

修士学位論文

カラー情報を利用した情景画像中の文字列の
高精度抽出に関する研究

東北大学大学院 工学研究科 電気・通信工学専攻

阿曾研究室 博士前期課程2年

平山 勝裕

目次

第 1 章	序論	1
1.1	研究の背景	1
1.2	研究の目的	2
1.3	本論文の構成	3
第 2 章	情景画像からの文字列領域の抽出	4
第 3 章	前処理 (平滑化)	6
第 4 章	2 値画像作成	9
4.1	カラー情報を利用した 2 値画像作成	9
4.1.1	緒言	9
4.1.2	カラークラスタリング	10
4.1.3	オブジェクトクラスタリング	12
4.1.4	複雑度による 2 値画像評価	14
4.2	明度情報を利用した 2 値画像作成	15
4.2.1	緒言	15
4.2.2	画素閾値決定	16
4.2.3	画像補正	20
第 5 章	文字列領域の抽出	22
5.1	緒言	22
5.2	文字列領域候補抽出	22
5.2.1	文字領域候補抽出	22
5.2.2	マーキング処理	23
5.2.3	文字列領域候補及び漢数字候補抽出	25
5.3	文字列領域抽出	26
第 6 章	実験	28
6.1	実験準備	28
6.1.1	評価基準	28
6.1.2	従来手法	30
6.2	実験結果	30

6.3 実験まとめ	38
第 7 章 結論及び今後の課題	40
7.1 結論	40
7.2 今後の課題	42
謝辞	44
参考文献	45

目 次

2.1	処理手順の流れ	5
3.1	WMFAD	6
3.2	WMFAD によるエッジ部分を残したまま平滑化する例	7
3.3	WMFAD の平滑化によるノイズ除去の例	7
3.4	WMFAD	8
4.1	カラークラスタリング	11
4.2	カラークラスタリングによって作成されたクラスタリング画像及び文字領域の拡大画像	12
4.3	正規化 RGB 空間	13
4.4	オブジェクトクラスタリング	13
4.5	参照領域	16
4.6	オブジェクトを良好に分解できない例	16
4.7	準備	17
4.8	閾値を変化させた場合の 2 値画像	18
4.9	NAT 法による 2 値画像作成	18
4.10	画素閾値決定法における平均色	19
4.11	オブジェクトを余計に連結してしまう画素	20
4.12	画像補正	20
4.13	画像補正結果例	21
5.1	4 連結による領域分割及び外接方形の形状による分類	23
5.2	各マーキング処理の例	25
5.3	垂直方向ヒストグラムからの文字列範囲の決定	27
5.4	文字列領域決定	27
6.1	文字列統合画像における再現率、適合率の計算例	29
6.2	文字列領域抽出画像における評価例	30
6.3	文字列を成す領域が照明の影響により不安定な画像	31
6.4	文字列を成す領域が照明の影響により似たような明度値を保持している画像	32
6.5	文字列を成す領域が画像内において小さい画像	33
6.6	文字列を成す領域が画像内において小さい画像の文字列領域抽出	34

6.7	文字列を成す領域が画像内において小さい画像	35
6.8	文字列を成す領域が照明の影響により文字の上部と下部とで特徴の異なる画像 . . .	36
6.9	文字列を成す領域が照明の影響により文字の上部と下部とで特徴の異なる画像の文字列領域抽出	36
6.10	曲線的に文字が配置されている	37
6.11	四角形の領域が並んでいる画像	38

表 目 次

6.1 実験結果 (文字列統合画像)	30
6.2 実験結果 (文字列領域抽出画像)	31

第1章

序論

1.1 研究の背景

現代社会で生活していく中で文字情報はあらゆる場所に存在し、様々な情報を我々に提供してくれる。例えば、店舗の名前や住所、地名などの書かれた看板上に存在する文字、また経路情報や交通状況などの道路交通情報の書かれた情報板上に存在する文字などがある。これら現代社会に溢れている文字情報を何らかの移動機能とともに視覚機能を保持するシステムが正確に取得し、理解することができれば様々なことに応用できると考えられる。

例えば、視覚機能を保持した自律移動ロボットは看板や情報板上の文字情報から位置情報や作業内容を取得することで、目的地までの移動をより確実に行うことや、目的としている作業をより正確に行うことができる。またデジタルビデオカメラを搭載した自動車は、看板や情報板上の文字情報から位置情報や経路情報、また交通状況などを取得することで、自動車の自動運転支援や運転手に対する事故や渋滞、交通状況などの情報提供を行うことができる。

しかし、デジタルカメラなどで撮影された画像から文字情報を取得しようとする場合、画像中には文字情報以外にも様々なオブジェクト(空、建物、車、人など)が存在し、また撮影時の天候や時間、撮影角度などによって文字情報に陰影が掛かってしまう場合などがあり、文字情報のみを正確に取得することを困難にさせる要因が多々存在する。

そのため、画像中から文字情報を正確に取得するためには、まず文字情報が存在している領域を全て正しく抽出し、画像中のどこに存在しているかを理解する必要がある。そこで、デジタルカメラを用いて様々な状況下における文字情報が存在する情景を撮影した画像(以下情景画像)から文字列の領域を正確に抽出するための研究が行われている。

1.2 研究の目的

本研究の目的は、種々の状況下における情景画像中から文字列領域を高精度に抽出する手法を開発することである。

情景画像に存在する文字領域の抽出に関する従来手法は大きく画像中の文字領域の書式が知識として与えられている手法と、そうでない手法の2つに分けることができる。

前者の手法として、抽出対象を円形道路標識に限定し、情景画像内から道路標識に含まれる色を判別することで円形領域を抽出し、抽出された領域において特定色の面積比率を利用することで正確に円形道路標識を抽出する手法 [1] がある。

また後者の手法の中で明度情報を利用して文字列領域を抽出する手法として、文字線とその背景の濃度コントラストは高く、文字線の太さや濃度は均一であるなどの特徴を利用して文字線領域を抽出し、文字の普遍的な性質を利用して文字領域を抽出する手法 [2] や明度値で表した明度情景画像に対し、閾値を変化させながら得られる2値画像を平均隣接数によって評価し、文字と背景とを良好に分解しているであろう2値画像を複数枚選択し、選択された2値画像から文字列を成す文字同士の連結性を利用して文字列領域を抽出する手法 [3] などがある。またカラー情報を利用して文字列領域を抽出する手法としてHSV変換によって色相・彩度・明度を求め、それを利用することで複数枚の2値画像を作成し、作成された2値画像から文字列を成す文字同士の連結性を利用して文字列領域を抽出する手法 [4] や正規化RGB空間上でのrとgのヒストグラムから画像内に存在しているオブジェクトの分割位置を決定することで複数枚の2値画像を作成し、作成した2値画像から文字列は一直線に並んでいるという特徴を利用したAMA(Aligning-and-Merging-Analysis)法を利用して文字列領域の抽出を行う手法 [5] などがある。

しかし明度情報を利用して文字列領域を抽出する手法は、明度情報の特徴を利用して情景画像中から文字列領域の抽出を行うため、照明の影響などにより文字列部分とその背景部分との明度値が似ていたり、文字列の文字の上部と下部とで明度値が異なっているような画像においては良好に文字列領域を抽出することが困難となる場合がある。またカラー情報を利用して文字列領域を抽出する手法は、カラー情報の特徴を利用して情景画像中から文字列領域の抽出を行うため、情景画像を撮影した際に照明が少なく、画像全体が暗くなりグレースケールに近くなりカラー情報が少なくなってしまう画像においては良好に文字列領域を抽出することが困難となる場合がある。

また画像全体を利用して明度やカラーの特徴を利用して文字列領域を抽出する手法の場合、画像中に存在する文字列が小さい場合などに文字列部分の特徴が現れにくく、良好に文字列領域を抽出することが困難となる場合がある。

そこで本研究では、まずカラー情報と明度情報の両者を利用し、文字とそれ以外の部分とを良好に分解した2値画像を複数枚作成する。これにより明度情報のみやカラー情報のみで2値画像を作成したために文字とそれ以外の部分とを良好に分解できなかった画像においても良好に分解した2値画像を得ることができる。そして作成された2値画像から文字列を成す文字の連結性を利用して文字の領域を抽出し、その抽出された領域を利用して最終的に文字列の領域を決定する手法を提案する。

1.3 本論文の構成

本論文の構成は以下のとおりである。

第1章 研究の背景及び目的について述べた。

第2章 本研究で提案する情景画像中からの文字列抽出手法の処理手順及び概要について述べる。

第3章 前処理として行う平滑化について述べる。

第4章 前処理を施した情景画像に対し、カラー情報と明度情報とをそれぞれ利用し、文字とそれ以外の部分とを良好に分解する2値画像を作成する手法について述べる。

第5章 作成された全ての2値画像から文字列領域候補を抽出し、各2値画像で抽出された文字列領域候補から文字列領域を抽出する手法について述べる。

第6章 実験条件、実験結果及び考察について述べる。

第7章 結論と今後の課題について述べる。

第2章

情景画像からの文字列領域の抽出

情景画像中に存在する文字列は、その文字列自体の種類や特徴、意味などによって様々な形や色で存在している。また撮影時の天候や時間、文字列との距離や角度によって様々な大きさや明度で存在する。これらの問題に対し、文字の形や色などの制限を設けず、情景画像中において視認ができる大きさの文字列の領域を高精度に抽出するために、以下の手順に従って文字列領域を抽出する。また処理手順の流れを表すフローチャートを図 2.1 に示す。

(1) 前処理 (平滑化)

情景画像は複数のオブジェクトで構成されている。これら複数存在するオブジェクトの中で、同じオブジェクトを表している画素が似たような RGB 値を保持するような画像を作成するために情景画像を平滑化する。

以上の内容を 3 章で述べる。

(2) 2 値画像作成

前処理を施した平滑化画像に対してカラー情報と明度情報とをそれぞれ利用して、文字とそれ以外の部分とを良好に分解した 2 値画像を作成する。

- カラー情報を利用した 2 値画像作成

情景画像に対し $L^*a^*b^*$ 空間上でカラークラスタリングを行い、クラスタリング画像を作成する。次にクラスタリング画像の正規化 RGB 空間上での特徴からオブジェクトの分割位置を複数個決定し、2 値化を行うことで複数枚の 2 値画像を作成する。

- 明度情報を利用した 2 値画像作成

情景画像の RGB 値から明度値を計算し明度情景画像を作成する。この明度情景画像を小領域に分割し、各小領域において 4 つの参照領域から 4 個の閾値を決定する。次に各小領域に所属する各画素において、ある判断基準を基に 4 個の閾値に重みを付加することで各画素の閾値を決定し、1 枚の 2 値画像を作成する。

以上の内容を4章で述べる。

(3) 文字列領域抽出

作成された各2値画像に対して文字を表していると考えられる領域を文字領域候補として抽出する。そして文字領域候補の中から文字列を構成する文字を表していると考えられる領域を文字列領域候補として抽出する。この文字列領域候補を利用して最終的に文字列を表している領域を抽出する。

● 文字列領域候補抽出

各2値画像から閉領域を抽出し、その外接方形枠を求める。この外接方形の形状によって各閉領域を文字領域候補、部分領域候補、漢数字候補、不用の4種類に分類する。文字領域候補と部分領域候補は文字の一部の可能性があるともみなし、マーキング処理を施す。このようにして抽出された文字領域候補の中から文字列を成す文字の連結性を利用して文字列領域候補を抽出する。

● 文字列領域抽出

各2値画像で抽出された文字列領域候補を1つの画像に全て重ね合わせることで文字列統合画像を作成する。この文字列統合画像において垂直及び水平方向に対し文字列領域候補の重なりを示すヒストグラムを作成し、最大値を含む山を文字列の存在する範囲とし、文字列領域として抽出する。

以上の内容を5章で述べる。

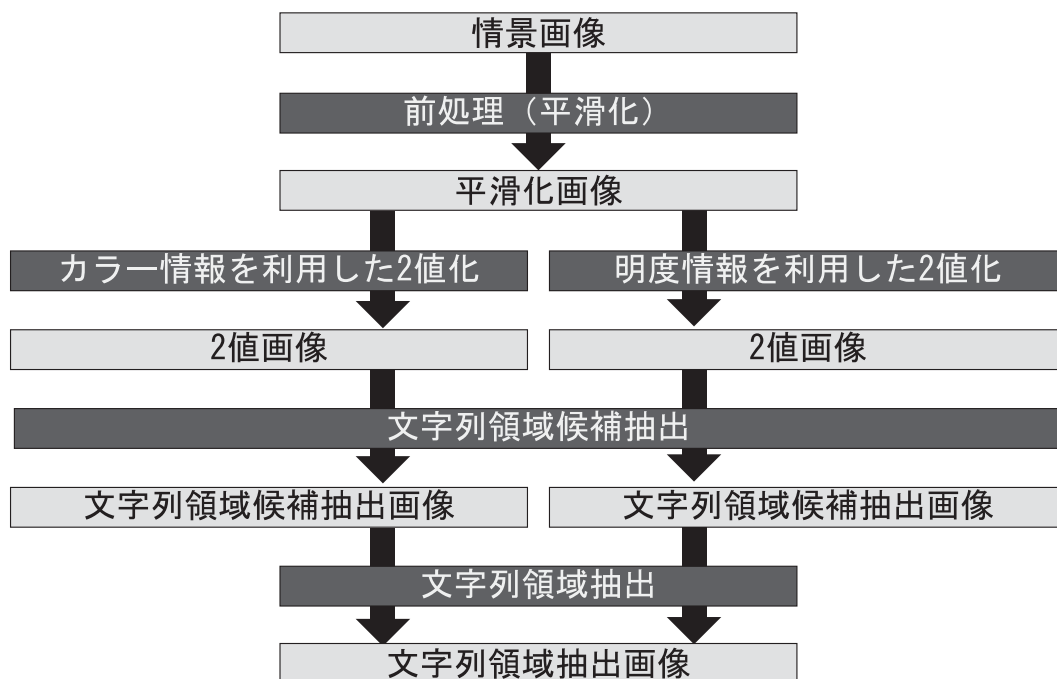


図 2.1: 処理手順の流れ

第3章

前処理 (平滑化)

情景画像は複数のオブジェクトで構成されている。しかし、情景画像内で同じオブジェクトを表している画素が必ずしも似たような RGB 値を保持しているとは限らない。これは照明の影響などオブジェクト外の原因がある。また情景画像を取得する際に、何らかの原因によって画像中にノイズが混入してしまう場合もある (図 3.1 参照)。同一オブジェクトが似たような RGB 値を保持していない場合、文字とそれ以外の部分とを良好に分解する 2 値画像を作成しようとする際に、同一オブジェクトであるにも関わらず別領域として 2 値化されてしまう。またノイズが混入している場合、同一オブジェクトを無駄に分割したり、統合してしまい良好な文字列領域の抽出を困難にさせる。

そこで上下左右 4 方向からの RGB 値の変化分に対し、明度値にメディアンフィルタを掛けた値の差分で重み付けをして平滑化を行う WMFAD(Wighted Median Filter based Anisotropic Diffusion)[6] を情景画像に施す。

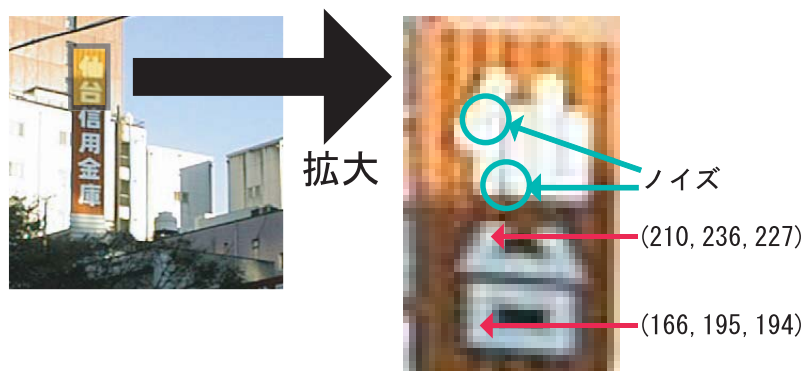


図 3.1: WMFAD

WMFAD による平滑化後の画素値は以下の式により求められる。

$$I(t+1) = I(t) + \lambda \sum_D c_D \nabla_D I(t) \quad (3.1)$$

ここで $I(t)$ は注目画素の RGB 値、 $I(t+1)$ は平滑化後の RGB 値、 D は上下左右の 4 方向、 $\nabla_D I(t)$ は 4 近傍の画素と注目画素の $I(t)$ の差、 c_D は 4 近傍に対する重みを表し

$$c_D = \exp(-(\nabla_D \text{median}(Y)/K)^2) \quad (3.2)$$

で表される。 $\text{median}(Y)$ は明度値にメディアンフィルタを掛けた値、 λ, K は定数を表す。

式 (3.1) において、 c_D は式 (3.2) から明らかなように注目画素と 4 近傍の画素との $\text{median}(Y)$ の差分に反比例する値となる。そのため注目画素と 4 近傍の画素との $\text{median}(Y)$ の差分が大きい画素の $\nabla_D I(t)$ はあまり反映されず、差分が小さい画素の $\nabla_D I(t)$ が反映されるため、局所的なエッジ部分を残したまま平滑化することができる (図 3.2 参照)。また c_D は $\text{median}(Y)$ という明度値にメディアンフィルタを掛けた値で決定されるため、ノイズ画素においても 4 近傍の画素と近い $\text{median}(Y)$ 値を保持し、4 近傍の画素の $I(t)$ が反映されることによりノイズを除去することができる (図 3.3 参照)。

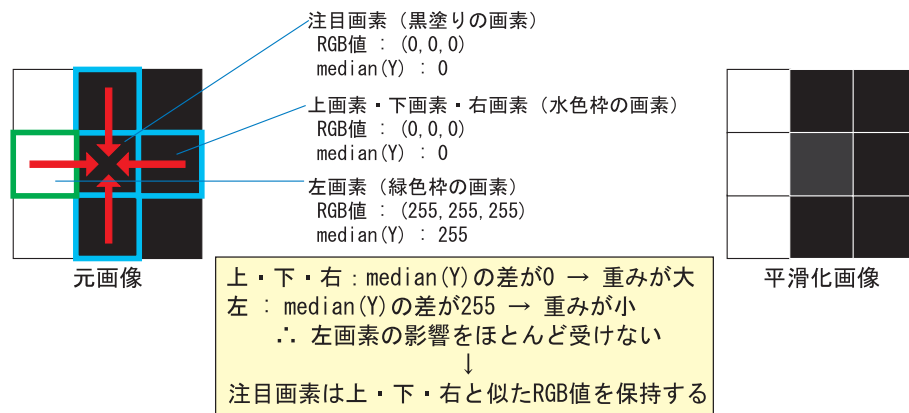


図 3.2: WMFAD によるエッジ部分を残したまま平滑化する例

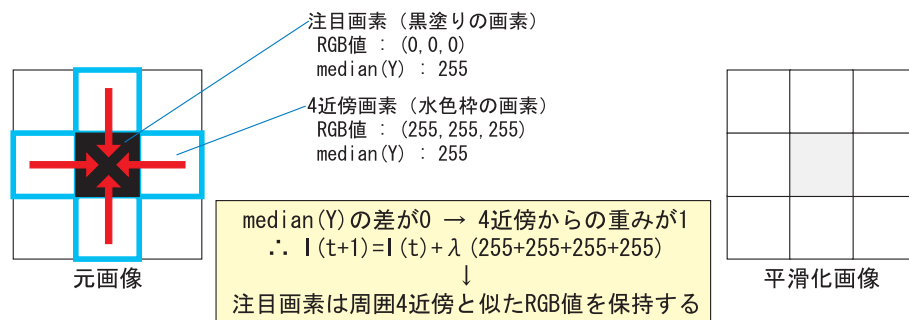


図 3.3: WMFAD の平滑化によるノイズ除去の例

実際に図 3.1 の画像に関して WMFAD による平滑化を行った例を図 3.4 に示す。WMFAD により同じオブジェクトを表す画素が似たような RGB 値を保持しており、かつノイズを除去できていることがわかる。

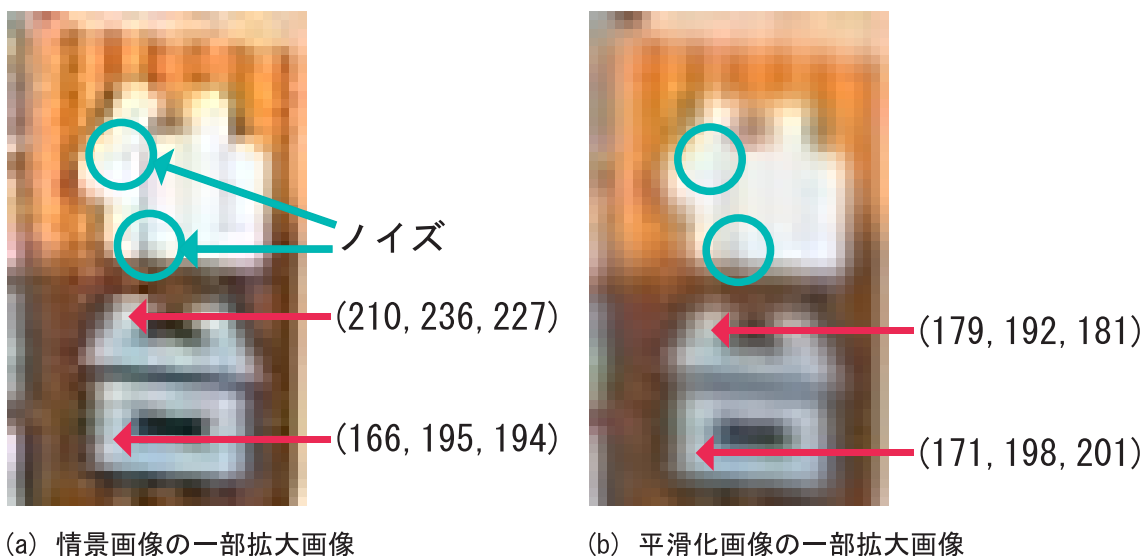


図 3.4: WMFAD

取得した情景画像に対し前処理として WMFAD による平滑化を行うことで、文字というオブジェクトや看板というオブジェクトなどを表す画素がそれぞれ似たような RGB 値を保持し、かつ何らかの原因によって混入していたノイズを除去した平滑化画像を作成することができる。これにより作成された平滑化画像を利用してそれぞれのオブジェクトを分類し、文字とそれ以外の部分とを良好に分解した 2 値画像を作成する。

第4章

2値画像作成

4.1 カラー情報を利用した2値画像作成

4.1.1 緒言

カラー情報を利用した2値画像作成の従来手法として2003年のWangらの手法[5]がある。この手法は正規化RGB空間上での正規化R、正規化G及び明度値の3種類のヒストグラム上において、オブジェクトを分割する閾値はヒストグラムにおいてある程度の深さを持った谷を表すとして histogram-based valley-searching という手法を利用してヒストグラム上から谷を探索し、その谷を分割位置として複数枚の2値画像を作成する手法である。しかし、照明の影響により文字と背景部分の色が似てしまい、文字と背景というオブジェクトを良好に分割する閾値がヒストグラム上において明確な谷の特徴を表さない場合や、画像中の文字列領域が小さいためにヒストグラム上において文字と背景というオブジェクトを良好に分割する閾値での谷がある程度の深さを持たない場合に、オブジェクトの分割位置を良好に選択できないため、文字とそれ以外の部分とを良好に分解する2値画像を作成することが困難となる。

そこで本研究では平滑化画像に対し、まず $L^*a^*b^*$ 空間上においてカラークラスタリングを行い、クラスタリング画像を作成する。そして作成されたクラスタリング画像の正規化R、正規化G及び明度値の3種類のヒストグラムを作成し、ヒストグラム上の極大値を利用してオブジェクトの分割位置を複数個決定することでオブジェクトクラスタリングを行い2値画像を複数枚作成する。その中から複雑度を評価基準として文字とそれ以外の部分とを良好に分解していると考えられる2値画像を選択することで複数枚の2値画像を作成する。

4.1.2 カラークラスタリング

平滑化画像に対してカラークラスタリングを行う。まず平滑化画像中の全画素を以下の式によって $L^*a^*b^*$ 空間に投影する (図 4.1(a))。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4.1)$$

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 & \text{if } \left(\frac{Y}{Y_n} > 0.008856 \right) \\ 903.29 \left(\frac{Y}{Y_n} \right) & \text{otherwise} \end{cases}$$

$$a^* = 500 \left[f \left(\frac{X}{X_n} \right) - f \left(\frac{Y}{Y_n} \right) \right] \quad (4.2)$$

$$b^* = 200 \left[f \left(\frac{Y}{Y_n} \right) - f \left(\frac{Z}{Z_n} \right) \right]$$

$$\text{where } \begin{cases} f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } (t > 0.008856) \\ 7.787 \times t + \frac{16}{116} & \text{otherwise} \end{cases} \\ X_n = 242.36628, Y_n = 255.00000, Z_n = 277.63227 \end{cases}$$

$L^*a^*b^*$ 空間は視覚的に似ている画素が空間上で近い位置にプロットされるという特徴を持った空間である。また平滑化画像は WMFAD を行ったことで同一オブジェクトが似たような RGB 値を保持している画像である。そのため $L^*a^*b^*$ 空間内において同一オブジェクトの画素は近い距離に投影される。

全画素が投影された $L^*a^*b^*$ 空間において、投影された画素における $L^*a^*b^*$ の最大値、最小値を求め、それを利用して全画素を囲む直方体を作成する (図 4.1(b))。次に直方体の $L^*a^*b^*$ 各辺を K 分割 (実験では $K = 30$) し、直方体の中に K^3 個の Cube を作成する (図 4.1(c))。そして各 Cube 内に存在する画素数をカウントし、周囲 26 個の Cube よりも Cube 内に存在する画素数が多い Cube を選択し、選択された Cube の重心を代表色とする (図 4.1(d))。ここで周囲 26 個の Cube とは注目 Cube に対して $3 \times 3 \times 3$ の Cube、つまり 27 個の Cube を考えた際に、中央に存在する Cube を注目 Cube として、その周囲の 26 個の Cube のことを指す。

そして各画素において全代表色との $L^*a^*b^*$ 空間内での距離を測定し、最も距離の近い代表色へと変換する (図 4.1(e))。そして代表色の $L^*a^*b^*$ 値を式 (4.3) で RGB 値へ逆変換しクラスタリング画像を作成する。

$$X = X_n \times \left(\frac{L^* + 16}{116} + \frac{a^*}{500} \right)^3$$

$$Y = Y_n \times \left(\frac{L^* + 16}{116} \right)^3 \quad (4.3)$$

$$Z = Z_n \times \left(\frac{L^* + 16}{116} - \frac{b^*}{200} \right)^3$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.240479 & -1.537150 & -0.498535 \\ -0.969256 & 1.875992 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4.4)$$

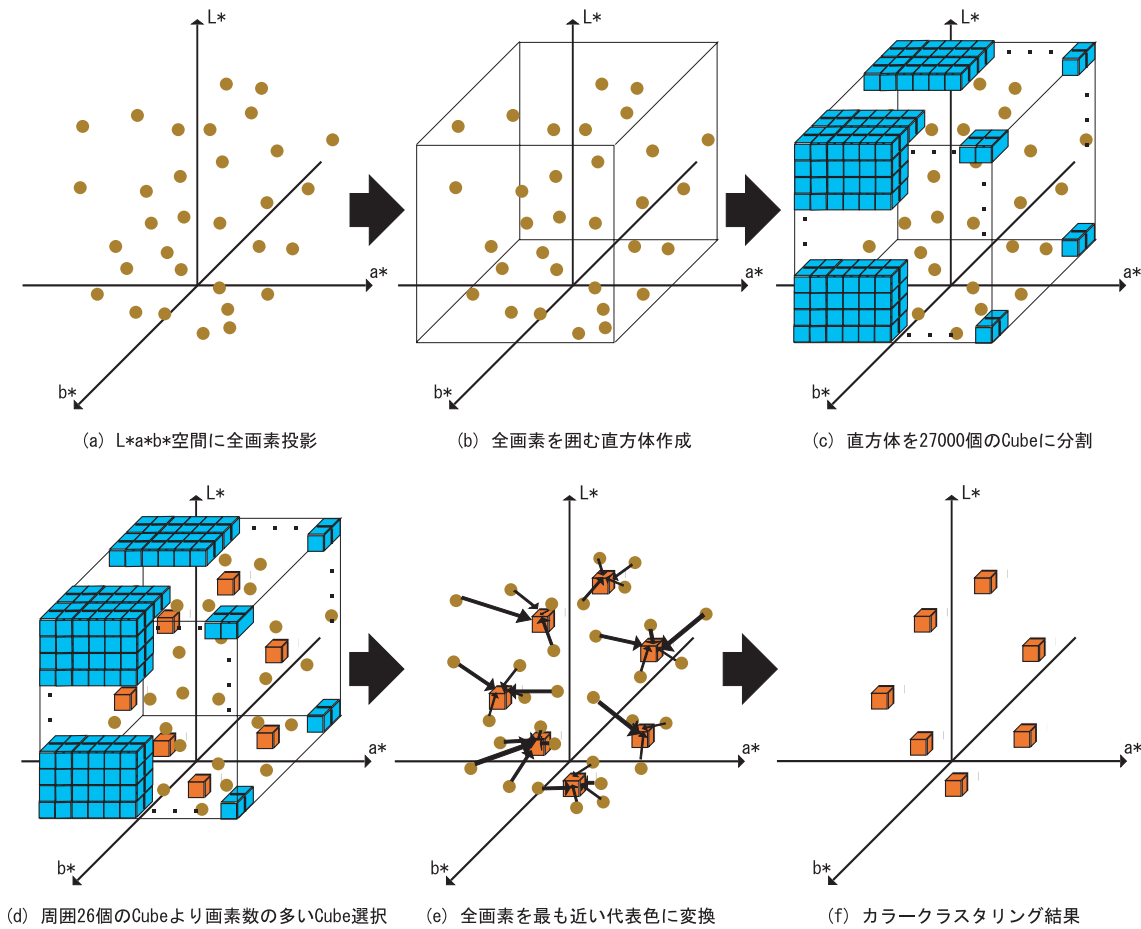


図 4.1: カラークラスタリング

このカラークラスタリングを行うことで、似たような RGB 値を保持している画素が同じ代表色へと変換されるため、平滑化画像において似たような RGB 値を保持している画素がクラスタリング画像において同じ RGB 値を保持するようになる。また周囲 26 個の Cube よりも Cube 内に存在する画素数が多い Cube を選択するため平滑化画像において画素数の少ないオブジェクトも代表色として選択される。

同様のカラークラスタリング手法として Cube のサイズを固定する手法 [7] がある。[7] では $L^*a^*b^*$ 空間が実際取り得る範囲を直方体で囲み、その中に固定長の Cube を作成し代表色を選択する手法である。しかし入力画像によって $L^*a^*b^*$ 空間内での分布の広がりは異なり、固定長の Cube では代表色を良好に選択することが困難な場合があるため、本研究では投影された画素における $L^*a^*b^*$ 値の最大値、最小値を求め直方体を作成し、作成した直方体を K^3 個の Cube へと分割する手法をとった。これにより全画素の $L^*a^*b^*$ 空間内での分布が狭い場合でも複数の代表色を選択

することが可能となる。

このクラスタリング画像の正規化 R、正規化 G 及び明度値ヒストグラムを利用して次にオブジェクトクラスタリングを行う。

4.1.3 オブジェクトクラスタリング

情景画像に対し前処理を行い、同一オブジェクトを表している画素が似たような RGB 値を保持するような平滑化画像を作成した。そしてその平滑化画像に対し、カラークラスタリングを行ったことで、似たような RGB 値を保持する同一オブジェクトを表す画素は同じ代表色へと変換される。しかし入力画像はどのような文字の色をしているか、またどのような照明条件下で撮影されたかなどは未知である。そのため同じ文字列上の文字でも陰影が掛かっているために平滑化画像内でも同じオブジェクトが異なる RGB 値を保持し、カラークラスタリングにおいて異なる代表色に変換されることがある (図 4.2 参照)。そのため文字というオブジェクトを構成している代表色とそれ以外の部分を構成している代表色とを良好に分類するためにオブジェクトクラスタリングを行い、文字とそれ以外の部分とを良好に分解した 2 値画像を作成する。



図 4.2: カラークラスタリングによって作成されたクラスタリング画像及び文字領域の拡大画像

オブジェクトクラスタリングは $L^*a^*b^*$ 空間上ではなく、正規化 RGB 空間上で行うため、まず $L^*a^*b^*$ 空間上でのカラークラスタリングによって作成されたクラスタリング画像の RGB 値を図 4.3 に示す対応で正規化 RGB 値へと変換し、正規化 RGB 空間上でのヒストグラムを作成する。オブジェクトクラスタリングを $L^*a^*b^*$ 空間上ではなく正規化 RGB 空間上で行う理由についてはこの節の最後に述べる。クラスタリング画像の RGB 値から正規化 RGB 値 (r, g, b) へは以下の式によって変換される。

$$(r, g, b) = \begin{cases} (0, 0, 0) & \text{if } (R + G + B = 0) \\ \left(\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B} \right) = \left(\frac{R}{R+G+B}, \frac{G}{R+G+B}, 1 - r - g \right) & \text{otherwise} \end{cases} \quad (4.5)$$

正規化 RGB 空間は図 4.3 のように表されるので正規化 R 及び正規化 G のヒストグラムのみを考慮すればよい。作成された正規化 R 及び正規化 G のヒストグラム上から高さが 100 以上の極大値を全て探索する。そして正規化 R 及び正規化 G のヒストグラム上で極大値を示す部分がオブジェ

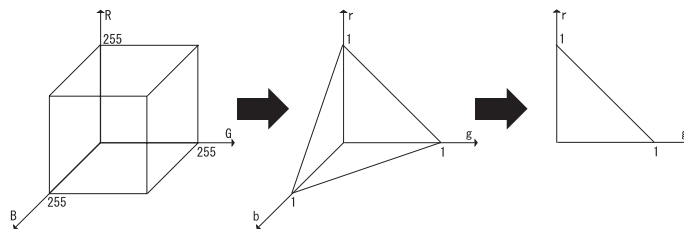


図 4.3: 正規化 RGB 空間

クトを表している値であると仮定し、オブジェクトの分割位置を隣り合う極大値の平均値に設定する。そして正規化 R 及び正規化 G のヒストグラム上で設定されたそれぞれのオブジェクトの分割位置の組合せをオブジェクトクラスタとし、複数枚の 2 値画像を作成する。オブジェクトクラスタは rD, gD をオブジェクトの分割位置 (図 4.4 参照) として以下の式で表される。

$$objectcluster(m, n) = \sum_{r=rD(m)}^{rD(m+1)} r(x, y) \cap \sum_{g=gD(n)}^{gD(n+1)} g(x, y) \quad (4.6)$$

仮に正規化 R ヒストグラム上に 3 個の極大値が存在し、正規化 G ヒストグラム上に 4 個の極大値が存在した場合を考える。このとき正規化 R ヒストグラム上には 4 個のオブジェクト分割位置が、また正規化 G ヒストグラム上には 5 個のオブジェクト分割位置が設定される (図 4.4)。オブジェクトクラスタは図 4.4 のように正規化 R 及び正規化 G の 2 次元平面上で表される。このときオブジェクトクラスタは 30 個であるため、2 値画像は 30 枚作成される。

また入力画像がグレースケールの場合、正規化 RGB 空間上には特徴が現れないため、正規化 R 及び正規化 G ヒストグラム上でのオブジェクトクラスタリングでは文字とそれ以外とを良好に分解する 2 値画像を作成することができない。そこでクラスタリング画像から明度値ヒストグラムを作成し、正規化 R 及び正規化 G ヒストグラムと同様にオブジェクト分割位置を決定し、そのオブジェクト分割位置を閾値として 2 値画像を作成する。

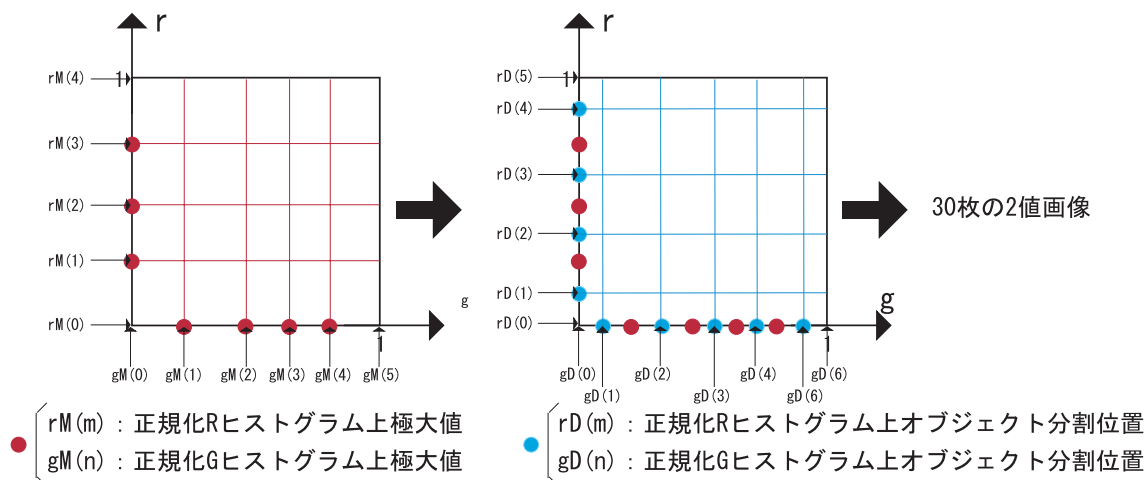


図 4.4: オブジェクトクラスタリング

このオブジェクトクラスタリングをクラスタリング画像の正規化 R、正規化 G 及び明度値のヒストグラム上で行うことによって、複数の代表色で構成された文字をそれ以外の部分と良好に分解した 2 値画像を作成することが可能となる。

ここでオブジェクトクラスタリングを $L^*a^*b^*$ 空間上で代表色を利用して行わず、 $L^*a^*b^*$ 空間から RGB 値へと変換し作成したクラスタリング画像の正規化 RGB 空間上で行った理由について述べる。仮に代表色が 10 個選択されたとすると、画像中の文字はいくつの代表色から構成されているか未知なので、 $L^*a^*b^*$ 空間上の代表色でオブジェクトクラスタリングを行うと計 637 枚 (${}_{10}C_1 + {}_{10}C_2 + {}_{10}C_3 + {}_{10}C_4 + {}_{10}C_5$) の 2 値画像が作成される。これを正規化 RGB 空間上で行った場合、最大で正規化 R、正規化 G 及び明度値ヒストグラム上にそれぞれ 10 個の極大値が現れるため計 132 枚 ($11 \times 11 + 11$) の 2 値画像が作成されることになり、 $L^*a^*b^*$ 空間上で代表色を利用してオブジェクトクラスタリングを行った場合と比較すると作成される 2 値画像が少数で済むことがわかる。実際に図 4.2 のクラスタリング画像の場合、選択される代表色は 45 個存在する。そのため $L^*a^*b^*$ 空間でオブジェクトクラスタリングを行うと膨大な枚数の 2 値画像が作成されてしまう。しかし、正規化 RGB 空間上でオブジェクトクラスタリングを行った場合、正規化 R ヒストグラム上に 13 個、正規化 G ヒストグラム上に 5 個、明度値ヒストグラム上に 21 個のオブジェクト分割位置が決定されるため、正規化 R 及び正規化 G ヒストグラムから 48 枚 ($(13 - 1) \times (5 - 1)$) の 2 値画像が、そして明度値ヒストグラムから 20 枚の 2 値画像が作成され、計 68 枚の 2 値画像で済む。

ここでオブジェクトクラスタリングを $L^*a^*b^*$ 空間上で代表色を利用して行った場合と正規化 RGB 空間上で行った場合で作成される 2 値画像が 505 枚の差があることがわかる (仮に代表色数を 10 とした場合)。この 505 枚に関しては入力画像上で正規化 R、正規化 G 及び明度値全てにおいて差がないオブジェクトを分類していることになるので無駄な 2 値画像であると考えられる。よって本研究では $L^*a^*b^*$ 空間上で代表色を利用してオブジェクトクラスタリングを行わず、代表色を RGB 値に変換し正規化 R、正規化 G 及び明度値ヒストグラム上でオブジェクトクラスタリングを行うことで 2 値画像を複数枚作成した。

4.1.4 複雑度による 2 値画像評価

カラークラスタリングとオブジェクトクラスタリングによって 2 値画像を作成する場合、入力画像によってはオブジェクトがほとんど存在しないような 2 値画像も多く作成される。そこで作成された 2 値画像の境界線の長さから複雑度を計算し、文字とそれ以外とを良好に分解していると考えられる 2 値画像を選択する。2 値画像の複雑度は以下の式で計算される。

$$complexity = \frac{boundary}{2 \times height \times (width - 1)} \quad (4.7)$$

ここで *boundary* は画像中の境界線の長さを 4 連結でカウントしたもの、また *height* 及び *width* は画像 1 辺の高さと幅を表す。

上式によって計算された複雑度が $\theta_c (=0.01)$ 以上である 2 値画像を文字とそれ以外の部分とを良好に分解している 2 値画像であるとして選択する。境界線の長さによる複雑度の閾値 θ_c は予備実

験により $\theta_c = 0.01$ とし。これは画像内に存在する文字列が小さい場合でも境界線の長さによる複雑度は 0.01 以上になり、微小なノイズ部分のみが 2 値化されているような画像は 0.01 以下になるという結果から設定した。実際に図 4.2 の場合、オブジェクトクラスタリングによって 68 枚の 2 値画像が作成されていた。これらの 2 値画像に境界線の長さによる複雑度によって評価を行うと、選択された 2 値画像は 25 枚で済んだ。

これによりカラー情報を利用して、平滑化画像から文字とそれ以外の部分とを良好に分解している 2 値画像を複数枚作成する。

このカラー情報を利用した 2 値画像作成法により、照明の影響により文字とその背景部分の色が似てしまい、直接的な色ヒストグラム上では文字と背景というオブジェクトを良好に分解する閾値付近に明確な特徴が現れないために文字とそれ以外の部分とを良好に分解する 2 値画像を作成することが困難であった画像においても、カラークラスタリングを行い、見ために差があるオブジェクトを異なる代表色へと変換することで、オブジェクトの明確な分割位置を決定することができ、良好な 2 値画像を作成することができる。また画像中の文字列が小さいために直接的な色ヒストグラム上で文字とそれ以外の部分とを良好に分解する閾値付近に小さな特徴しか現れないために、文字とそれ以外の部分とを良好に分解する 2 値画像を作成することが困難であった画像においても、カラークラスタリングにおいて周囲 26 個の Cube よりも Cube 内の画素数が多い Cube を代表色として選択するため、小さなオブジェクトも無視することなく選択でき、良好な 2 値画像を作成することができる。

4.2 明度情報を利用した 2 値画像作成

4.2.1 緒言

明度情報を利用した 2 値画像作成の従来手法として 1997 年の松尾らの手法 [3] がある。この手法は、まず入力画像から明度値を計算して作成される明度情景画像に対し、閾値を 0 ~ 255 まで変化させ計 256 枚の 2 値画像を作成する。作成された各 2 値画像において、平均隣接数による評価を行い、各閾値で作成された 2 値画像の単純さを測定し、文字とそれ以外の部分とを良好に分解している 2 値画像は単純であるという仮定により複数個の閾値を選択することで 2 値画像を作成する手法である。しかし、文字列とその背景部分との明度値が似ているような画像の場合、明度情報のみでは文字とそれ以外の部分とを良好に分解する 2 値画像を作成することが困難になる。また照明の影響により文字列部分に陰影が掛かってしまい、一つの文字上で明度値に大差が存在するような画像の場合、文字とそれ以外の部分とを良好に分解する閾値を選択することが困難となる。

そこで本研究では平滑化画像から、まず明度値を計算し明度情景画像を作成する。そして作成された明度情景画像を小領域に分割し、各小領域における 4 つの参照領域から NAT 法 [8] を利用して 4 つの領域閾値を設定する。次に各小領域に所属している各画素において正規化 RGB 空間上での平均色を計算し、各画素に対し 4 つの領域閾値から 1 つの画素閾値を決定する。このようにして全画素において閾値を設定し、1 枚の 2 値画像を作成する。

4.2.2 画素閾値決定

(1) 画像分割及び参照領域決定

まず平滑化画像から明度値を以下の式により計算し、明度情景画像を作成する。

$$L = (R + G + B) / 3 \quad (4.8)$$

次に作成された明度情景画像を 4×4 pixel の小領域に分割する。そして各小領域において図 4.5 のように参照領域を 4 つ決定し、各参照領域において NAT 法を利用することで注目小領域に対し領域閾値を 4 個決定する。NAT 法については次の (2) 節で説明する。

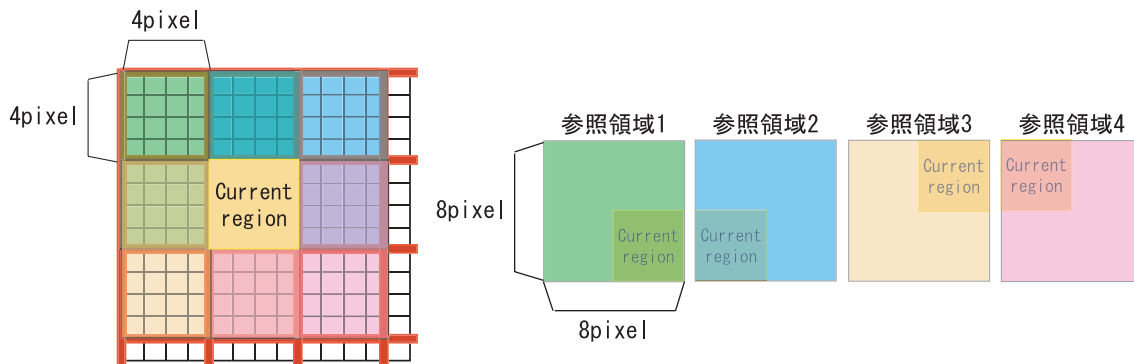


図 4.5: 参照領域

(2) NAT 法による領域閾値決定

各小領域に対する 4 つの参照領域それぞれにおいて NAT 法を利用することで、参照領域内に存在するオブジェクトを良好に 2 値化する閾値を決定する。

NAT 法は 2001 年に Hon-Son Don により提案された手法 [8] で、図 4.6 のような文字と背景と裏書き文字の 3 種類のオブジェクトが存在し、画像の各明度値における画素数を表すヒストグラムを利用して 2 値化を行うと良好な 2 値画像作成が困難であるような画像において、ノイズの特性を利用して 2 値化することで文字とそれ以外の部分とを良好に分解する 2 値画像を作成する手法である。

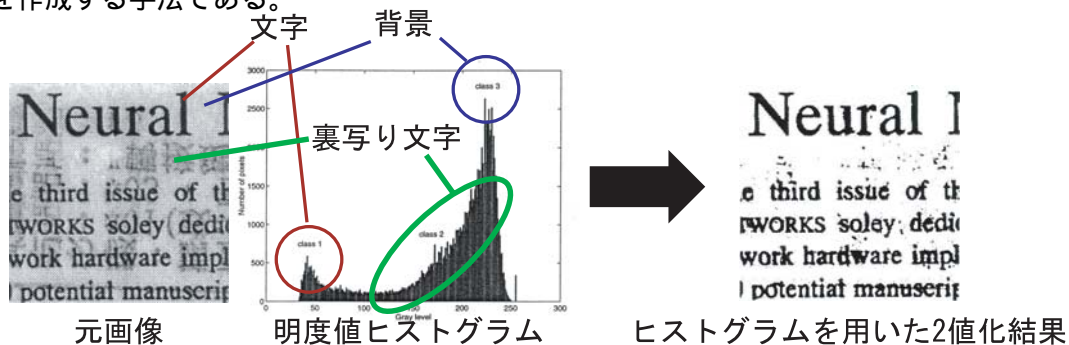


図 4.6: オブジェクトを良好に分解できない例

NAT 法の説明のため図 4.7 の赤い線上の明度値ヒストグラムを考える。

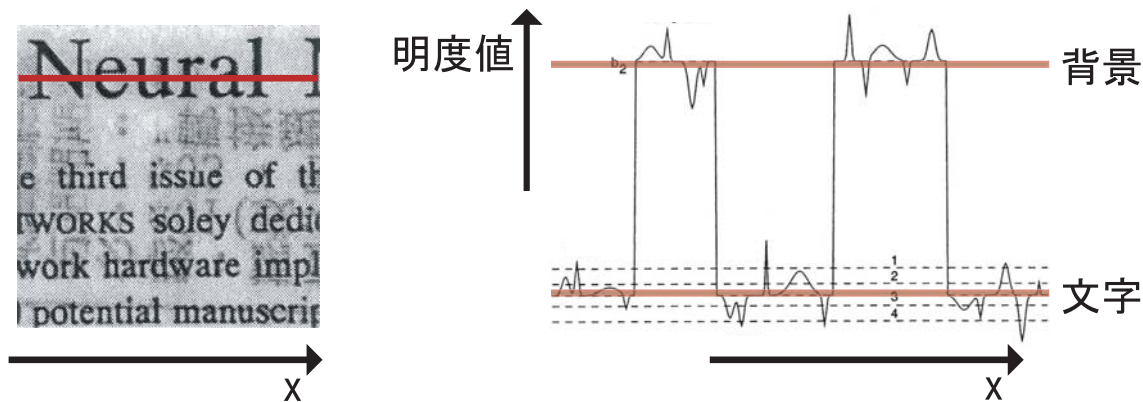


図 4.7: 準備

図 4.7 において図 4.8(a-1) のように閾値を設定した場合、2 値画像は図 4.8(a-2) のようになり、2 値画像は 5 領域から構成される。この閾値を図 4.8(b-1) のようにオブジェクトである文字の付近に設定した場合、2 値画像は図 4.8(b-2) のようになり、2 値画像は 11 領域で構成される。また同様に閾値を図 4.8(c-1) のように設定すると 2 値画像は図 4.8(c-2) のようになり、2 値画像は 13 領域で構成される。このことから閾値をオブジェクトである文字の付近の閾値に設定すると 2 値画像を構成する領域数が増加することがわかる。これはオブジェクトが同一の明度値で存在するのではなく似たような明度値で存在するために、オブジェクト付近の閾値を選択すると図 4.7 の明度値ヒストグラム上に現れているようなノイズの影響を受けるために起こる現象である。同様に閾値を背景オブジェクト付近に設定しても領域数が増加する。つまり閾値を変化させながら 2 値画像を作成すると画像内に存在するオブジェクト付近で 2 値画像を構成する領域数が増加するという特徴があるということがわかる。この特徴を利用して閾値に対し領域数を表すヒストグラムを作成し、画像内に存在するオブジェクトを良好に分解する閾値を選択する手法が NAT 法である。

実際に、図 4.7 の画像に対し NAT 法を適用し、閾値に対して領域数を表すヒストグラムを作成すると、図 4.9 のようになり、文字と背景と裏書き文字の明度値付近で明らかに領域数が増加していることがわかる。このヒストグラムを利用して 2 値化を行うことで、文字とそれ以外の部分とを良好に分解する 2 値画像を作成することができる (図 4.9)。

平滑化画像は同じオブジェクトを表す画素が似たような RGB 値を保持しているため、この NAT 法を各小領域において決定された 4 つの参照領域に適用することで各参照領域内においてオブジェクトを良好に分解する閾値を求めることができる。このようにして各参照領域から 4 個の領域閾値を得る。次にこのようにして得た 4 個の領域閾値から各小領域に所属する各画素に対して画素閾値を決定する。

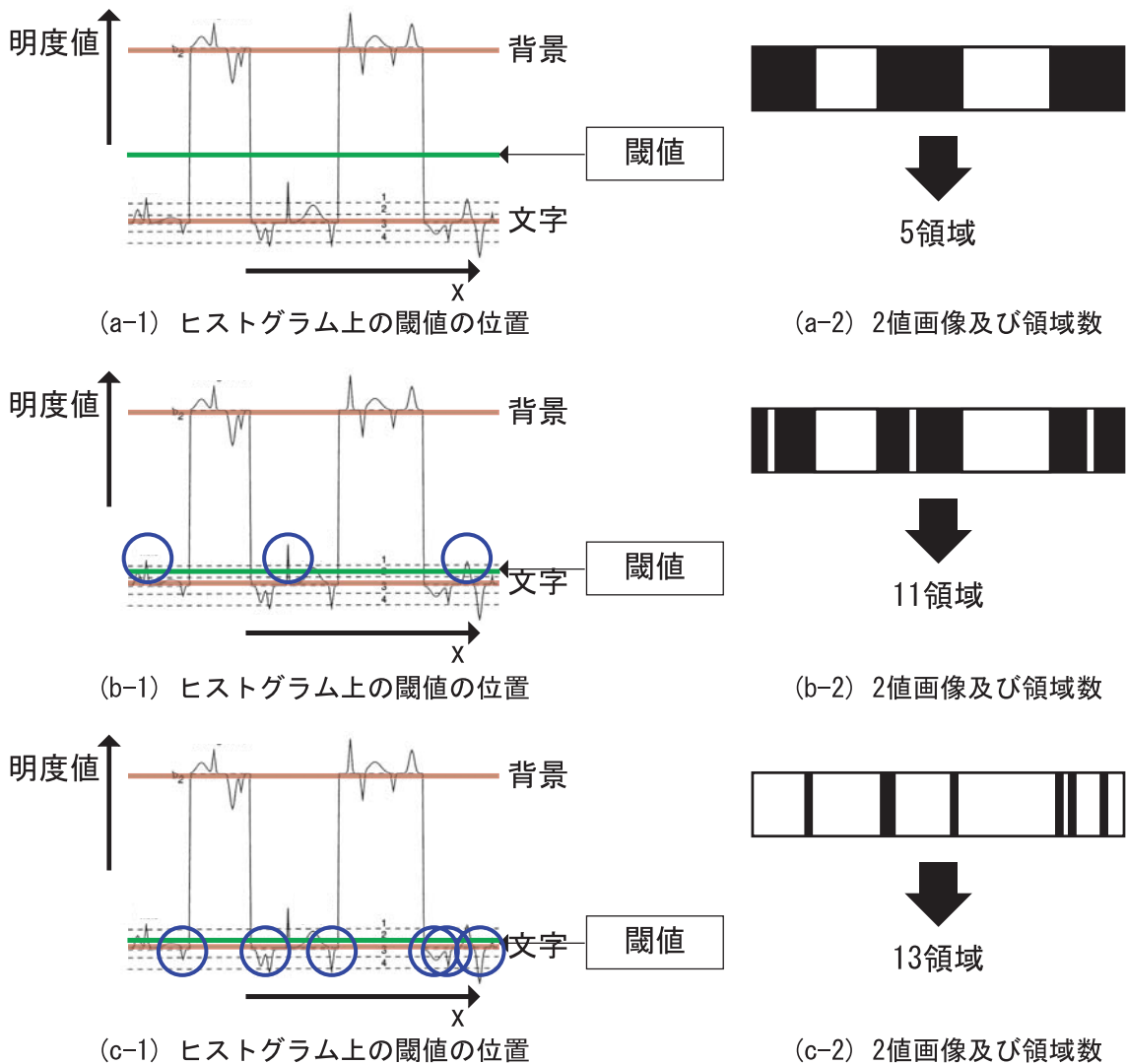


図 4.8: 閾値を変化させた場合の2値画像

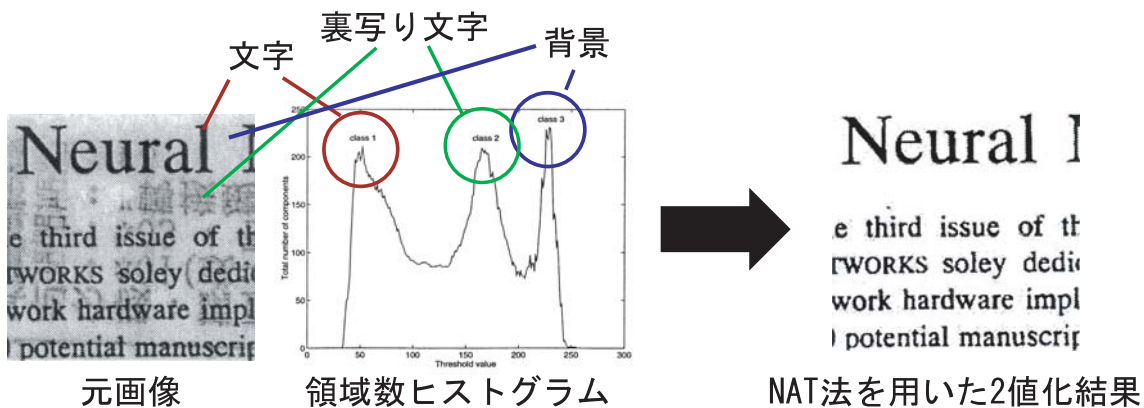


図 4.9: NAT 法による2値画像作成

(3) 色比較による画素閾値決定

各小領域において NAT 法を利用することで 4 個の領域閾値が決定された。これにより各小領域に所属する各画素においても 4 個の領域閾値が決定されたことになる。この各画素に決定された 4 個の領域閾値から、今度は正規化 RGB 空間上での平均色を利用することで各画素に対する画素閾値を決定する。

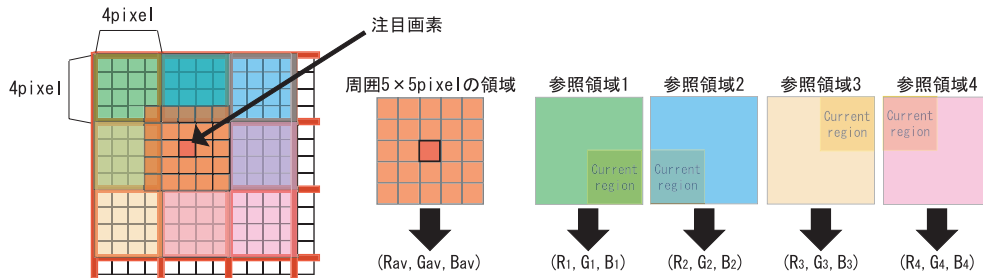


図 4.10: 画素閾値決定法における平均色

まず図 4.10 で示されるように注目画素 $c(i, j)$ の周囲 5×5 pixel の正規化 RGB 空間上での平均色 (R_{av}, G_{av}, B_{av}) を以下の式により求める。

$$\begin{aligned}
 R_{av} &= \sum_{a=i-2}^{i+2} \sum_{b=j-2}^{j+2} \frac{R_{ab}}{R_{ab} + G_{ab} + B_{ab}} \\
 G_{av} &= \sum_{a=i-2}^{i+2} \sum_{b=j-2}^{j+2} \frac{G_{ab}}{R_{ab} + G_{ab} + B_{ab}} \\
 B_{av} &= \sum_{a=i-2}^{i+2} \sum_{b=j-2}^{j+2} \frac{B_{ab}}{R_{ab} + G_{ab} + B_{ab}}
 \end{aligned} \tag{4.9}$$

同様にして注目画素の領域閾値を決定した 4 つの各参照領域の正規化 RGB 空間上での平均色 (R_n, G_n, B_n) ($n = 1 \sim 4$) を以下の式により求める。

$$\begin{aligned}
 R_n &= \sum_a \sum_b \frac{R_{ab}}{R_{ab} + G_{ab} + B_{ab}} \\
 G_n &= \sum_a \sum_b \frac{G_{ab}}{R_{ab} + G_{ab} + B_{ab}} \\
 B_n &= \sum_a \sum_b \frac{B_{ab}}{R_{ab} + G_{ab} + B_{ab}}
 \end{aligned} \tag{4.10}$$

そして注目画素の周囲 5×5 pixel の平均色 (R_{av}, G_{av}, B_{av}) と各参照領域での平均色 (R_n, G_n, B_n) ($n = 1 \sim 4$) との距離 D_n を以下の式により求める。

$$D_n = \sqrt{(R_n - R_{av})^2 + (G_n - G_{av})^2 + (B_n - B_{av})^2} \tag{4.11}$$

この $D_n (n = 1 \sim 4)$ が最小である参照領域において求められた閾値 TH_n に対し大きな重みを付加することで、注目画素の画素閾値を決定する。ここで重みは予備実験により 1.2 とした。仮に図 4.10 の注目画素において、 D_3 が最小であった場合、注目画素の画素閾値 T_{ij} は以下の式によって求まる。

$$T_{ij} = \frac{TH_1 + TH_2 + 1.2 \times TH_3 + TH_4}{4.2} \quad (4.12)$$

このようにして各画素に決定された 4 個の領域閾値から平均色を利用することで画素閾値を求める。

この画素閾値を全画素において求め、1 枚の 2 値画像を作成する。

4.2.3 画像補正

各画素において画素閾値を決定し 2 値画像を作成すると図 4.11 のようにオブジェクトを余計に連結してしまうような画素が存在する場合がある。このように文字と他のオブジェクトとが余計に連結された場合、文字は他のオブジェクトと同領域と見なされ、文字を抽出することを困難にさせる。

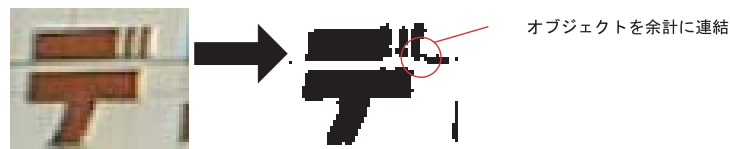


図 4.11: オブジェクトを余計に連結してしまう画素

そこで図 4.12 のように注目画素に対し、上下の画素値及び左右の画素値がそれぞれ同じ値を保持し、かつ上下の画素値と左右の画素値が異なる画素値を保持している場合、注目画素を無視する。これにより 1 pixel で存在しているような線を排除することができ、オブジェクトを余計に連結してしまっているような画素を無視することができる。実際に図 4.11 でオブジェクトを余計に連結した画素が存在している画像に対し、画像補正を行うと図 4.13 のようになり、オブジェクトを余計に連結している画素を無視することができる。

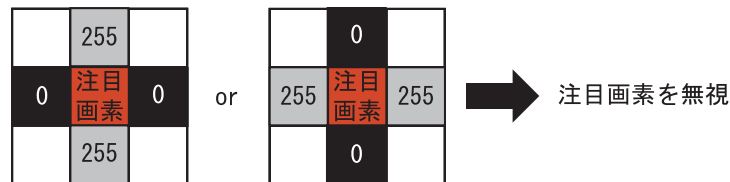


図 4.12: 画像補正

これにより明度情報を利用して、平滑化画像から文字とそれ以外の部分とを良好に分解する 2 値画像を 1 枚作成する。



図 4.13: 画像補正結果例

この明度情報を利用した2値画像作成法により、照明の影響により文字列とその背景部分の明度値が似ているために、画像全体のヒストグラムから2値化を行うと文字列とその背景部分とを良好に分解する閾値付近の特徴が小さく、文字とそれ以外の部分とを良好に分解する2値画像作成が困難であった画像においても、明度情景画像を小領域に分割し各小領域ごとに領域閾値を求めることにより、その領域以外の影響を受けることなく2値化することが可能なため、良好な2値画像を作成することができる。同様に画像中の文字列が小さいために文字以外の部分の影響を受けてしまい良好な2値画像を作成することが困難であった画像においても、小領域ごとに領域閾値を求め2値化することで文字とそれ以外の部分とを良好に分解する2値画像を作成することができる。また照明が文字列上に入り込み、一つの文字上で明度値に大差が存在し一つの文字を同じ領域として2値化することができず、文字とそれ以外とを良好に分解する2値画像作成が困難であった画像においても、小領域ごとに周囲の参照領域から複数の領域閾値を決定し、さらに各画素において平均色を利用して画素閾値を求めることで2値化することにより、一つの文字内で明度値の高い画素と低い画素とでそれぞれ異なる閾値を設定することができるため、良好な2値画像を作成することができる。

第5章

文字列領域の抽出

5.1 緒言

これまでに入力された情景画像に対し前処理として平滑化を行い平滑化画像を作成し、作成された平滑化画像からカラー情報を利用して複数枚の2値画像を、明度情報を利用して1枚の2値画像を作成した。これらの作成された全2値画像から文字列領域を高精度に抽出するために、本研究ではまず文字を表している領域は黒画素または白画素の閉領域で存在しているとし、文字領域候補を抽出する。次に漢字や平仮名などの日本語は特に2つ以上の部分(例：偏と旁)から成るとして、同じ文字を表していると考えられる文字領域候補を統合するマーキング処理を行う。それによって抽出された文字領域候補の中から文字列を成す文字の連結性を考慮し、文字の領域として文字列領域候補を抽出する。このようにして各2値画像から文字列領域候補を抽出する。そしてこの各2値画像で抽出された文字列領域候補を一つの画像に統合し文字列統合画像を作成し、この文字列統合画像の特徴から文字列領域を決定し文字列領域抽出画像を作成する。

5.2 文字列領域候補抽出

5.2.1 文字領域候補抽出

これまでに文字とそれ以外の部分とを良好に分解する2値画像が作成されたため、各2値画像内において文字を表す領域は黒画素または白画素の閉領域で存在している。そこで各2値画像において、同一の画素値を持つ閉領域を4連結でラベルを与えていき同じ領域として統合する。これによって2値画像はラベルによる領域分割をされたことになる。次にこの領域分割された各閉領域の x,y 座標の最大値、最小値を求め、その最大値、最小値を始点、終点として閉領域を方形で囲み各閉領域に外接方形を与える。

このようにして各閉領域に与えられた外接方形の形状によって全閉領域を文字領域候補、部分領

域候補、漢数字候補、不用の 4 種類に分類する。それぞれに分類される条件は以下のようになる。

(1) 文字領域候補

- 短辺が 5 画素以上
- 長辺と短辺の辺比率が 1.6 未満

(2) 部分領域候補

- 短辺が 5 画素以上
- 長辺と短辺の辺比率が 1.6 以上 4.0 未満

(3) 漢数字候補

- 長辺が 5 画素以上
- 閉領域において黒画素または白画素の全画素に対する比率が 8 割以上

(4) 不用

- それ以外

これにより閉領域に与えられた外接方形の短辺が 5 画素以上で外接方形が正方形に近い閉領域は文字領域候補に、正方形の半分に近い長方形の閉領域は部分領域候補に分類される。漢数字候補に関しては、細長い閉領域を構成しており、外接方形と閉領域の形にあまり差がない場合分類される。これにより漢数字の「一」や「三」などを構成する閉領域が漢数字候補に分類される。これら 3 つに分類されない場合、その閉領域は文字以外の部分を表しているとして不用に分類する(図 5.1 参照)。

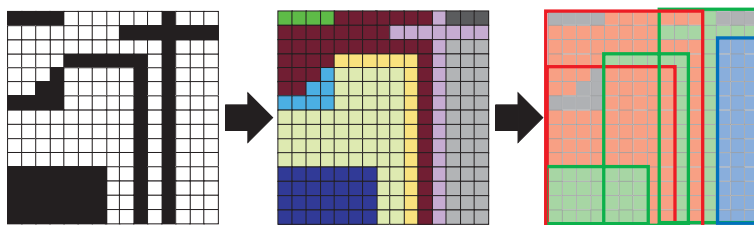


図 5.1: 4 連結による領域分割及び外接方形の形状による分類

(文字領域候補 : 赤, 部分領域候補 : 緑, 漢数字候補 : 青, 不用 : 外接方形なし)

5.2.2 マージング処理

前節において 2 値画像内に存在する閉領域を外接方形により分類し、文字領域候補、部分領域候補、漢数字領域候補を抽出した。しかし画像内に存在している文字には偏と旁から構成される漢字などがあり、1 つの文字が 2 つ以上の候補から成り立っている場合がある。偏と旁から構成されるような文字で 2 つ以上の候補から成り立っている文字を 1 つの方形で囲むようにするため外接方形を統合するマージング処理を行う。マージングとは、ある 2 つの閉領域の外接方形の位置

関係や形状などに基づいて、その2つの閉領域を統合する処理である。この外接方形のマーキング処理は、馬場口らの提案した手書き文字の文字単位ごとの切り出しによる手法 [9] を応用する。

本研究で行うマーキング処理は外接方形による分類において文字領域候補または部分領域候補に分類された閉領域に与えられた外接方形に対してのみ行う。文字領域候補または部分領域候補に分類された閉領域に対して以下のような条件を利用し、3種類のマーキング処理を行う。

(1) 文字領域候補と文字領域候補のマーキング

このマーキング処理は偏や旁が2つの領域に分離しており、それぞれが文字領域候補として抽出されているような場合の文字を想定して行う (図 5.2(a) 参照)。

- 2つの外接方形は交差
- 2つの外接方形の面積比率が 1.5 未満
- 2つの外接方形の短辺の辺比率が 1.6 未満かつ長辺の辺比率が 1.6 未満
- 2つの外接方形の上下左右の辺いずれか1つでも2画素以内のずれ
- 2つの閉領域を統合して新しく作成された外接方形が文字領域候補の条件を満たす。

(2) 文字領域候補と部分領域候補のマーキング

このマーキング処理は偏と旁のいずれかが部分領域候補として抽出されており、残りの部分が文字領域候補として抽出されているような場合の文字を想定して行う (図 5.2(b) 参照)。

- 2つの外接方形は交差または2画素以内に存在
- 2つの外接方形の面積比率が 3.0 未満
- 2つの外接方形の短辺の辺比率が 1.6 未満かつ長辺の辺比率が 2.5 未満 または 短辺の辺比率が 2.5 未満かつ長辺の辺比率が 1.6 未満
- 2つの外接方形の上下左右の辺いずれか1つでも2画素以内のずれ
- 2つの閉領域を統合して新しく作成された外接方形が文字領域候補の条件を満たす。

(3) 部分領域候補と部分領域候補のマーキング

このマーキング処理は偏と旁が2つの領域に分離しており、それぞれが部分領域候補として抽出されているような場合の文字を想定して行う (図 5.2(c) 参照)。

- 2つの外接方形は交差または2画素以内に存在
- 2つの外接方形の短辺の辺比率が 1.6 未満かつ長辺の辺比率が 2.0 未満
- 2つの外接方形の面積比率が 1.5 未満
- 2つの外接方形の上下左右の辺のいずれか1つでも2画素以内のずれ
- 2つの閉領域を統合して新しく作成された外接方形が文字領域候補の条件を満たす。

条件の意図を若干説明する。各マーキング処理の条件の1つとして「2つの外接方形の上下左右の辺のいずれか1つでも2画素以内のずれ」という条件がある。これは1つの文字の一部を表して

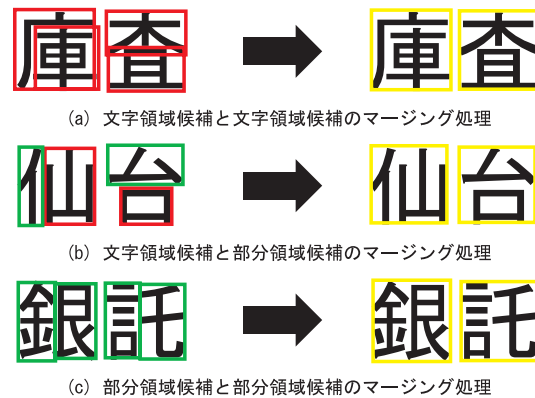


図 5.2: 各マーキング処理の例

いる閉領域であれば、水平方向（上または下）または垂直方向（左または右）のいずれかにおいて、2つの閉領域はほぼ平行に存在しているという条件を表している。

これらの条件を当てはめることによってマーキング処理を行う（図 5.2 参照）。これにより偏と旁から構成されているために 2 つ以上の候補によって成り立っていた文字を一つの方形で囲むことが可能になる。

この時点でマーキング処理されず部分領域候補となっている閉領域に関しては文字を表している閉領域ではないと判断し、不用とする。

5.2.3 文字列領域候補及び漢数字候補抽出

ここまで抽出されている文字領域候補の中から文字列を成している文字を表していると考えられる領域を文字列を成す文字の連結性を利用して文字列領域候補として抽出する。文字列を成す文字の連結性を 2 つの文字領域候補ごとに確認し、文字列領域候補を選択する。文字列を成す文字の連結性は以下ようになる。

- 2 つの外接方形の面積比率が 1.5 未満
- 2 つの外接方形の重心のずれが 5 画素以内
- 2 つの外接方形の重心の距離が $3 \times (\text{方形の長さ})$ 以内
- 2 つの外接方形の縦、横の一辺の長さの比が共に 1.2 以内

これらの条件を満たす 2 つの文字領域候補を文字列領域候補として抽出する。このとき 2 つの文字列領域候補の近くに存在する漢数字候補の抽出も同時に行う。漢数字候補の抽出条件は以下ようになる。

- 2 つの文字列領域候補と漢数字候補の重心のずれが 5 画素以内
- 2 つの文字列領域候補と漢数字候補の平均色の差が 30 以内

平均色は情景画像における正規化 RGB 空間上での外接方形内の全画素の平均を用いる。漢数字候補と 2 つの文字列領域候補は、同じ文字列上の文字である場合、色が似ているとして人間が識別困難であるとされる正規化 RGB 空間上での平均色の差が 30 以内という条件にした。

このようにして各 2 値画像において文字列を成す文字として文字列領域候補を抽出する。

5.3 文字列領域抽出

各 2 値画像において抽出された文字列領域候補を利用して文字列領域を決定する。まず各 2 値画像で抽出された文字列領域候補を 1 つの画像に統合し、文字列統合画像を作成する。この文字列統合画像を作成することで 1 枚 1 枚の 2 値画像では文字列上の文字が欠落している場合でも、文字列統合画像において全ての文字列上の文字を抽出する文字列領域候補が存在することになる。

このようにして作成された文字列統合画像から文字列領域を決定する。まず文字列統合画像の水平方向及び垂直方向における文字列領域候補の外接方形の量を示すヒストグラムを作成する。

この 2 つのヒストグラムにおいて、まず画像内に存在する文字列が縦一列の文字列かまたは横一列の文字列かを決定する。水平方向及び垂直方向それぞれのヒストグラムにおいて最大値を探索し、比較することで縦一列か横一列の文字列か決定する。文字列が存在する範囲において文字列領域候補を囲む方形は垂直または水平方向に並んで存在し、その範囲内において 2 つのヒストグラムにおける最大値を示すため、2 つのヒストグラムにおいて最大値の存在するヒストグラムの方向に文字列が存在する。仮に垂直方向のヒストグラムにおける最大値の方が水平方向のヒストグラムにおける最大値よりも大きい場合、その情景画像内に存在する文字列は縦一列の文字列となる。

次に文字列が存在する方向のヒストグラムにおいて文字列領域の存在する範囲を決定する。まず文字列が存在する方向のヒストグラム上において文字列領域候補の外接方形の量が 4 以下の場合、その位置におけるヒストグラム値を 0 にする。

そして仮に垂直方向のヒストグラムを考えた場合、最大値から右方向及び左方向それぞれにおいて最初にヒストグラム値が 0 を示す位置をそれぞれ文字列左端位置及び文字列右端位置として、その範囲内に文字列が存在するとする (図 5.3)。

ヒストグラムから文字列領域を決定する際にヒストグラム値が 4 以下の場合 0 にすることで余計な外接方形を無視することができ、余計な範囲を文字列領域として抽出しなくて済む (図 5.4)。

そしてこの範囲内に存在する文字列領域候補を全て囲む新しい方形を作成し、文字列領域を決定し、文字列領域抽出画像を作成する。

このようにして、情景画像中から文字列領域を決定する。

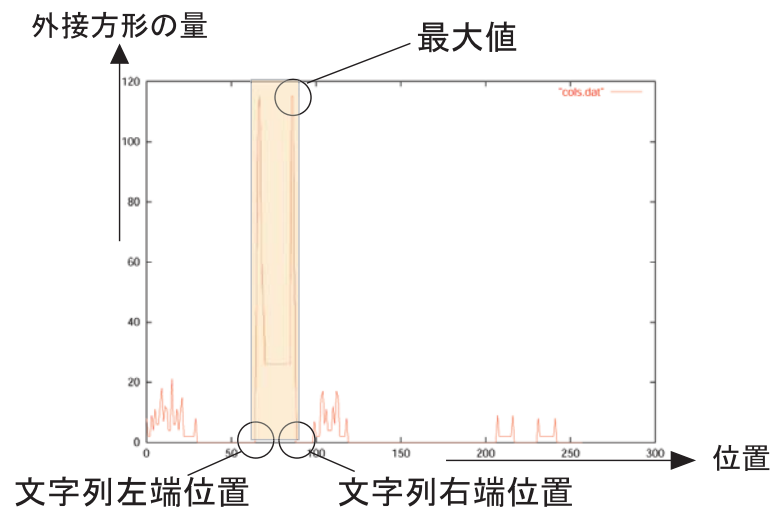


図 5.3: 垂直方向ヒストグラムからの文字列範囲の決定

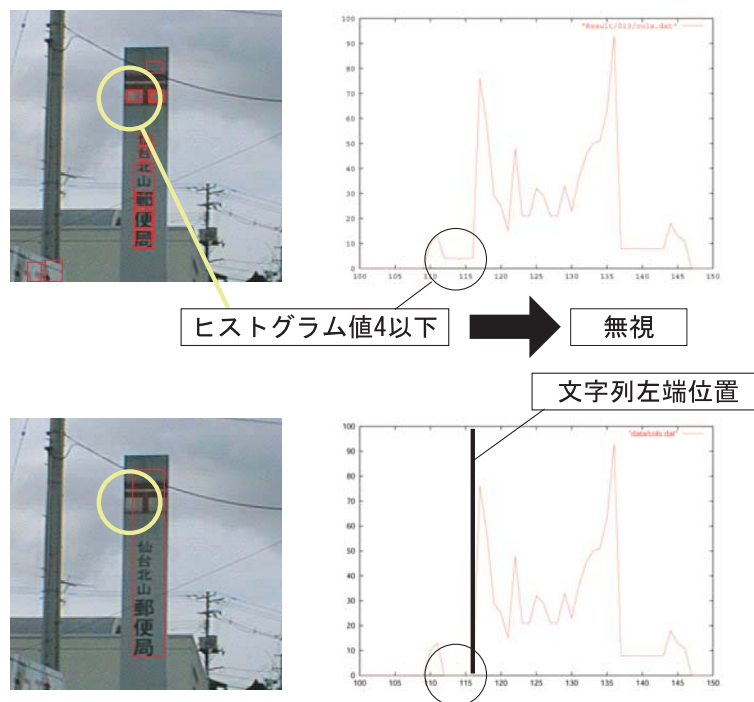


図 5.4: 文字列領域決定

第6章

実験

6.1 実験準備

今回提案した手法に関して、有効性を検証するために実験を行った。実験は種々の状況下において撮影された文字情報を含む情景画像 100 枚に対して行った。実験に利用する情景画像の条件を画像内に存在する文字情報が目視にて読み取れることとし、文字情報の画像内での位置、大きさ、種類などは一切制限しない。

6.1.1 評価基準

抽出精度の評価基準として、文字列統合画像における文字列領域候補と文字列領域抽出画像における文字列領域の 2 種類の抽出程度によって判断する。

(1) 文字列統合画像

文字列統合画像内において、画像内に存在する文字列を成す文字の数を (文字総数)、抽出された文字列領域候補の数を (方形総数)、文字列を成す文字を囲んだ文字列領域候補の数を (文字包囲方形) として、以下の 2 種類の評価基準を利用して文字列統合画像の評価をする。

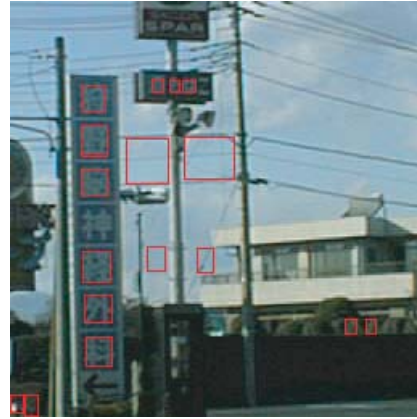
- 再現率 = (文字包囲方形) / (文字総数)
- 適合率 = (文字包囲方形) / (方形総数)

再現率は画像内に存在する文字列を成す文字をどれだけの確率で抽出できたか、また適合率は抽出した文字列領域候補がどれだけの確率で文字を抽出しているかを示す。つまり再現率、適合率共に 100 % のとき最も精度良く文字列を成す文字を抽出することができたことになる。実際に文字列統合画像における再現率及び適合率の計算の仕方の例を図 6.1 に示す。

(2) 文字列領域抽出画像



文字総数 : 6
 方形総数 : 14
 文字包囲方形数 : 6
 再現率 : $6/6 \times 100 = 100\%$
 適合率 : $6/14 \times 100 = 43\%$



文字総数 : 7
 方形総数 : 17
 文字包囲方形数 : 6
 再現率 : $6/7 \times 100 = 86\%$
 適合率 : $6/17 \times 100 = 35\%$

図 6.1: 文字列統合画像における再現率、適合率の計算例

画像内に文字列が一行のみ存在する場合において、その文字列をどれだけ正確に抽出することができているかを目視により次の4種類に分類することで文字列領域抽出画像の評価をする。

- Perfect
 画像内に存在する文字列を正確に抽出している場合。
- Include
 画像内に存在する文字列以外の領域も含まれるが文字列を抽出している場合。
- Part
 画像内に存在する文字列の一部を抽出している場合。
- Miss
 画像内に存在する文字列をまったく抽出できなかった場合、または文字列の一部または全部を含む領域を抽出しているが、余計な領域の方が大きい場合。

文字列領域を抽出した後に文字認識を行うことを考えると、文字列領域が Perfect または Include で抽出された場合に成功、Part または Miss の場合が失敗と考えられる。つまり Perfect 及び Include の画像が多いほど抽出精度が良くなる。実際に文字列領域が Perfect、Include、Part 及び Miss に評価された例を図 6.2 に示す。

これらの評価基準を基に、提案手法と従来手法とを比較する。

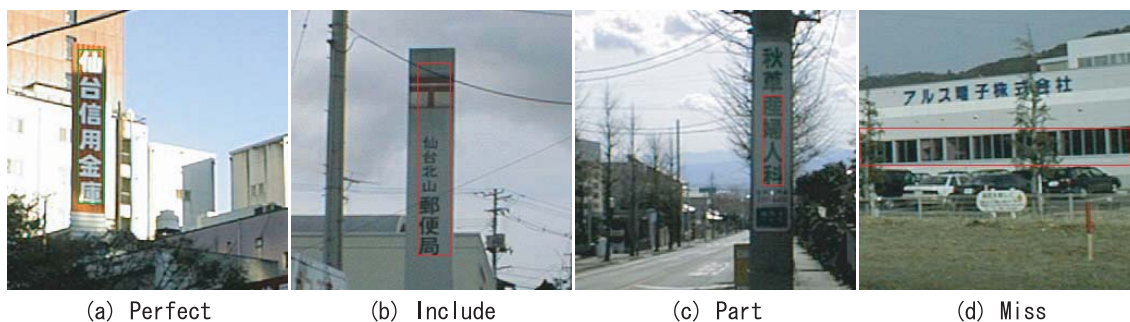


図 6.2: 文字列領域抽出画像における評価例

6.1.2 従来手法

今回、提案手法の有効性を評価するために実験において従来手法との性能比較を行う。従来手法 1 として 1997 年に松尾らによって提案された明度情報を利用して 2 値画像を作成する手法 [3] を、従来手法 2 として 2003 年に Wang らによって提案されたカラー情報を利用して 2 値画像を作成する手法 [5] を比較対象とする。ただし、従来手法 1、2 共に 2 値画像を作成する時点まで従来手法通りの手法を利用し、2 値画像から文字列領域を抽出する手法に関しては提案手法と同じ手法を利用している。

6.2 実験結果

種々の状況下における文字情報を含む情景画像 100 枚に対して実験を行った。文字列統合画像における実験結果は 100 枚の画像中に含まれる 1105 文字に対する結果である (表 6.1)。文字列領域抽出画像における実験結果は 100 枚の画像の内、文字列が一行のみ存在する画像 63 枚の 63 列の文字列に対する結果である (表 6.2)。

これらの評価基準を利用して提案手法と 2 つの従来手法の実験結果を比較する。

表 6.1: 実験結果 (文字列統合画像)

	提案手法	従来手法 1	従来手法 2
再現率	69.8 %	52.4 %	62.1 %
適合率	53.7 %	34.9 %	37.6 %

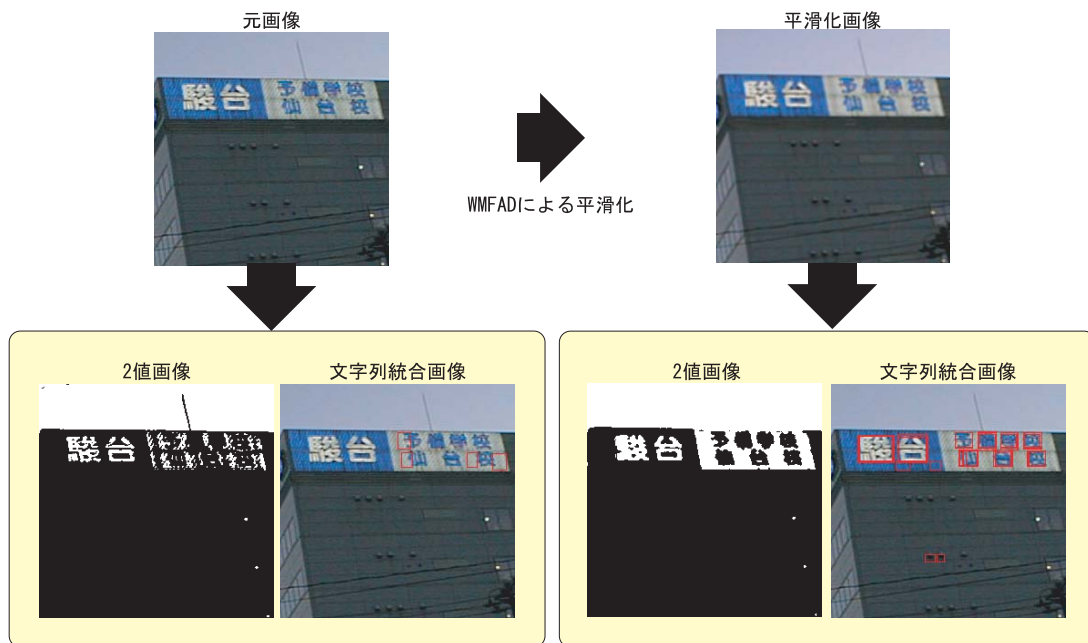
表 6.2: 実験結果 (文字列領域抽出画像)

	提案手法	従来手法 1	従来手法 2
Perfect	24 列	7 列	19 列
Include	15 列	3 列	13 列
Part	7 列	9 列	12 列
Miss	17 列	44 列	19 列

種々の情景画像 100 枚に対して行った実験において以下のような結果が得られた。

(1) 文字列を成す領域が照明の影響により不安定な RGB 値を保持している画像 (図 6.3 参照)

図 6.3 のように文字列領域が不安定な RGB 値を保持しているような画像では、従来手法を利用して文字と背景の部分とを良好に分解する 2 値画像を作成することが困難となり、文字列を成す文字を文字列領域候補として良好に抽出することができない。しかし提案手法では前処理として WMFAD により局所的なエッジ部分を残したまま余計なノイズを除去する平滑化を行う。この平滑化によって画像内で同じオブジェクトを表す画素が似たような RGB 値を保持し、かつ余計なノイズを除去した平滑化画像を作成することができる。この平滑化画像を利用して 2 値化を行うため、文字とそれ以外の部分とを良好に分解する 2 値画像を作成ことができ、結果として文字列を成す文字を文字列領域候補として良好に抽出することができる。



(a) 元画像から従来手法1による2値化及び文字列領域候補抽出 (b) 平滑化画像から提案手法により2値化及び文字列領域候補抽出

図 6.3: 文字列を成す領域が照明の影響により不安定な画像

(2) 文字列を成す領域が照明の影響により似たような明度値を保持している画像 (図 6.4 参照)

図 6.4 のように文字列領域に陰影が掛かっているために、文字とその背景部分との明度値が似ているような画像では、明度値を利用して 2 値画像を作成する従来手法 [3] では文字とそれ以外の部分とを良好に分解する閾値を選択することが困難となり、文字列を成す文字を文字列領域候補として良好に抽出することができない。しかし提案手法ではカラー情報を利用して $L^*a^*b^*$ 空間上でのカラークラスタリングを行いクラスタリング画像を作成する。これにより明度値の似ている領域も $L^*a^*b^*$ 空間上で異なる代表色へと変換することができ、この代表色を利用してオブジェクトクラスタリングを行うことで文字とその背景部分との明度値が似ている領域を良好に分解する 2 値画像を作成することができ、結果として文字列を成す文字を文字列領域候補として良好に抽出することができる。

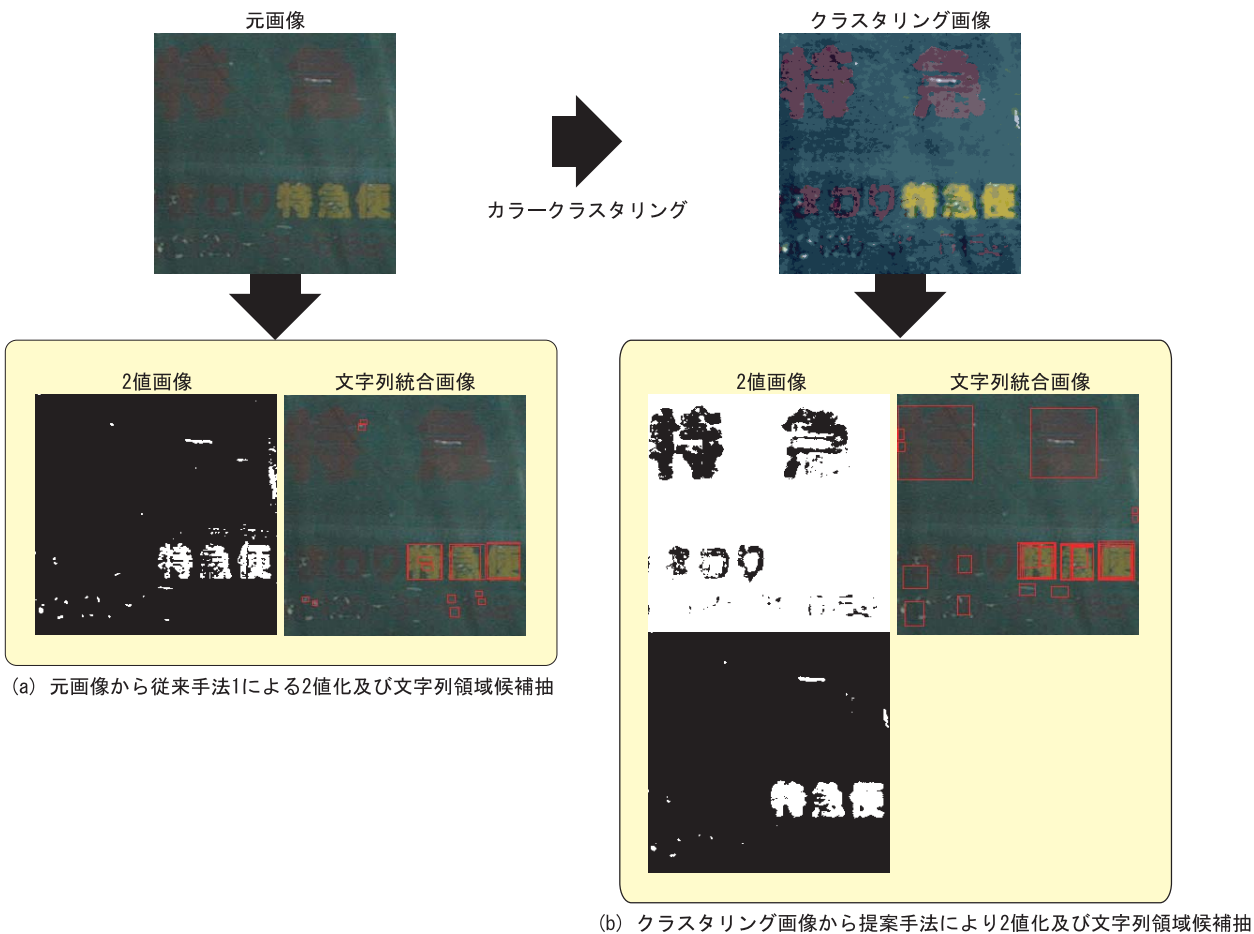


図 6.4: 文字列を成す領域が照明の影響により似たような明度値を保持している画像

(3) 文字列を成す領域が画像内において小さい画像 (図 6.5 参照)

図 6.5 のように画像内において文字列領域が小さい場合、従来手法 [3] のように閾値を変化させて作成した 2 値画像の画像全体での平均隣接数を評価基準として文字とそれ以外の部分とを良好に分解する閾値を選択する手法では、平均隣接数に明確な変化が現れないため、良好な 2 値画像を作成することが困難となる。また従来手法 2 [5] のように画像全体の正規化 RGB

空間上でのヒストグラムからオブジェクトの分割位置を決定する手法では、文字とそれ以外の部分とを良好に分解する分割位置に明確な特徴が現れず、良好な2値画像を作成することが困難となる。しかし提案手法では平滑化画像に対し、 $L^*a^*b^*$ 空間上で周囲26個のCubeよりもCube内に存在する画素数の多いCubeを代表色としてカラークラスタリングを行う。そのため図6.5のように画像内に存在する文字列領域が小さい場合でも文字列部分を表すCubeが代表色として選択されるため、その代表色を利用してオブジェクトクラスタリングを行うと文字とそれ以外の部分とを良好に分解する2値画像を作成することができ、結果として文字列を成す文字を文字列領域候補として良好に抽出することができる。

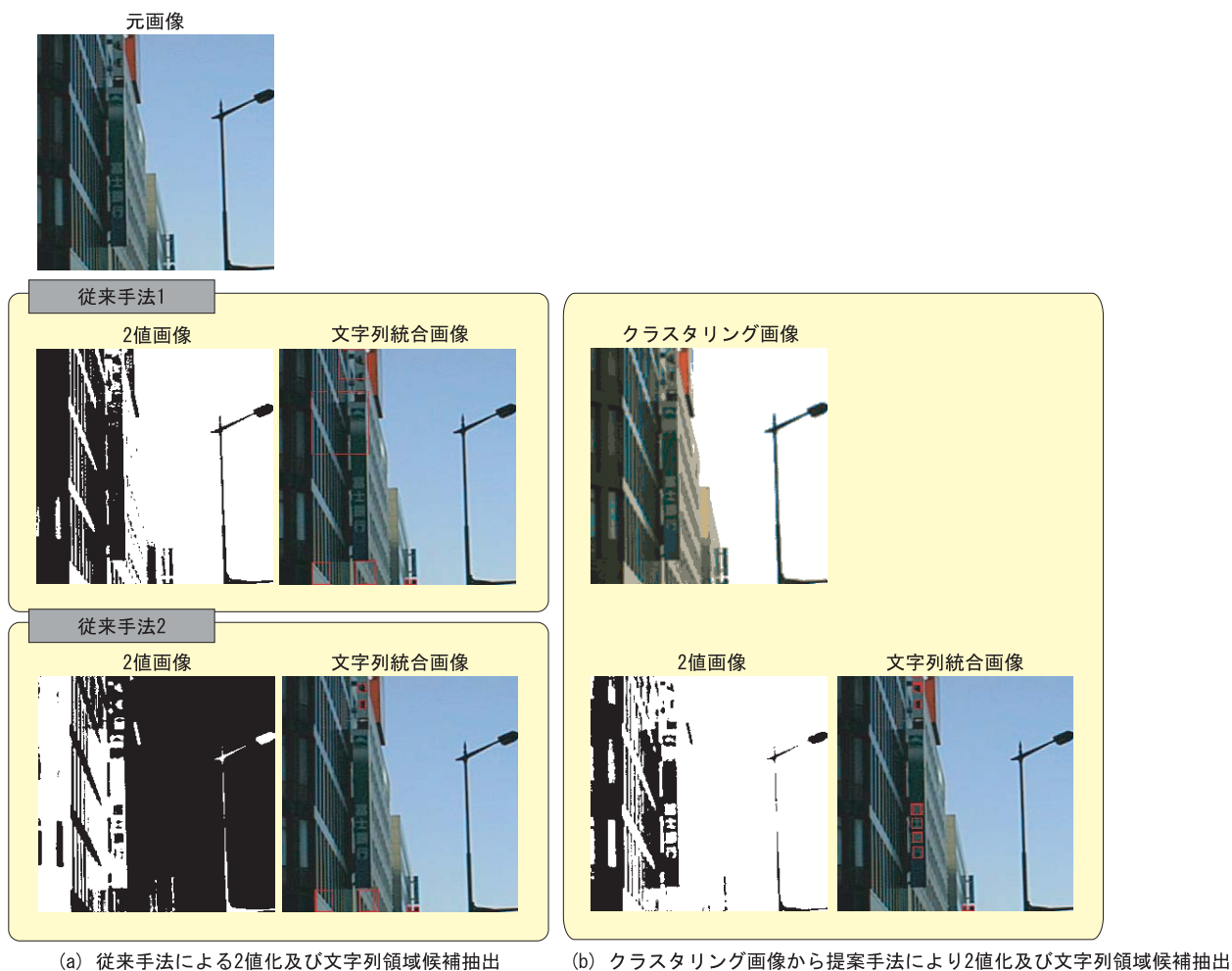


図 6.5: 文字列を成す領域が画像内において小さい画像

そして文字列を成す文字を良好に抽出することができた結果、文字列領域も正確に抽出することができる(図6.6参照)。従来手法1では文字列を成す文字を正確に抽出できなかったため、文字列領域以外の領域を文字列領域として抽出している。また従来手法2では偶然、文字列の上部と下部に文字列を成す文字以外の文字列領域候補が存在したため、文字列領域を抽出しているかのように見える。しかし、これは偶然でしかないので正確に抽出できたとは言えない。しかし提案手法では文字列の上部に余計な文字列領域を抽出してしまっているた

め評価としては Include ではあるが、良好に文字列を成す文字を抽出したため、良好に文字列領域を抽出している。

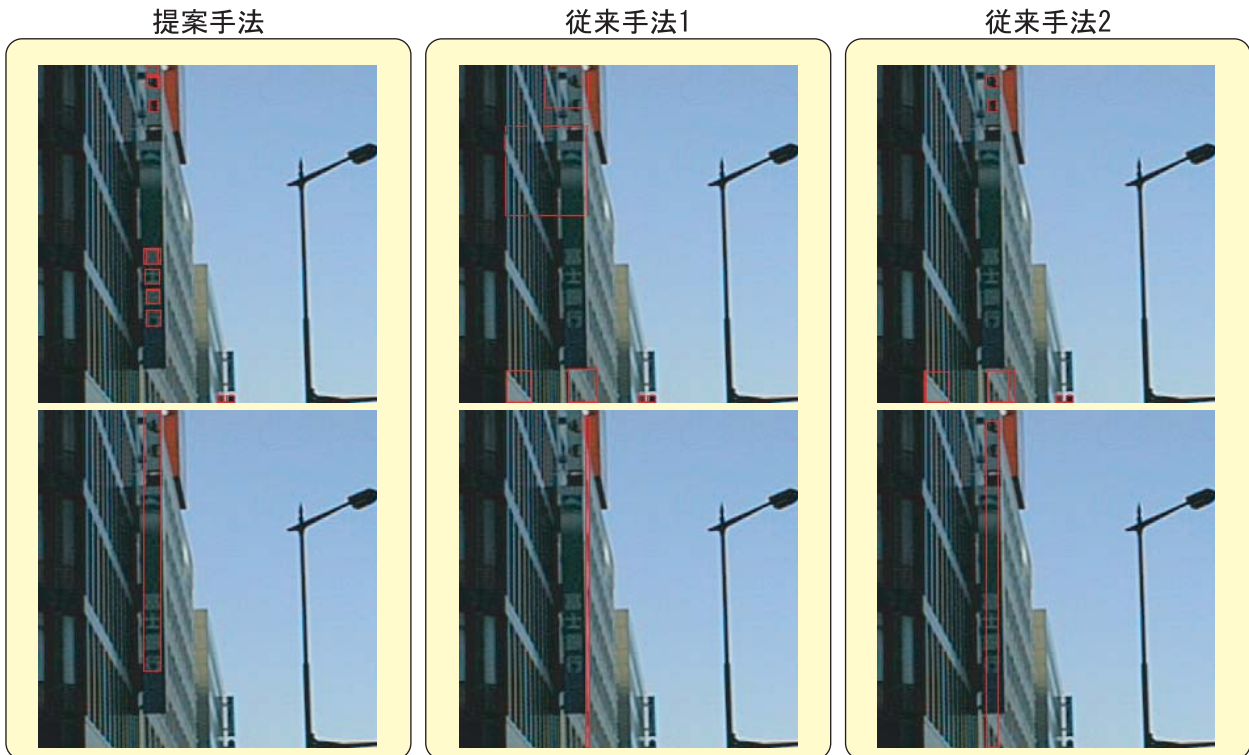


図 6.6: 文字列を成す領域が画像内において小さい画像の文字列領域抽出

(4) 文字列を成す文字同士が接近している画像 (図 6.7 参照)

図 6.7 のように文字列を成す文字同士が接近しており、かつ文字列を成す領域が小さい場合、従来手法 1[3] のように閾値を変化させて作成した 2 値画像の画像全体での平均隣接数を評価基準として文字とそれ以外の部分とを良好に分解する閾値を選択する手法では、平均隣接数に明確な変化が現れなかったり、現れたとしても文字と文字の間の部分を除いた閾値を選択してしまうため、良好な 2 値画像を作成することが困難となる。また従来手法 1[5] のように画像全体の正規化 RGB 空間上でのヒストグラムからオブジェクトの分割位置を決定する手法では、文字とそれ以外の部分とを良好に分解する分割位置に明確な特徴が現れなかったり、現れたとしても文字領域をある程度含んだ分割位置を選択してしまうため、文字領域を明確な閉領域として 2 値化することが困難となり、文字列を成す文字を文字列領域候補として抽出することができない。しかし提案手法では画像を 4×4 の小領域に分割し、各小領域において 4 つの参照領域から 4 個の領域閾値を決定し、その領域閾値を利用して各画素に画素閾値を決定する。これにより文字と文字の間の領域も明確に 2 値化することができ、文字とそれ以外の部分とを良好に分解する 2 値画像を作成することができ、結果として文字列を成す文字を文字列領域候補として良好に抽出することができる。

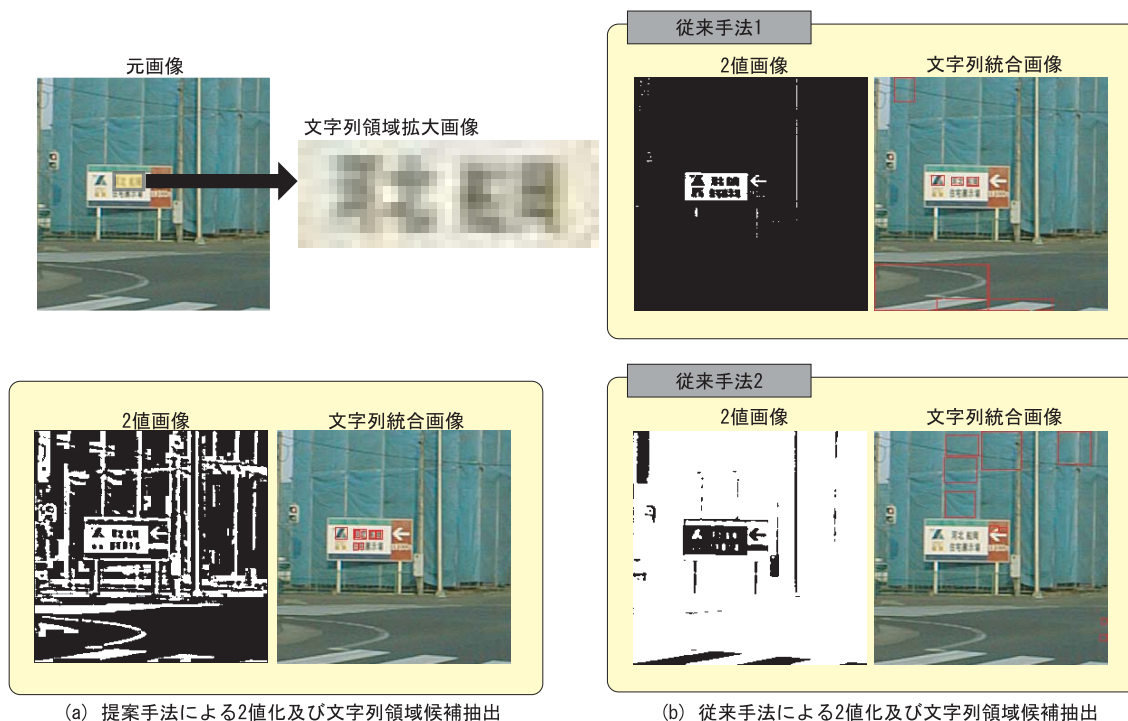


図 6.7: 文字列を成す領域が画像内において小さい画像

(5) 文字列を成す領域に陰影が掛かっている画像 (図 6.8 参照)

図 6.8 のように照明の影響によって文字列に陰影が掛かっており、文字列を成す文字の上部と下部とで明度値や色の特徴が異なるような画像において、従来手法 1[3] のように明度情報を利用して 2 値画像を作成する手法でも、照明の当たっている背景部分と照明の当たっていない文字部分とが似たような明度値を保持するため、文字とそれ以外の部分とを良好に分解する 2 値画像を作成することができない。また従来手法 2[5] のように正規化 RGB 空間上でのヒストグラムからオブジェクトの分割位置を決定する手法では、照明の当たっている背景部分と照明の当たっていない文字部分とを同じオブジェクトとして分割してしまい、文字とそれ以外の部分とを良好に分解する 2 値画像を作成することができない。しかし提案手法では画像を 4×4 の小領域に分割し、各小領域において 4 つの参照領域から 4 個の領域閾値を決定し、さらに各小領域に所属する各画素において平均色を利用して領域閾値に重みを付加し 2 値化する。これにより照明の当たっている領域と当たっていない領域とで異なる閾値を各画素が保有することが可能となる。また平均色を利用して 4 個の領域閾値に重みを付加するため、照明の当たっている画素では、照明の当たっている参照領域の閾値に大きな重みを付加し、当たっていない画素では当たっていない領域の閾値に大きな重みを付加することが可能となる。これにより文字とそれ以外の部分とを良好に分解する 2 値画像を作成ことができ、結果として文字列を成す文字を文字列領域候補として良好に抽出することができる。そして文字列を成す文字を良好に抽出することができた結果、文字列領域も正確に抽出することができる (図 6.9 参照)。実際、2 種類の従来手法では文字列を成す文字を正確に抽出することができなかつたため、文字列領域の一部のみ抽出されている。このように文字列領域の

一部しか抽出できなかった場合、文字列領域に文字認識をかけたとしても文字を理解することができないため、実質失敗と言える。しかし、提案手法では良好に文字列を成す文字を抽出したため、正確に文字列領域を抽出することができている。

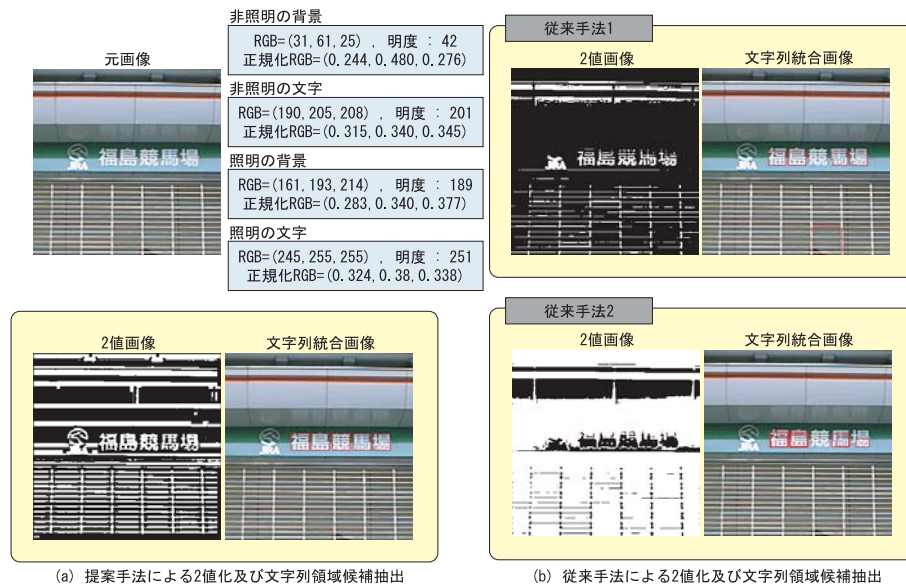


図 6.8: 文字列を成す領域が照明の影響により文字の上部と下部とで特徴の異なる画像

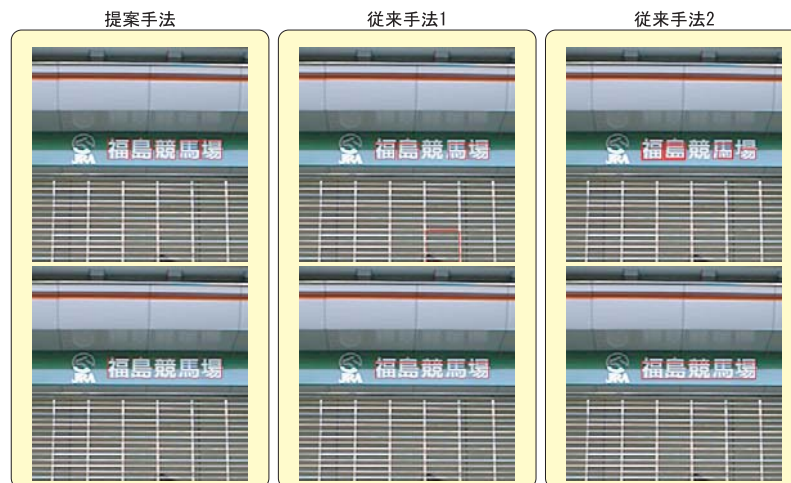


図 6.9: 文字列を成す領域が照明の影響により文字の上部と下部とで特徴の異なる画像の文字列領域抽出

(6) 曲線的に文字が配置されている画像

図 6.10 のように文字が曲線的に配置されている画像において、提案手法を利用して 2 値化を行うと図 6.10(a) のように文字とそれ以外の部分とを良好に分解した 2 値画像を作成することができる。また 2 値画像内から文字領域候補を抽出すると図 6.10(b) のように文字を文字領域候補として文字の領域を抽出することができる。しかし、文字列領域候補を抽出する際に文字列を成す文字の連結性を利用して抽出を行うため、水平・垂直方向へ直線的に並んでいない文字を抽出することができず、文字列統合画像において文字を抽出することができなかった。



図 6.10: 曲線的に文字が配置されている

これは文字列領域候補を抽出する段階において、曲線的に文字が配置されている文字列をも考慮した手法にすれば、このような文字列も抽出可能であると考えられる。

(7) 四角形の領域が並んでいる画像

図 6.11 のように窓などの四角形領域が並んでいる画像において、提案手法を利用して 2 値化を行うと図 6.11(a) のように文字とそれ以外の部分とを良好に分解した 2 値画像を作成することができる。しかし、同様に図 6.11(b) のように窓などの四角形領域を分解した 2 値画像も作成されてしまう。そのため文字列統合画像 (図 6.11(c) 参照) において文字以外の領域も多く文字列領域候補として抽出されてしまう。これにより文字列領域候補方形数のヒストグラム上において最大値を含む山の範囲が文字以外を囲む範囲になり、文字列領域抽出画像 (図 6.11(d) 参照) において文字列以外の領域を抽出してしまった。

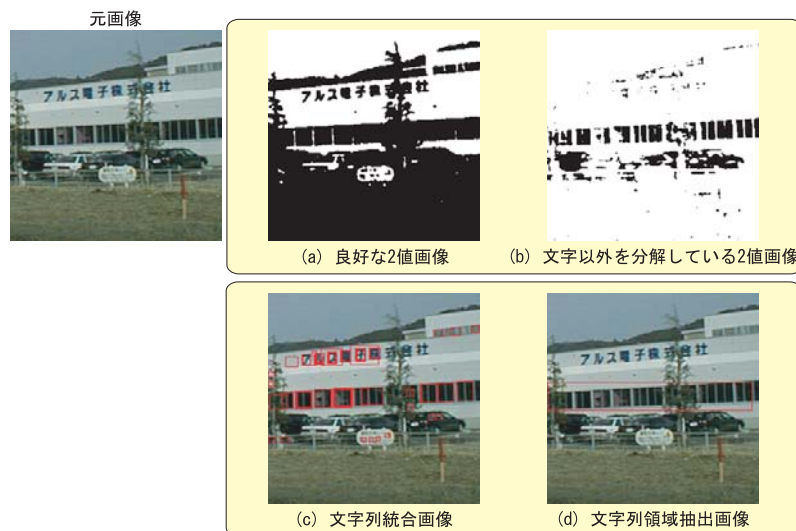


図 6.11: 四角形の領域が並んでいる画像

これは文字列領域候補を抽出する際に方形内の文字らしさを測る尺度を利用して文字以外を抽出している方形を排除する手法 [5] などを利用することで解決できる問題であると考えられる。

6.3 実験まとめ

提案手法における前処理において、WMFAD を利用して局所的なエッジ部分を残したまま余計なノイズを除去する平滑化を行ったことで、情景画像の文字列部分が不安定な RGB 値を保持していても良好な 2 値画像を作成することができた。

提案手法におけるカラー情報を利用した 2 値画像作成において、 $L^*a^*b^*$ 空間上でカラークラスタリングを行い、代表色を選択し 2 値画像を作成したため、文字とその背景部分とで明度値が似ている場合でも良好な 2 値画像を作成することができた。またカラークラスタリングにおいて周囲 26 個の Cube よりも Cube 内の画素数が多い Cube を代表色として選択するため、文字列領域の小さい画像においても良好な 2 値画像を作成することができた。

提案手法における明度情報を利用した 2 値画像作成において、画像を 4×4 の小領域に分割し、各小領域において複数の領域閾値を決定し 2 値画像を作成したため、文字列領域が小さく、文字同士が接近していても良好な 2 値画像を作成することができた。また文字列部分に陰影が掛かっている場合も、各小領域に所属する各画素に平均色を利用して画素閾値を決定し 2 値画像を作成したため、文字列領域において陰影の掛かっている画素とそうでない画素とで異なる閾値を与えることができ、良好な 2 値画像を作成することができた。

今回、種々の状況下における文字情報を含んだ情景画像 100 枚に対して実験を行った。実験結果から文字列統合画像における評価基準の再現率、適合率共に提案手法において最も高い精度を出すことができた。つまり最も少ない数の文字列領域候補で最も精度良く文字列を成す文字を抽出することができた。また文字列領域抽出画像の評価において Perfect 及び Include の枚数が提案

手法において最も多く、最も高い精度で文字列領域を抽出することができた。これは文字列統合画像における再現率、適合率が提案手法において最も精度が良いことから明らかな結果である。以上のことから提案手法の有効性を示した。

第7章

結論及び今後の課題

7.1 結論

カラー情報を利用して種々の状況下における文字情報を含む情景画像中から文字列を成す文字の領域を抽出する手法を提案した。

その手法は、まず前処理として WMFAD による平滑化を行うことで画像内において同じオブジェクトを表している画素が似たような RGB 値を保持し、かつ情景画像取得までに何らかの原因で混入したノイズを除去した平滑化画像を作成する。

この平滑化を行うことで、入力画像内に存在する文字というオブジェクトを表す画素や看板というオブジェクトを表す画素が、それぞれ似たような RGB 値を保持するような平滑化画像を作成することができる。また入力画像に照明や撮影時の天候などの影響によって混入してしまったゴミ塩ノイズのような画素も平滑化により除去した平滑化画像を作成することができる。

そしてこの平滑化画像に対し、カラー情報と明度情報を利用して文字とそれ以外の部分とを良好に分解する 2 値画像を作成する。

カラー情報を利用した 2 値化では、まず $L^*a^*b^*$ 空間においてカラークラスタリングを行い、視覚的に似ている色を表す画素を同じ代表色へと変換しクラスタリング画像を作成する。そして作成されたクラスタリング画像において正規化 RGB 空間上でのオブジェクトクラスタリングを行うことで、画像内に存在するオブジェクトを良好に分解する 2 値画像を複数枚作成する。そしてオブジェクトクラスタリングによって作成された複数枚の 2 値画像を境界線の長さによる複雑度によって評価し、文字とそれ以外の部分とを良好に分解している 2 値画像を複数枚選択する。このようにしてカラー情報を利用して複数枚の 2 値画像を作成する。

このカラークラスタリングを $L^*a^*b^*$ 空間上で行い、代表色を決定してクラスタリング画像を作成することにより、視覚的に似たような RGB 値を保持する画素が同じ代表色へと変換される。これにより平滑化を行ったことで同じオブジェクトを表す画素が似たような RGB 値を保持している平滑化画像内において、同じオブジェクトを表す画素は同じ代表色へと変換される。またカラー

クラスタリングにおける代表色を周囲 26 個の Cube よりも Cube 内の画素数が多い Cube に設定することにより、画像内に存在するオブジェクトの領域が小さく画素数が少ないようなオブジェクトでも代表色として選択される。さらに $L^*a^*b^*$ 空間内で考慮する空間の大きさと Cube のサイズを固定してカラークラスタリングを行う手法 [7] とは違い、両者とも入力画像によって可変としたため、画像内の全画素が $L^*a^*b^*$ 空間内で狭い分布しか持たない場合でも、代表色を良好に複数個設定することができる。そして選択された代表色を RGB 値に変換し、正規化 RGB 空間上での正規化 R、正規化 G のヒストグラム及び明度値のヒストグラム上でオブジェクトクラスタリングを行うことにより、代表色を直接利用してオブジェクトクラスタリングを行う場合と比較して少ない枚数の 2 値画像を作成でき、良好な 2 値画像を作成することができる。また正規化 R、正規化 G 及び明度値のヒストグラム上で高さが 100 以上のすべての極大値を利用してオブジェクトクラスタリングを行うため、クラスタリング画像内で文字というオブジェクトが複数個の代表色から構成されている場合でも、文字とそれ以外の部分をそれぞれ表している代表色を分離することができ、良好な 2 値画像を作成することができる。さらにこのようにして作成された 2 値画像の境界線による複雑度を評価することで、ほとんど何も 2 値化されていないような 2 値画像やノイズ領域のみ 2 値化されている 2 値画像などを取り除くことができ、最終的に良好な 2 値画像を複数枚作成することができる。

また明度情報を利用した 2 値化では、まず情景画像から明度値を計算し明度情景画像を作成する。次に明度情景画像を小領域へと分割し、各小領域において 4 つの参照領域から領域閾値を求める。そして各小領域に所属する各画素に対して、得られた 4 つの領域閾値に平均色による重みを利用した重み付き平均で画素閾値を求め、1 枚の 2 値画像を作成する。そして作成された 2 値画像から 1pixel の幅の線を除去し余計な画素を無視することにより、文字とそれ以外の部分とを良好に分解する 2 値画像を作成する。

明度情景画像を 4×4 の小領域へと分割し、各小領域ごとに 4 個の参照領域を設定することで小領域における領域閾値を 4 個決定する。これを利用することにより小領域内に存在する複数のオブジェクトの明度値が似ているために画像全体を利用した 2 値化では、分解することのできないオブジェクトも良好に分解することができる。また参照領域から領域閾値を決定する際に、NAT 法 [8] を利用する。この NAT 法は、オブジェクトが同一の明度値ではなく似たような明度値を保持しているという性質により、オブジェクトの明度値付近を閾値として 2 値化を行うと 2 値画像の領域数が増加するという特徴を利用した閾値決定法である。平滑化画像は WMFAD により同じオブジェクトが似たような RGB 値を保持するように作成された画像であるため、参照領域において NAT 法を利用することで、領域内に存在する複数のオブジェクトを良好に分解する領域閾値を決定できる。各小領域に所属する各画素において、注目画素の周囲の平均色と所属している小領域の領域閾値を決定した 4 個の参照領域の平均色との距離の評価をもとに画素閾値を決定する。照明の当たっている画素と当たっていない画素の両方が存在する小領域において、照明の当たっている画素の画素閾値を決定する場合、注目画素の周囲の平均色は照明の当たっている参照領域との距離が近くなるため、照明の当たっている参照領域から決定された領域閾値に大きな重み付けがなされ、大きな画素閾値になる。逆に照明が当たっていない画素に対しては小さな画素閾値になり、照明などの影響を受けずに良好な 2 値化を行うことが可能となる。このようにして各画素に画素閾値を与えて 2 値化を行うと異なるオブジェクトを余計に連結してしまうような画素が現

れる。このような画素を無視するために、画像内において 1pixel の幅の線を成す画素を除去する。このようにして良好な 2 値画像を 1 枚作成する。

このようにして作成された複数枚の 2 値画像において 4 連結による領域分割を行い、各領域に外接方形を与え、その外接方形の形状によって文字領域候補、部分領域候補、漢数字候補、不用の 4 種類に分類する。そして文字領域候補と部分領域候補に関しては、文字を構成する一部の可能性があるため、条件付の 3 種類のマーキング処理を施す。これにより文字が偏と傍の 2 領域からなる文字も 1 つの外接方形で囲むことができる。

次に各 2 値画像で抽出された文字領域候補の中から文字列を成す文字の連結性を利用して文字列領域候補を選択する。

このようにして各 2 値画像において抽出された文字列領域候補を 1 つの画像に統合し、文字列統合画像を作成する。これにより、1 つの 2 値画像では文字列領域候補が欠落している文字も文字列統合画像では抽出されることになる。

この文字列統合画像から垂直及び水平方向において、文字列領域候補の外接方形の重なりを表すヒストグラムを作成し、その中から最大値を取るヒストグラムを選択することで文字列の方向を決定し、さらに最大値を含むヒストグラムの山を選択することで文字列の範囲を決定する。この決定された範囲内に存在する文字列領域候補を全て囲む新しい方形を作成し、文字列領域を決定する。

デジタルカメラで撮影された種々の状況下における文字情報を含むカラー情景画像 100 枚を対象として文字列領域抽出の実験を行った。従来手法と比較して再現率、適合率共に上昇させることができ、また文字列領域の抽出に関しても最も精度よく抽出することができることを確認した。これにより本手法の有効性を実証した。

7.2 今後の課題

本研究では文字列が水平・垂直方向に直線的に並んでいることを仮定して文字列領域候補を抽出している。そのため、曲線的に文字が配置されている文字列や斜めから文字列を撮影したために画像内で水平・垂直方向に並んでいない文字列などを抽出することができなかった。曲線的に配置されている文字は文字領域として抽出されていても、直線的な文字列を仮定した文字列領域候補の抽出において排除される。そのため曲線的に配置された文字列をも考慮した文字列領域候補の抽出が必要である。

また画像内に存在する文字列で、それを構成する文字と文字が非常に接近しておりかつその背景部分の RGB 値が似ているような文字列の場合、文字列領域を良好に抽出することができなかった。これは前処理として平滑化を行った時点で文字と文字の間の領域と文字部分が似たような RGB 値を保持してしまったことが原因である。このような画像においてカラー情報を利用した 2 値化におけるカラークラスタリングを行うと文字と文字の間の領域と文字部分が同じ代表色に変換されてしまう。また明度情報を利用した 2 値化の場合でも、小領域内において文字列とその周囲の背景と文字間画素という 3 つのオブジェクトが存在し、参照領域から決定される領域閾値は文字列と背景を分解する閾値となるため、文字間画素と文字列を良好に分解することができない。このような状況の画像に前処理を行わず 2 値画像を作成すると、文字間画素と文字列を良好に分解す

ることができる。従って、よりエッジ部分を明確に残すことができ、かつ文字間画素を平滑化してしまわないような前処理が必要となる。

今回は前処理として WMFAD という平滑化のみを行った。しかし実際に入力画像は未知であり、かつ照明条件や撮影時の天候によって無限通りの問題を抱えた画像が入力される可能性がある。このような入力画像に対する前処理として、画像をエッジによって領域分割し、各領域でダイナミックレンジを変えることで照明条件の影響を緩和する手法 [10][11] や色空間上での画素の分布から Cast の有無を判断し Cast を取り除くことで画像の視認性を向上させる手法 [12] などがある。2 値画像を利用して文字列領域を決定する場合、2 値画像を作成する前の段階で非常に安定した画像が得られていれば文字列領域の抽出精度は格段に向上できると考える。本研究では前処理として WMFAD のみを利用したが、このような前処理の考慮も必要である。

さらに文字情報を視覚機能を伴ったシステムにおいて利用するためには、文字認識が必要となる。また実際にこれらのシステムに組み込むことを考えると計算時間も実用的な数字を出さなくてはならない。これらの問題は本研究では考慮していないが実際に実用を考えた場合には絶対的に考慮が必要な内容である。

謝辞

本研究を行うにあたり、多大な御指導を頂いたと共に、本研究を行う機会を与えてくださった東北大学大学院工学研究科阿曾弘具教授に深く感謝致します。

また、本研究を行うきっかけを与えてくださったと共に、本研究全般に渡り多大な専門的御意見及び御指導をしてくださった東北大学大学院工学研究科大町真一郎助教授に深く感謝致します。

さらには、本論文をまとめるにあたり貴重な御意見を頂いた東北大学大学院工学研究科川又政征教授、加藤寧教授に深く感謝致します。

最後に日々の研究において多大な御意見と御指導をしてくださった阿曾研究室の方々、また日常生活において私を支えてくださっている家族、友人の方々に深く感謝致します。

参考文献

- [1] 松浦大祐，山内仁，高橋浩光：
“ 特定色判別と領域限定を用いた円形道路標識野抽出，”
電子情報通信学会論文誌 D-II Vol.J85-D-II No.6 pp.1075-1083 June 2002.
- [2] 大谷淳，塩昭夫：
“ 情景画像からの文字のパターン抽出と認識 ”
電子情報通信学会論文誌 D Vol.J71-D No.6 pp.1037-1047 June 1988.
- [3] 松尾賢一，上田勝彦，梅田三千雄：
“ 適応しきい値法を用いた情景画像からの看板文字列領域抽出 ”
電子情報通信学会論文誌 D-II Vol.J80-D-II No.6 pp.1617-1626 June 1997.
- [4] 松尾賢一，上田勝彦，梅田三千雄：
“ 濃淡及び色情報による情景画像からの文字列抽出 ”
信学技報，PRU92-121，Jan 1993
- [5] Kongquian Wang，Jari A. Kangas：
“ Character location in scene images from digital camera ”
Pattern Recognition 36 (2003) 2287-2299
- [6] M.Gabbouj，P.Haavisto，Y.Neuvo：
“ Recent advances in median filtering ”
E.Arikan(Ed.),Communication,Control, and Signal Processing, Vol.II,
Elsevier Science Publishers B, Ankara, Turkey (1990), pp.1080-1094
- [7] Hiroyuki Hase，Masaaki Yoneda，Shogo Tokai，Jien Kato，Ching Y.Suen：
“ Color segmentation for text extraction ”
International Journal on Document Analysis nad Recognition Volume 6,Number 4 pp.271-284

- [8] Hon-Son Don :
“ A noise attribute thresholding method for document image binarization ”
International Journal on Document Analysis and Recognition (2001) 4:131-138
- [9] 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 ”
信学論 (D) Vol.J68-D No.12 pp.2123-2131 , Dec 1985.
- [10] 小林直樹, 斎藤英雄, 中島真人 :
“ 自然画像表示のための高速な局所的コントラスト強調 ”
電子情報通信学会論文誌 D-II Vol.J77-D-II No.3 pp.502-509 1994 年 3 月
- [11] 安藤慎吾, 今井聡子, 中島真人 :
“ カラーデジタル画像における局所的な視認性改善の自動化 ”
電学論 C , Vol.122-C , No.10 , 2002
- [12] Francesca Gasparini , Raimondo Schettini :
“ Color balancing of digital photos using simple image statistics ”
Pattern Recognition 37 (2004) 1201-1217

研究業績

学会発表

平山勝裕，大町真一郎，阿曾弘具

“ カラー情報を用いた情景画像中からの文字列抽出 ”

第2回情報科学技術フォーラム FIT2003，I-030，2003年9月.

平山勝裕，大町真一郎，阿曾弘具

“ 明度情報とカラー情報を用いた情景画像中からの文字列抽出 ”

第3回情報科学技術フォーラム FIT2004，I-030，2004年9月.

平山勝裕，大町真一郎，阿曾弘具

“ 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 ”

電子情報通信学会パターン認識・メディア理解研究会 (PRMU)，2005年3月.(発表予定)