図る．

本稿の構成は以下の通りである．第 2 節にて関連研究,第 3 節にて提案手法に手法についての説明を行う．第 4 節にて評価実験および結果についての考察を行う．最後に，第 5 節にてまとめる．

2. 関連研究

本節では，本研究に関連する研究について述べる．シーンテキストの検出手法はテクスチャベースと文字領域ベースとその両方を組み合わせたハイブリッド手法の 3 種類に分類される．

2.1. テクスチャベースの手法

2.1.1. Gang ら[1]の手法

2011 年に Gang ら[1]は,テクスチャ特徴量として，Histogram of Oriented Gradients(HOG)，Mean of Gradients(MG), Local Binary Patterns を用いた手法を提案した．これら 3 つの特徴量を sliding window を用いて，入力画像の一部の矩形領域(window)に注目し，その領域の座標，サイズ，比率などを変化させながら，検出器にかけることでシーンテキストの検出を行う．

Gang らの手法の問題として，主に水平方向のテキストデータセットを用いて学習しているので，水平でないテキストの検出ができないと述べている.つまり，検出したいテキストと同じ傾きを持つテキストの訓練データを用意しなければ傾きに対して頑健な検出ができない．また，大量の window 毎に計算して評価する必要があり，高速な検出には不向きである．

2.2. 文字領域ベースの手法

近年のシーンテキストの検出手法では文字領域ベースの手法が用いられる傾向にある．この手法では 1 文字単位で文字領域を検出し，その検出した複数の文字領域を結合してテキスト位置を検出する．1 文字単位で文字領域を検出する手法はエッジベース，連結成分ベース，ストロークベースの 3 つの手法に分類できる．

2.2.1. エッジベースの手法

Epshtein ら[4]はエッジベースの手法として，2010 年に Stroke Width Transform(SWT)を提案した.SWT は Canny エッジ検出器を用いて入力画像の画素値を，その画素が含まれているストロークの幅の値に変換して出力する局所記述子である． SWT によって変換した隣接する画素値の差が閾値以下のとき，これらの隣接する画素を同一領域とすることで，変換処理を行った画像に対して連結成分ベースの手法を適用する．

Epshtein らの手法の問題として,文字のストローク幅はほぼ一定であるという特徴と同一の連結成分で構成される文字であることを前提としているため，日本語の明朝体や書道フォントなどでは前提条件を満たしていない．また，エッジを利用するため，背景色と文字色が類似している場合の検出は困難となる．

2.2.2. 連結成分ベースの手法

連結成分ベースの手法[2]では，文字の各画素は類似した値を持つことを前提としている．類似した領域の抽出では RGB 色空間，HSV 色空間，グレースケールの輝度などを用いる．また，連結成分ベースの手法はエッジベースの手法やストロークベースの手法と比べて，効率よく文字単位の領域を検出できる手法であるため，文字認識フェーズでの文字単位の分割が容易となる．

Neumann らの手法では,まず ERs を用いて文字候補領域を抽出する．ERs とは連結成分であり，その連結成分の外側の境界に隣接する画素は内側の画素よりも高い値または低い値を持つ．次に，文字候補領域の非文字領域を削除するために，第 1 段階のフィルタとして Real AdaBoost,第 2 段階のフィルタとして,SVM の RBF カーネルを用いる．最後に，得られた文字領域を連結することでテキスト領域を得る．

Neumann らの手法の問題として，”i”と”j”以外のアルファベットは 1 つの連結成分で構成されているため，1 つの文字は 1 つの連結成分で構成されていることを前提としている．したがって，日本語の漢字のように複数の連結成分から構成される文字に対して頑健な手法ではない．

2.2.3. ストロークベースの手法

Liu ら[3]は 2014 年にエッジや連結成分よりもストロークの方が文字を構成する基本要素であると考えられるとし，difference of Gaussian(Dog) filter を用いた手法を提案した．具体的には，異なるスケール毎に相関 2 乗信号幅 w を設定し,その DoG 応答を用いることで，文字のエッジではなくストロークを抽出する．また，文字領域を結合して,テキスト位置を検出する際には，対象とする言語の持つ固有のレイアウトを利用している．英語を対象としているので，4 本の罫線をもとにした 4 つのスタイルのカテゴリ(“a” style, “h” style, “y” style, “f” style)に分けられることを利用している．また，英語に加えて数字も対象である．

Liu らの手法の問題として，英語以外の言語を対象

とする場合にはアルファベット固有のレイアウトを文字領域の結合のときに使用できないことが挙げられる．また，日本語のように複雑なストロークを持つ言語にそのまま適用することは難しい．

2.3. ハイブリッド手法

テクスチャと文字領域ベースのハイブリッド手法では，それぞれの手法の利点を取り入れることで検出精度の向上を図る．Tonouchi ら[7]は 2014 年にテクスチャベースの手法として sliding window を，文字領域ベースの手法として連結成分ベースの手法を用いるハイブリッド手法を提案した．ハイブリッド手法では，sliding window ベースの手法と連結成分ベースの手法のそれぞれから文字領域を求めて，最後に検出した領域を統合する．検出した領域の統合では，連結成分ベースの手法の結果を優先する．これは連結成分ベースの手法の方が sliding window ベースの手法と比べて正確な座標を検出できているからである．

この手法の問題としては，sliding window ベースの手法と連結成分ベースの手法の 2 種類の手法を処理する必要があり，高速な検出には不向きである．また，他の手法と同様に文字が 1 つの連結成分で構成されていることが前提である．

3. 多段クラスタリングによる文字領域の結合手法の提案

3.1. 概要

本論文では，日本語と英語を対象とした高速なシーンテキストの検出を目指している．提案手法では，従来の研究にならい，文字領域ベースの手法を採用する．これは，テクスチャベースの手法と比較して高い精度を得ることができるからである．次に，2.2.2 で述べたように Neumann らの手法[2]が英語を対象としており，そのままでは日本語に対応できない問題を解決する．さらに，領域併合時の処理を効率的に行うことで，高速なシーンテキストの位置検出を実現する．

ここに，Neumann らの手法からの改良を簡単にまとめる．Neumann らは文字が基本的に単一の連結成分で構成されることを仮定している．したがって，文字候補領域に対して多段階クラスタリングを行うことで，日本語のように文字が複数の連結成分で構成される場合に対応させる．

多段階クラスタリングを用いたシステムの概要図を図 3.1 に示す．エッジ検出結果をラベリングすることで得られた文字領域を A とする．1 段階目のクラスタリングによって，単一の連結成分で構成される文字領域クラスタ B と文字の一部を構成する領域のクラスタ C に分ける．ここで，文字の一部を構成する領域のクラスタ C から文字候補領域 E が得られる．次に，先の単一の連結成分で構成される文字領域クラスタ B に属する文字候補領域を D とする．2 段階目のクラスタリングによって，文字候補領域 D，E から文字領域クラスタ F を得る．最後に，文字領域クラスタ F 内の同一行のテキストを単語毎に分割することでテキスト領域 G を得る．

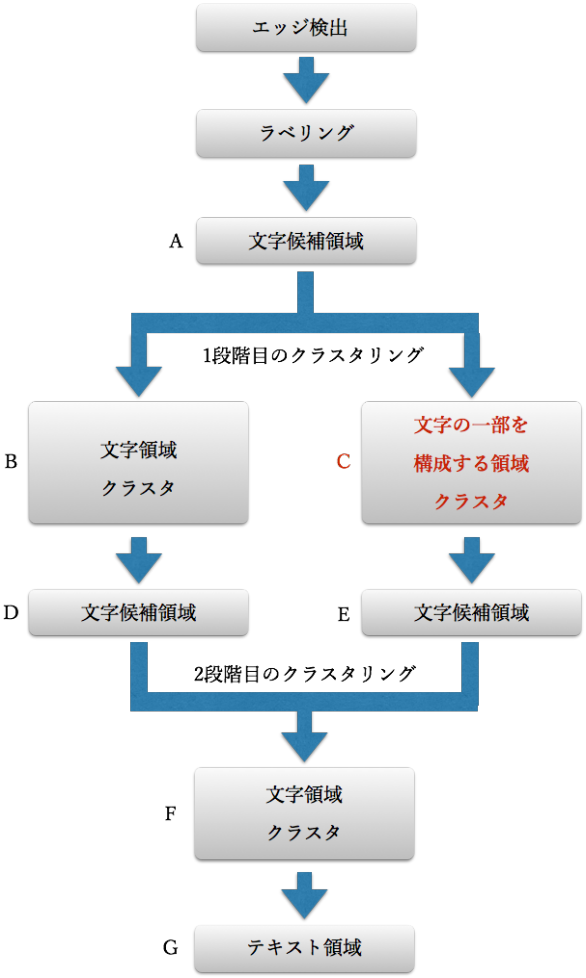

図 3.1 システム概要図

ここに，領域合併時の処理をどのように効率的に行うのかを簡単にまとめる．複数の文字候補領域の中心座標の配置に着目することで，複雑な特徴量の計算なしに文字領域から適切なクラスタを得る．多くの場合，隣り合うように配置されている文字のサイズが大きく異なることはないと考えられる．したがって，文字候補領域の座標とサイズからハッシュ値を生成し，同一のハッシュ値を持つ領域同士を比較して結合処理を行うことで効率よく領域の合併を行う．

最後に，本論文ではシーンテキストにおいて，以下のような仮定をおくこととする．

仮定(1) 水平方向に配置されている．

仮定(2) 一定の高さ以上である．

仮定(3) 幅と高さは一定の比率の範囲内である．

仮定(4) 隣りの文字とエッジを共有していない．

ここで，上記のような仮定をおいた場合も，十分実用的であることを説明する．多くの言語において，シーンテキストは水平方向に配置されることが多い．日本語のように垂直方向に配置されることもあるが，水平方向のシーンテキストの割合の方が多いため，仮定(1)をおいた場合でも十分に情報が得られると考えられる．仮定(2)，仮定(3)はシーンテキストの誤検出を抑える．シーンテキストを撮影するときには，対象とするテキストにフォーカスすることが想定され，提案手法の評価実験に用いる ICDAR のデータセットはフォーカスしたシーンテキストから構成されることから，仮定(2)は問題ないと考えられる．また，シーンテキストの多くは固有名詞や短い文章であり，長い文章である場合は少ない．したがって，検出精度を優先するために仮定(3)を設けた．

提案手法では，エッジベースの手法を用いて文字単位の領域を検出するため仮定(4)が必要となる．仮定(4)が成り立たない場合の例として，シーンテキストの文字が小さい場合は文字がつぶれることで，隣り合う文字とエッジが一体化してしまい，複数の文字を連結した領域を検出してしまう．このとき，小さいシーンテキストは仮定(2)によって検出の対象外となっているので，仮定(4)に反するシーンテキストの数はそれほど多くないと考えられる．

3.2. 文字領域の検出

文字領域の検出について述べる．まず，3.2.1 では文字領域を検出するための前処理を述べ，3.2.2 ではラベリング方法について述べる．

3.2.1. 文字領域の検出の前処理

文字領域の検出の前処理として，Epshtein ら[4]のエッジベースの手法である SWT とは異なり，近傍の画素値を用いた単純なエッジ検出を行う．画素(x,y)の近傍の画素値として横方向の走査では$(x-1,y)$，縦方向の走査では$(x,y-1)$を用いることで，横方向と縦方向のエッジをそれぞれ検出する．このとき，横と縦のそれぞれの方向に関して画素値を走査し，ある画素と前回の画素の値の差の絶対値が一定の閾値θ_x，θ_y以上である場合はエッジであるとする．横方向と縦方向のエッジの和集合によって得られた画像を次の処理に用いる．また，画素値の値として YCbCr 色空間の輝度を表す$Y \in [0,255]$を用いる．

3.2.2. 輪郭ラベリングによる文字領域の検出

検出したエッジの輪郭ラベリングにより文字候補領域を検出する．Neumann ら[2]の手法のように ERs を検出せずに，エッジを用いた単純な輪郭ラベリングによって文字候補領域を検出する．輪郭ラベリングでは，画素のつながりを 8 連結で考え，画素(x,y)に隣接している画素は$(x\pm1,y)$と$(x,y\pm1)$と$(x\pm1,y\pm1)$と$(x\pm1,y\mp1)$となる．

ラベリングによって割り当てられた番号が同一の画素の集合を考える．その画素の集合のx座標の最小値x_{min}，最大値x_{max}，y座標の最小値y_{min}，最大値y_{max}とする．このときに，矩形$(x_{min}, y_{min}, x_{max}, y_{max})$を文字候補領域とする．

3.3. 文字領域のクラスタリング手法

3.3.1. 1 段階目のクラスタリング

1 段階目のクラスタリングでは，図 3.2 に示したように単一の連結成分で構成される文字領域クラスタ B と文字の一部を構成する領域のクラスタ C に分ける．まず，単一の連結成分で構成される文字領域クラスタ B を求める．その後に，そのクラスタ B に分類されなかった領域に関して，クラスタリングを行い，文字の一部を構成する領域のクラスタ C とそれ以外の領域に分ける．このとき，実線で表された矩形が文字候補領域である．

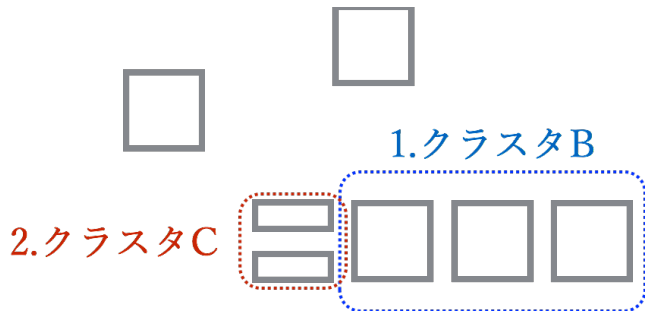

図 3.2 1 段階目のクラスタリングの例

ここで，単一の連結成分で構成される文字領域クラスタ B のクラスタリング手法について述べる．文字候補領域の集合を A とし，その集合の要素として$r_i, r_j \in A\ (i \neq j)$を考える．その領域から計算されるハッシュ値を$hash(r_i)$，$hash(r_j)$で表す．ハッシュ値の計算方法は 3.3.3 で述べる．また，r_i，r_jに対応する文字の色をC_i，C_jとする．文字の色を RGB 色空間で考え，R,G,B∈[0.0,1.0]とする．このとき，2 つの色C_i，C_jの距離を式(3.1)により算出する．

$$dist(C_i, C_j) = \sqrt{\frac{(R_j - R_i)^2 + (G_j - G_i)^2 + (B_j - B_i)^2}{3}} \quad (3.1)$$

領域r_iは矩形であるため，$(x_{left}, y_{top}, x_{right}, y_{bottom})$と表現でき，中心の$y$座標$y_{center}$=$(y_{top} + y_{bottom})/2$となる．領域$r_i$と$r_j$において，それぞれの$y$座標の差の絶対値は$|y_{top_j} - y_{top_i}|$，$|y_{center_j} - y_{center_i}|$，$|y_{bottom_j} - y_{bottom_i}|$となる．そして，その絶対値のいずれかが 2 つの領域の高さの平均に比例する値$y_{const}\frac{y_1+y_2}{2}$以下ならば，$f(y_{const}, y_1, y_2)$を満たすとする．このとき，$y_{const}$

はパラメータである．

また，文字は水平方向に配置されていることを仮定している．$w = x_{right} - x_{left}$として，領域r_iを拡張した領域を$gr_i = (x_{left} - w,\ y_{top}, x_{right} + w, y_{bottom})$とする．このとき，領域$r_i$，$r_j$が重なりあうときは$|gr_i \cap gr_j| > 0$を満たす．

そして，次の条件(3.2)を満たすr_i，r_jは同一のクラスタに属する．また，θ_{color}は閾値のパラメータである．

$$\{r_i, r_j \mid dist(C_i, C_j) \le \theta_{color} \wedge hash(r_i) = hash(r_j) \wedge |gr_i \cap gr_j| > 0 \wedge f(y_{const}, y_i, y_j)\} \quad (3.2)$$

次に，文字の一部を構成する領域のクラスタ C のクラスタリング手法について述べる．文字の一部を構成する 2 つの領域の横・縦のサイズ比$ratio_x$，$ratio_y$は両者ともに$\frac{1}{2} \le ratio_x, ratio_y \le 2$であると仮定し，$r_i$，$r_j$に対応する横・縦のサイズ比を$ratio_{x_ij}$，$ratio_{y_ij}$とする．条件(3.2)によってクラスタリングされなかった領域r_i，r_jに関して，次の条件(3.3)を満たすならば同一のクラスタとする．

$$\{r_i, r_j \mid dist(C_i, C_j) \le \theta_{color} \wedge hash(r_i) = hash(r_j) \wedge |gr_i \cap gr_j| > 0 \wedge \frac{1}{2} \le ratio_{x\_ij}, ratio_{y\_ij} \le 2\} \quad (3.3)$$

2 段階目のクラスタリングを行う前処理として，図 3.3 に示したように，クラスタ B に属する領域を文字候補領域 D とし，クラスタ C の領域をそのまま文字候補領域 E とする．このとき，点線で表された矩形が削除された文字候補領域である．

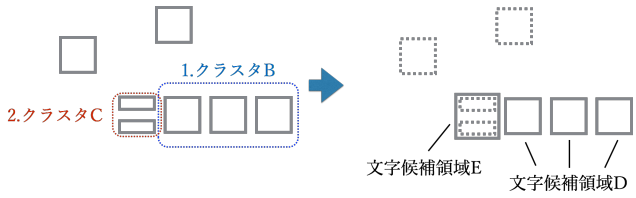

図 3.3 2 段階目のクラスタリングの前処理の例

3.3.2. 2 段階目のクラスタリング

2 段階目のクラスタリングでは，図 3.4 に示したように文字候補領域 D，E に対して，1 段階目のクラスタリングで単一の連結成分で構成される文字領域クラスタ B を用いた手法をそのまま用いることで文字領域クラスタ F を得る．

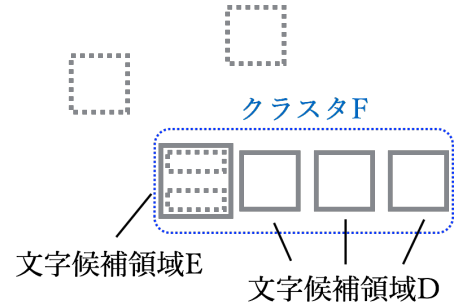

図 3.4 2 段階目のクラスタリングの例

2 段階目のクラスタリングの最後の処理として，図 3.5 に示したように文字領域クラスタ F 内の同一行のテキストを単語毎に分割することでテキスト領域 G を得る．文字領域の間隔x_1，x_2，…，x_nの平均値を$\bar{x}$，分散をσとし，$x_i \ge x + 2\sigma$を満たす場合に分割を行う．

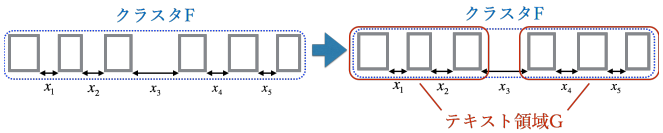

図 3.5 同一行のテキストを単語毎に分割する例

また，提案手法で検出するシーンテキストの仮定をもとにした条件(3.4)，条件(3.5)，条件(3.6)により，テキスト領域 G にフィルタをかける．

$$n_{min} \le |G_i| \quad (3.4)$$

$$y_{min} \le h \quad (3.5)$$

$$ratio_{min} \le \frac{w}{h} \le ratio_{max} \quad (3.6)$$

このとき，i 番目のテキスト領域をG_i，シーンテキストの幅をw，高さhとする．また，n_{min}はテキスト領域に含まれる文字領域数の最小値，y_{min}はシーンテキストの高さの最小値，$ratio_{min}$，$ratio_{max}$はそれぞれ幅と高さの比の最小値と最大値を表すパラメータである．

3.3.3. ハッシュ値を用いたクラスタリング

クラスタリングでは，ハッシュ値を用いることで効率よく文字候補領域のクラスタリングを行う．それぞれの文字候補領域の中心座標(x, y)と文字の高さhの値によって構築したハッシュテーブルを用いて，効率よく総当りの処理を行う．

入力画像の大きさ(w_{image}, h_{image})において，横と縦をn等分した領域を考える．図 3.6 に$n = 5$のときの例を示す．黒い太線で表された文字候補領域は実線で表された領域に属することになる．つまり，(1,1)，(2,1)，(1,2)，(2,2)に属する．

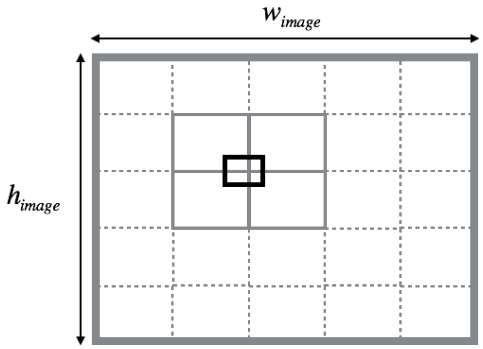

図 3.6 入力画像の領域をn等分した例($n=5$)

さらに，文字候補領域の高さhを図 3.7 のように$2^i \le h \le 2^{i+1}(0 \le i \le m)$と$2^i + 2^{i-1} \le h \le 2^{i+1} + 2^i(0 \le i \le m-1)$のそれぞれ該当する範囲に分ける．このとき，$m$は$2^m < h_{image}$を満たす最大の整数である．このとき，高さ$h$の範囲が重複するように設定することで境界値付近の値を持つ領域が異なるハッシュ値を持つことを防ぐ．

図 3.7 文字候補領域の高さhの範囲

4. 評価実験

4.1. データセット

4.1.1. 英語のデータセット

英語のデータセットとして，ICDAR2013 にて用いられたデータセットを用いる．このデータセットは ICDAR 2011[1]にて用いられたデータセットのサブセットである．訓練データは 229 件，テストデータは 233 件である．データセットの例を図 4.1 に示す．

4.1.2. 日本語と英語のデータセット

現在，一般に公開され，実験の評価に用いられている日本語のデータセットは存在しないため，著者が Google 画像検索を用いて全部で 151 件の画像を収集し，訓練データは 77 件，テストデータは 74 件とした．このデータセットを Japanese and English Scene Text とし，以降は JEST と表記する．また，検索キーワードとして「看板」を用いた．ICDAR のデータセットと同様に，テキストの正解データは左上と右下の座標によって表される．正解データは著者が手作業で付与し，極端に小さいテキストであり文字認識が困難であるような場合は正解データとしていない．さらに，比較手法は水平方向のテキストを対象としているため，条件を揃えるために，水平方向のテキストを使用した．データセットの例を図 4.2 に示す．

[1] http://robustreading.opendfki.de/wiki/SceneText

図 4.1 Example of ICDAR2013 Robust Reading competition dataset[8]

図 4.2 Example of JEST dataset
(Google の画像検索より)

4.1.3. 比較手法で用いる文字単位の ERs の訓練データセット

英語を対象とした実験のための文字単位の ERs の訓練データセットとして，0~9 の数字と英語の小文字と大文字 52 種類の計 62 種類を文字 ERs として 2328 個，非文字 ERs として 1,686 個を用いる．データセットの例をそれぞれ，図 4.3，図 4.4 に示した．また，日本語の文字の種類が多いため Neumann ら[2]と同様にフォントデータを用いて全種類のひらがなとカタカナと常用漢字の全種類を網羅する．今回はフォントデータとしてゴシック体および明朝体のフォントデータを用いる．4.1.2 で作成した訓練データセット内から抽出した ERs と合わせて計 7,308 枚の画像を訓練データセットとする．また，英語のデータセットと共通の非文字 ERs を用いる．

図 4.3 文字 ERs のデータセット

図 4.4 非文字 ERs のデータセット

4.2. 比較手法

提案手法との比較手法として，Neumann ら[2]によって提案されたリアルタイムなシーンテキスト検出の手

法を用いる．Neumann ら[2]の論文では，文字単位の ERs の訓練データとして，ICDAR2003 training dataset[9]から手作業で文字 ERs を約 900 個，非文字 ERs を約 1400 個用いている．しかし，文字単位の ERs は手作業で作成したデータであり，公開されていない．したがって，完全に条件を一致させることはできないが，4.1.3 にて述べた文字単位の ERs の訓練データを用いて実験を行う．

4.3. 評価方法

本研究の評価には，ICDAR 2013 competition で使用されているソフトウェアの DetEval[2]を用いる．DetEval は Wolf ら[10]が提案した評価方法(one-to-one, one-to-many, many-to-many)を元に作成されたソフトウェアである．recall(再現率)，precision(適合率)，F-measure(F 値)はそれぞれ式(4.1)(4.2)(4.3)によって求めることができる．

$$recall(G,D,t_r,t_p) = \frac{\sum_i Match_G(G_i,D,t_r,t_p)}{|G|} \quad (4.1)$$

$$pricision(G,D,t_r,t_p) = \frac{\sum_j Match_D(D_j,G,t_r,t_p)}{|D|} \quad (4.2)$$

$$F-measure = 2\frac{pricision(G,D,t_r,t_p)\cdot recall(G,D,t_r,t_p)}{recall(G,D,t_r,t_p)+pricision(G,D,t_r,t_p)} \quad (4.3)$$

ここで，G と D はそれぞれ正解データの矩形，検出した矩形の集合である．$t_r \in [0,1]$と$t_p \in [0,1]$は recall と precision を決定する際の領域の面積を制限する定数である．$Match_D$と$Match_G$は one-to-one，one-to-many，many-to-many matches において異なる値を返す関数である．one-to-one は正解データの矩形 1 個に対して，検出した矩形が 1 個の場合，one-to-many は正解データの矩形 1 個に対して，検出した矩形が複数個の場合，many-to-many は正解データの複数個の矩形に対して，検出した矩形も複数個の場合である．また，パラメータとは DetEval のデフォルト値である 0.8，0.4 をそれぞれ用いた．

4.4. 結果と考察

4.4.1. パラメータ

提案手法におけるパラメータについて説明する．訓練データを用いた予備実験により，パラメータを次のように設定した．まず，エッジ検出をする際の閾値のパラメータは$\theta_x = 12$，$\theta_y = 12$とする．次に，クラスタリングを行う際のパラメータとして，条件(3.3)のパラメータ$y_{const} = 0.1$，$\theta_{color} = 0.1$とする．また，提案手法の仮定にもとづいた条件 (3.4)，条件 (3.5)，条件 (3.6) のパラメータは$n_{min} = 3$，$y_{min} = 16$，$ratio_{min} = 2.0$，$ratio_{max} = 20.0$とする．3.3.3 のハッシュ値を用いたクラスタリングでは，$n = 5$とする．

4.4.2. 既存手法との比較

既存手法と検出速度を比較するための条件を以下に示す．今回の実験に使用した計算機は 2 コア，1.7GHz，8G RAM，Mac OS X である．既存研究の計測では並列処理を行っていないため，本手法の評価実験においても並列処理は行わない．また，800x600 の画像に関して平均処理時間を計測しているので，データセットの 4:3 の比率の画像のスケールを調整して 800x600 の画像を作成した．実際に計測する検出時間は画像の読み込みが完了してからシーンテキスト位置の検出が完了するまでとし，キャッシュの影響を考慮して 3 回の実行結果の平均をとった．データセットでは 152 枚，JEST データセットでは 74 枚の画像を用いて実験を行い，既存手法の結果を表 4.1，提案手法の結果を表 4.2 に示した．表 4.1 の Neumann らの手法において，訓練データセットが英語，テストデータセットが ICDAR, クロック周波数が 3.4GHz のときの実験結果は Liu ら[3]の TABLE III の数値を参照した．このとき，Liu らは 3.4GHz の標準コンピュータを用いたと述べている．

Neumann らの手法において，日本語と英語を訓練データとした ICDAR テストデータセットの実験では，特に precision が低下した．英語と比べて日本語は複雑な形状の文字が多く存在するため，特徴量が有効に機能していないと考えられる．同様に日本語と英語を訓練データとした JEST テストデータセットの実験においても precision が低い値を示した．検出時間が増加した原因としては，precision がさらに低い値となっていることから，文字領域を削除するフィルタであると考えられる．つまり，第 1 段階目のフィルタである Real AdaBoost による文字領域の削除が有効に機能しなくなり，第 2 段階目のフィルタである SVM の特徴量の計算の処理時間が加わったためである．

提案手法において，日本語と英語を訓練データとした ICDAR テストデータセットの実験では，Neumann らの結果と比較して recall は低いが，precision が高いため，F 値が向上している．recall が低い値となっている理由として，提案手法では 1 つの文字が 1 つの連結成分から構成されることを前提としないので，英語よりも日本語に適した手法であるからと考えられる．また，ICDAR データセットは JEST データセットと

[2] http://liris.cnrs.fr/christian.wolf/software/deteval/

比べて，同一行のテキストを単語毎に適切に区切る必要があり，F 値が低下している．日本語と英語を訓練データとした JEST テストデータセットの実験において Neumann らの結果と比較して precision が高くなり，F 値が向上している．また，既存手法と比べて約 7~10 倍の高速化に成功した．

Neumann らの手法では文字候補領域のフィルタが機能しないため，検出時間が増大し，精度も低下した．提案手法では，ERs ではなく，エッジをもとにラベリングをして文字候補領域を検出したことと，計算コストが高いフィルタを用いることなく，文字候補領域の配置にもとづいてクラスタリングをすることで高速化に貢献したと考えられる．

表 4.1 Neumann らの手法の結果

データセット		recall	precision	F 値	検出時間(ms)		備考
訓練	テスト				1.7 GHz	3.4 GHz	
英語	ICDAR	0.647	0.731	0.687	-	589.9	[3] より引用
日本語/英語	ICDAR	0.512	0.263	0.347	1333.5	-	著者による実装
日本語/英語	JEST	0.564	0.208	0.304	1858.7	-	

表 4.2 提案手法の結果

データセット		recall	precision	F 値	検出時間(ms)	
訓練	テスト				1.7 GHz	3.4 GHz
日本語/英語	ICDAR	0.346	0.409	0.375	192.0	-
日本語/英語	JEST	0.496	0.516	0.506	184.8	-

5. まとめ

本稿では，日本語と英語を対象としたシーンテキスト位置の高速検出手法を提案した．複数の連結成分から構成される文字を考慮した手法として多段階クラスタリングによる文字候補領域の結合を提案し，精度を保ちつつ，速度の向上を図った．その結果，提案手法は既存手法と比べ F 値を ICDAR データセットでは 0.028，著者が作成した JEST データセットでは 0.202 向上させ，実行時間はそれぞれ約 6.9，10.1 倍の高速化に成功した．また，処理速度を保ちつつ，より高い精度を得るために，多くの訓練データを用いることや，計算量の少ない有用な特徴量を用いる手法の考案が今後の課題となる．

参 考 文 献

[1] Gang Ahou, Yuehu Liu, Quan Meng and Yuanlin Zhang: "Detecting multilingual text in natural scene", Proceedings of IEEE 1st International Symposium on Access Spaces (ISAS), pp.116-120, 2011.

[2] Neumann Lukáš and Jiří Matas: "Real-time scene text localization and recognition", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3538-3545, 2012.

[3] Liu Yi, Dongming Zhang, Yongdong Zhang and Shouxun Lin: "Real-Time Scene Text Detection Based on Stroke Model", Proceedings of IEEE 22nd International Conference on Pattern Recognition (ICPR), pp.3116-3120, 2014.

[4] Epshtein Boris, Eyal Ofek and Yonatan Wexler: "Detecting text in natural scenes with stroke width transform", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2963-2970, 2010.

[5] 野村 松信, 鈴木 拓真, 景山 陽一, 石沢 千佳子, 西田 眞: "背景色情報を活用した看板内の文字列領域抽出法", 電気学会論文誌 C (電子・情報・システム部門誌), Vol.134, No.1, pp.121-130, 2014.

[6] 平山勝裕, 大町真一郎, 阿曽弘具: "カラー情報を利用した情景画像中の文字列の高精度抽出", 電子情報通信学会信学技報, Vol.104, No.742, pp.91-96, 2005.

[7] Tonouchi Yojiro, Kaoru Suzuki and Kunio Osada: "A Hybrid Approach to Detect Texts in Natural Scenes by Integration of a Connected-Component Method and a Sliding-Window Method", Computer Vision-ACCV 2014 Workshops, Springer International Publishing, pp.106-118, 2014.

[8] Karatzas D., Shafait F., Uchida S., Iwamura M., Gomez i Bigorda L., Robles Mestre S., Mas J., Fernandez Mota D., Almazan Almazan J. and de las Heras L.-P.: "ICDAR 2013 robust reading competition", Proceedings of International Conference on Document Analysis and Recognition (ICDAR), pp.1484-1493, 2013.

[9] Lucas S. M., Panaretos A., Sosa L., Tang A., Wong S. and Young R.: "ICDAR 2003 robust reading competitions", Proceedings of ICDAR 2003 robust reading competitions, pp.682-687, 2003.

[10] Wolf Christian and Jean-Michel Jolion: "Object count/area graphs for the evaluation of object detection and segmentation algorithms", Proceedings of International Journal of Document Analysis and Recognition (IJDAR), Vol.8, Issue.4, pp.280-296, 2006.