

鋼材に手書きされた文字列の認識
に関する研究

平成 18 年 度

三重大学大学院工学研究科
博士前期課程 情報工学専攻

川 井 隆 史

修士論文

鋼材に手書きされた文字列の認識 に関する研究



平成 18 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

川井 隆史

はじめに

FA(Factory Automation) の普及に伴って工場の生産工程における生産指示や生産管理の手段として文字認識技術が用いられている。鋼材は形状の制約や錆・油等による表面の汚れなどからバーコードなどのタグを貼ることが大半の場合不可能である。このようなケースでは鋼材にチョークやスプレーで識別番号を印字又は手書きし、目視確認するといった非効率的な作業が主流である。これらを自動化し、識別番号を機械認識できれば生産効率の向上や人員の灼熱の悪環境からの開放が期待できる。しかし、鋼材をカメラで撮影した画像にはインクのむら、照明のむら、背景色のむらが多く文字認識することは非常に難しい。

画像から文字列の 2 値画像を得るためには、まず文字列の位置を特定する必要がある。本研究では、画像を分割して局所領域ごとに 2 値化しきい値を求めて一旦 2 値化、ラベリングを行い、連結成分の形状などから雑音を除去した後、膨張処理を施して文字と文字を連結させ、連結成分の形状から文字列の位置を特定する方法で切り出しを行う。入力画像は文字部分の面積に対して背景部分の面積が大きく、非常にかたよったヒストグラムとなる。そのため、画像 2 値化のための自動しきい値決定法として広く使われている判別分析法のみでは良好な 2 画像が得られない。

本研究では、ヒストグラムの変数変換と、ヒストグラム参照部分の限定を行い、判別分析法に適したヒストグラムに変換した上で判別分析法を適用する画像 2 値化と、それを用いた文字列の切り出しを行う。切り出した画像は 1 画像につき 1 文字列であるので、これを再度 2 値化し収縮膨張やラベリング等各種画像処理を施し、文字の整形、雑音除去を行い文字列の 2 値画像を得る。得られた 2 値画像を目視確認した結果、文字のにじみやかすれ等が少なく、雑音も少ないことから提案手法が有効であることが確認できた。また、64 次元の加重方向指数ヒストグラム特徴と擬似ベイズ識別関数を用いる文字認識実験においても提案手法の有効性が確認できた。

目次

はじめに	i
第 1 章 研究の背景	1
1.1 研究の背景	1
1.2 関連する研究	1
第 2 章 2 値化手法 [6]	3
2.1 多値画像の 2 値化	3
2.2 しきい値選択の手法	4
第 3 章 提案手法	8
3.1 入力画像	9
3.2 文字列の切り出し	10
3.3 文字列画像から文字切り出しと認識	19
第 4 章 実験	28
4.1 文字列切り出し	28
4.2 2 値化・認識実験	29
第 5 章 むすび	31
5.1 まとめ	31
5.2 今後の課題	31
謝辞	33

第 1 章

研究の背景

1.1 研究の背景

鋼材の加工工程において鋼材の異材混入事故防止や品質管理のために鋼材にマーキングやタグの張り付け等を施している。タグとして IC チップやラベルを張り付けるものなどがあり、マーキングとしてスプレー等による文字の印字やバーコード等の記号の印字などがある。

一部の品種でバーコード等の記号の印字による品質管理が普及しつつあるが、大半の鋼材は形状の制約や錆・油等による表面の汚れからバーコードを貼ることが不可能である。このようなケースでは鋼材へのラベルの貼り付けや刻印、チョークやスプレーによる手書きなどで現品をマークし、これを目視確認する、といった非効率な作業が主流である [1]。この作業を機械で自動化できれば人件費削減につながり、人員を灼熱の悪環境から開放することで安全面や健康面でも利点がある。

しかし、このうちラベルの張り付けや刻印は均一な文字や記号のマークが期待できるのに対し、チョークやスプレーによる手書きでのマークは、書き手の違いによる文字の特徴の変化、文字の濃淡のむら、年月の経過による文字のかすれがあり、均一な文字や記号が期待できない。さらに、マークされた文字を画像化する際には照明のむら、背景色のむらがあり、画像二値化や文字の切り出し、認識が困難である。

1.2 関連する研究

本研究に関連する研究として、画像 2 値化に基づいたナンバープレート認識 [2] や多値画像でのテンプレートマッチングを用いたナンバープレート認識 [3] などがあるが、画像 1 枚につき 1 文字列を対象としている。また、複数文字列も想定した研究としてコンテナ番号認識 [4] があるが、これは文字列中に特定の文字が出現することを利用したテンプレ

レートマッチングを行っている。

鋼材画像を扱った研究としてラプラシアン・ガウシアンフィルタを用いた画像二値化、文字認識法 [5] などが提案されているが、こちらも 1 画像につき 1 文字列を対象としている。

本研究では複数文字列が写っている鋼材画像からの文字列切り出し、文字認識手法を提案する。

第 2 章

2 値化手法 [6]

2.1 多値画像の 2 値化

多値画像とは、濃淡があり、各画素が一定の範囲の濃度値をもつ画像のことであり、2 値画像とは濃淡がなく、画素の値が 0, 1 の 2 つの値しかとらない画像のことである。多値画像から対象物を切り出し、背景と分離するために 2 値化処理を用いることが多い。画像の 2 値化は、座標 i, j の濃度値を $f(i, j)$ とすると次式のしきい値処理によって行われる。

$$f_t(i, j) = \begin{cases} 1 & (f(i, j) \geq t) \\ 0 & (f(i, j) < t) \end{cases} \quad (2.1)$$

しきい値処理の例を図 2.1 に、多値画像の例を図 2.2 に、しきい値処理によって得られた 2 値画像の例を図 2.3 に示す。

ここで問題となるのはしきい値 t の決め方である。 t を決める方法はしきい値選択といわれ、以下に代表的なしきい値選択の手法を紹介する。

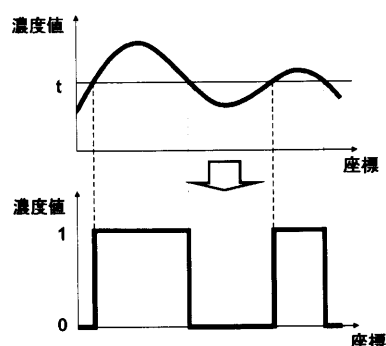


図 2.1: しきい値処理の例

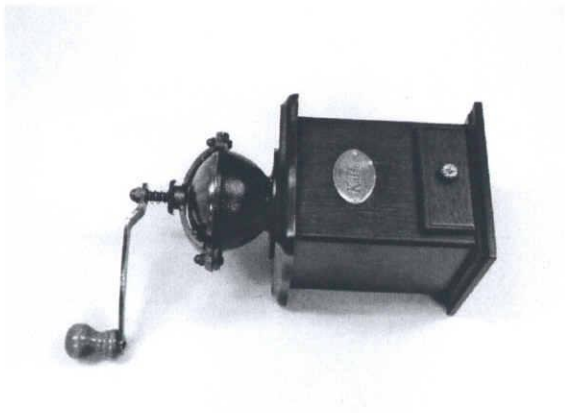


図 2.2: 多値画像の例

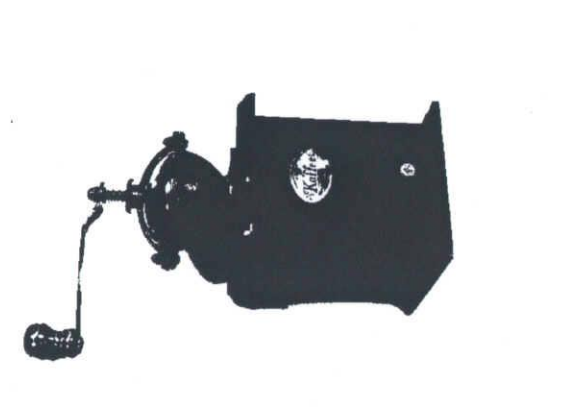


図 2.3: 2 値画像の例

2.2 しきい値選択の手法

2.2.1 モード法

与えられた画像の濃度ヒストグラムが対象図形と背景に対応する 2 つの山の分布になる場合この 2 つの山の間の谷底となる濃度値をしきい値 t とする方法 (図 2.4)。

画像中の対象図形と背景の濃度値の差が大きいときは、ヒストグラムに明確な谷ができるので、この方法が適している。しかしノイズの多い画像ではヒストグラムに明確な谷が生じず、又、ヒストグラムの山が 3 つ以上あるような画像では視覚上望ましいしきい値をどの谷から選べばよいかを決めるのはむずかしく、この方法の適用が困難な場合もある (図 2.5)。

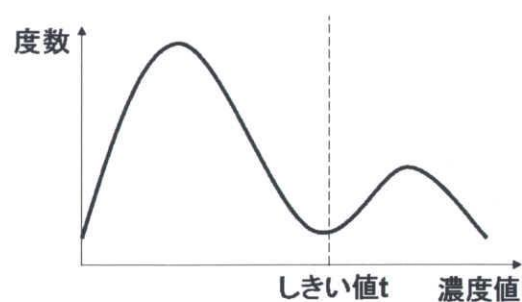


図 2.4: モード法によるしきい値選択

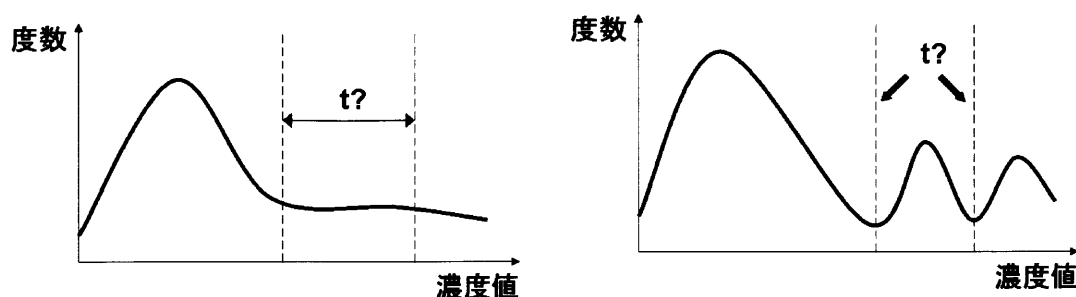


図 2.5: モード法の適用が困難なヒストグラム

2.2.2 p-タイル法

切り出すべき対象図形の面積がほぼ S_0 に等しいものとして、画像全体の面積 S に対する比率を $p = S_0/S$ とする。この場合には、濃度ヒストグラムにおいて濃度値が t 以上になる画素数が全画素数に対して p の割合になるように閾値 t を決める。そのためには、画像の濃度ヒストグラムから、濃度値が高い方からの（あるいは低いほうからの）累積相対度数分布を求め、累積相対度数が p （あるいは $1-p$ ）になる濃度値を求めればよい。

この方法は、図面や文書画像など、切り出すべき対象図形の面積がある程度推定できる場合によく用いられる。

2.2.3 判別分析法

画像の濃度ヒストグラムにおける濃度値の分布を閾値 t で2つのクラス（ t 以上と t 未満）に分割したとき、2つのクラスが最もよく分離するようにパラメータ t を決める方法として、大津の判別分析によるしきい値選定法 [7] がある。分離性の尺度としては、2つのクラスの平均値の分散（クラス間分散）と各クラスの分散（クラス内分散）の比（判別比）が用いられ、この判別比が最大になるようにしきい値 t が選択される。 t の求め方を以下に示す。

与えられた画像が L レベルの濃度値 $(1, 2, \dots, L)$ をもつものとする。ここで、しきい値を t として、 t 以上の濃度値をもつ画素と、それより小さな値を持つ画素の2つのクラスに分け、クラス1、クラス2とする。クラス i ($i = 1, 2$) の画素数を ω_i 、平均濃度値を M_i 、分散を σ_i^2 とおき、全画素数の平均濃度値を M_T とおくと、クラス内分散は

$$\sigma_W^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2} \quad (2.2)$$

クラス間分散は

$$\sigma_B^2 = \frac{\omega_1(M_1 - M_T)^2 + \omega_2(M_2 - M_T)^2}{\omega_1 + \omega_2} = \frac{\omega_1\omega_2(M_1 - M_2)^2}{(\omega_1 + \omega_2)^2} \quad (2.3)$$

で与えられる。さらに、全画素数の濃度値の分散を σ_T^2 とすると、次の関係

$$\sigma_T^2 = \sigma_W^2 + \sigma_B^2 \quad (2.4)$$

が成り立つので、判別比は

$$\frac{\sigma_B^2}{\sigma_W^2} = \frac{\sigma_B^2}{\sigma_T^2 - \sigma_B^2} \quad (2.5)$$

となる。全分散 σ_T^2 はしきい値とは無関係な定数となるので、判別比を最大にするには σ_B^2 を最大にすればよいことがわかる。すなわち、 t を変化させて σ_B^2 を最大にする t の値を求めればよい。

この方法は、ヒストグラムに明確な谷がない場合にもしきい値選択ができるので、広く利用されているが、背景と図形の面積が大きく異なる場合や複雑な画像などにおいて、得られるしきい値が視覚的に望まれるしきい値やモード法によって得られるしきい値からずれることがある。複雑な画像の例を図 2.6 に、図 2.6 に対して判別分析法を用いてしきい値選択し 2 値化した画像を図 2.7 に、しきい値を視覚的に望ましい値に手動で設定して 2 値化した画像を図 2.8 に示す。

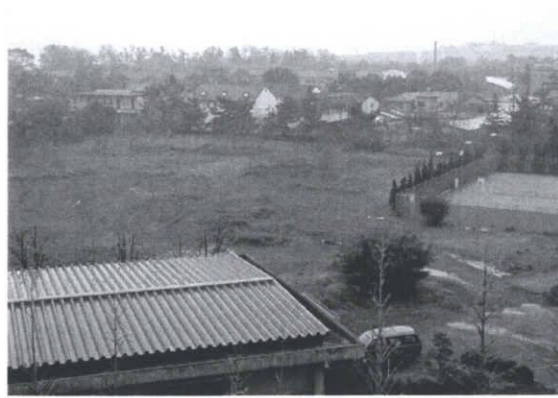


図 2.6: 複雑な画像の例

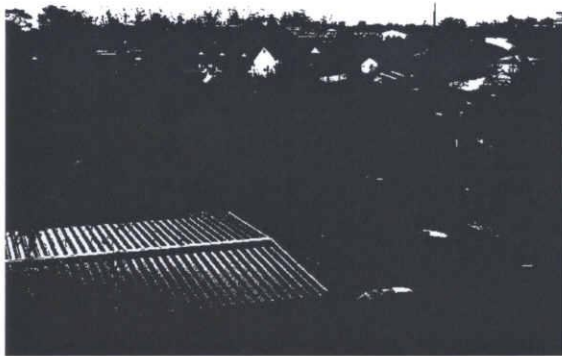


図 2.7: 判別分析法によるしきい値選択



図 2.8: しきい値を手動で設定

第 3 章

提案手法

本研究では入力画像中の文字を文字認識するために、

1. 入力画像から 1 文字列 1 画像となるような文字列画像の切り出しを行い、
2. 文字列画像から文字列を抽出し、文字の切り出しと認識を行う

手法を提案する。提案手法の流れを図 3.1 に示す。ここでは、本研究で扱う入力画像の特徴と、提案手法について述べる。

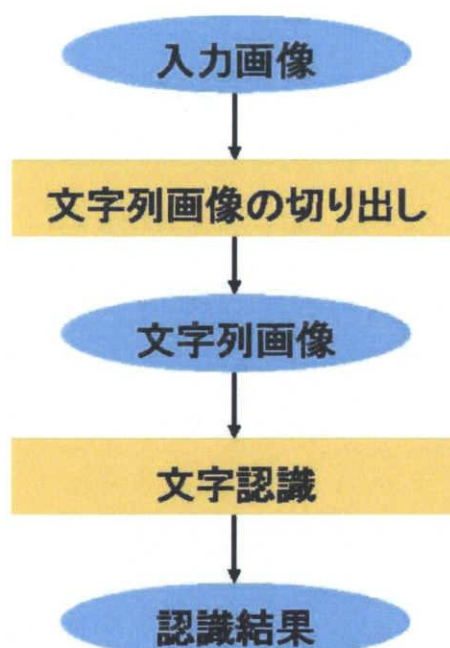


図 3.1: 提案手法の流れ

3.1 入力画像

本研究で扱う入力画像は，鋼材を複数枚重ねたものをカメラで撮影したもので，各鋼材には白い文字で英数字とハイフン「-」からなる記号が書かれている．入力画像の例を図 3.2 に示す．

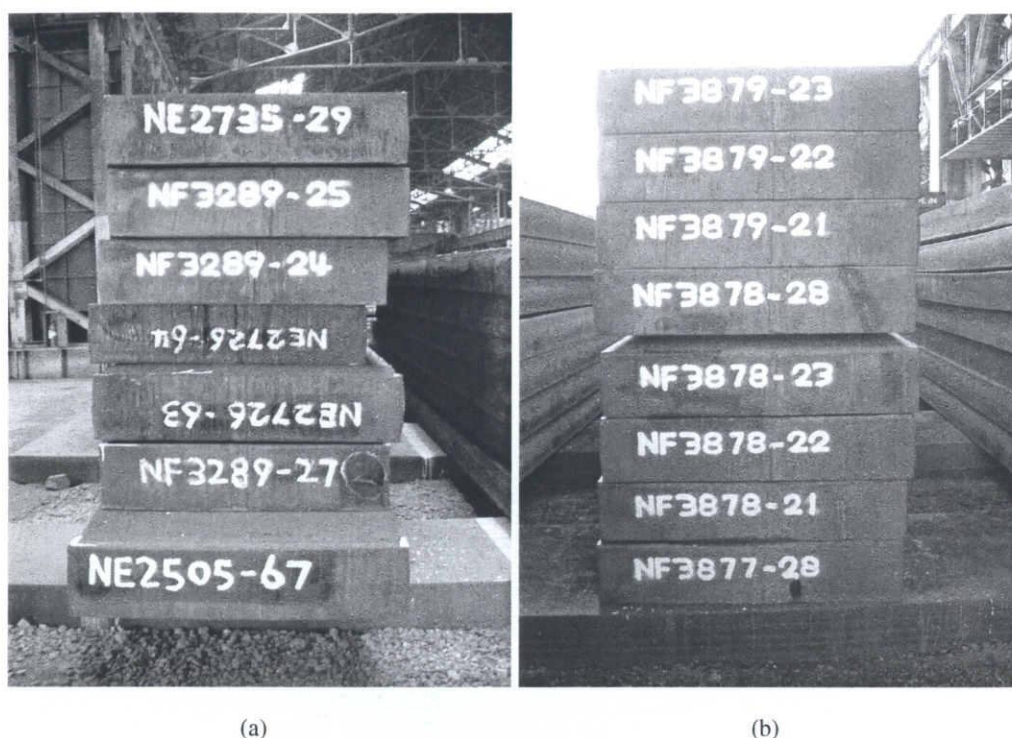


図 3.2: 入力画像の例

画像の特徴として

- 鋼材はほぼ画像の中央付近にある．
- 文字はおおむね同じ太さ，同じような書体である．
- 文字列はほぼ水平に書かれている．

等が挙げられる．一方，

- 重ねられた鋼材は整列していないものもある．
- 重ねられた鋼材の枚数は不定である．
- 鋼材の置かれている環境は様々で屋内のものも屋外のものもある．
- 屋外で撮影されたものには太陽光が差し込んでいるものもあり，また撮影された距離によりフラッシュの写り込み具合が異なり，照明条件は様々である．

- 画像は真正面以外から撮影されたものもある。

等の文字列切り出しを困難にする要素も多く含んでいる。

3.2 文字列の切り出し

3.2.1 領域分割を用いた画像 2 値化

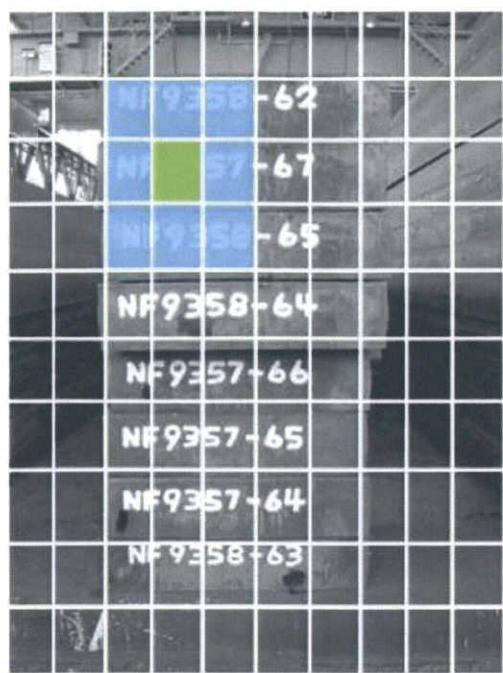
入力画像から文字列画像を切り出すために、文字列領域を抽出する必要がある。入力画像を見ると、文字列は明るい白色で、鋼材は文字に比べて暗い灰色であるので、うまく画像 2 値化のしきい値を設定できれば文字と背景を分離できると考えられる。しかし画像は単一しきい値ではどのしきい値を選択してもうまく 2 値化できない。入力画像を判別分析法で 2 値化した画像例を図 3.3(b) に示す。原因は、単一しきい値では場所ごとに明るさの差が激しい入力画像に対応できないためである。そこで、画像を縦 n 、横 n の $n \times n$ ブロックに分割し、各ブロックで画像 2 値化のためのしきい値を求め、2 値化する。領域分割の例を図 3.3(a) に示す。

図 3.3(a) は 10×10 ブロックに分割した例である。ブロックごとに判別分析法で 2 値化しきい値を求めることで場所ごとに明るさの違う画像でもしきい値選定のミスによる文字のかすれや消失を減らすことができる。領域分割して 2 値化した例を図 3.3(c) に示す。図 3.3(c) を見ると、領域分割を行ったことで文字部分の多くが背景と分離できているのが確認できるが、分割領域の境界部分で 2 値化しきい値にズレが発生し、不自然なブロックノイズが発生してしまう。そこで、図 3.3(a) の緑色部分のしきい値を求める際に、周囲の水色の部分も含めた 3×3 の領域のヒストグラムから求めるように変更する。周囲の色情報も参照することで不自然なブロックノイズを軽減することができる。得られた 2 値画像を図 3.3(d) に示す。単一しきい値の場合と比較して、不自然なブロックノイズを抑えつつ文字部分をきれいに 2 値化できているのが確認できる。しかし、まだ文字部分の 2 値化しきい値が適切でない箇所もあり、また、周囲のノイズも多く存在する。この問題も解消するためにヒストグラム変換を行い 2 値化を行う。このヒストグラム変換については 3.3.1 節で述べる。

3.2.2 フーリエ変換を用いた画像 2 値化

光学的フーリエ変換

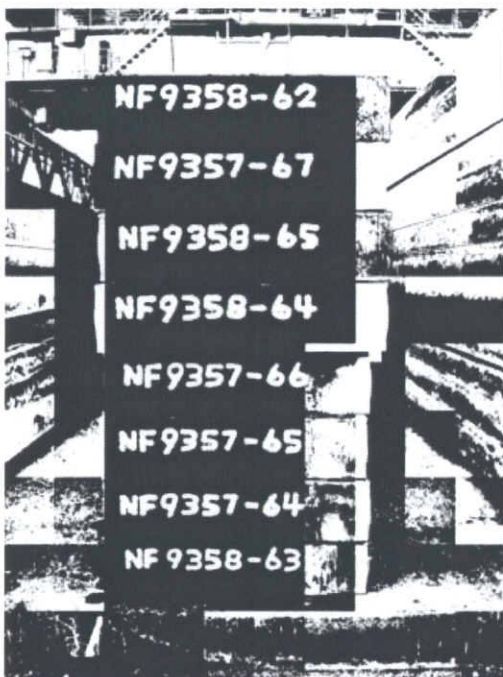
2 次元信号である画像情報に対して、どのような空間周波数成分から構成されているかを調べる方法として 2 次元離散フーリエ変換 (DFT) がある。2 次元離散フーリエ変換対は以下の式 (3.1) で定義される。



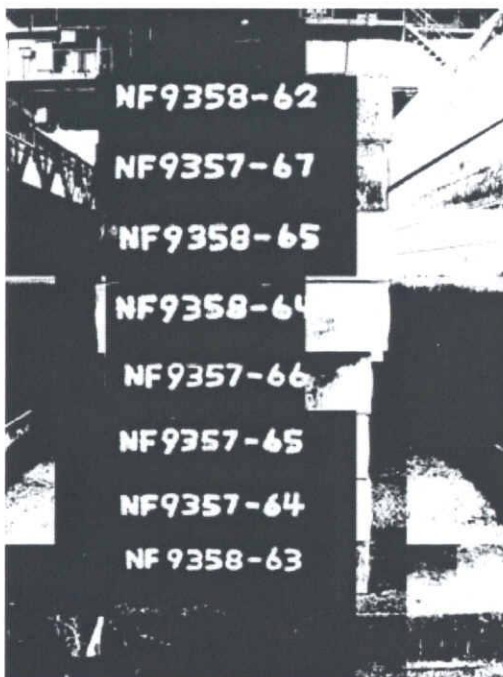
(a) 領域分割の様子



(b) 単一しきい値で2値化



(c) 領域分割して2値化



(d) 領域分割して2値化;周囲の情報も加味

図 3.3: 領域分割と2値化

$$G(k, l) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g(m, n) W_1^{mk} W_2^{nl} \quad (3.1a)$$

$$G(m, n) = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k, l) W_1^{-mk} W_2^{-nl} \quad (3.1b)$$

ただし,

$$W_1 = \exp(-j2\pi/M)$$

$$W_2 = \exp(-j2\pi/N)$$

特に、画像中央を直流成分に対応させるような 2 次元 DFT は、光学的フーリエ変換と呼ばれる。画像 $g(m, n)$ について、画像の大きさ M, N が各々 4 の倍数である場合には、光学的フーリエ変換は以下の式 (3.2) で表せる。

$$G(k, l) = (-1)^{k+l} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (-1)^{m+n} g(m, n) W_1^{mk} W_2^{nl} \quad (3.2)$$

式 (3.2) による 2 次元 DFT において、DFT 結果を格納する 2 次元配列中の周波数成分の配置は図 3.4 のようになる。すなわち、配列中央が直流成分に相当し、配列中央部近辺が低周波数成分、配列の左上・右上・左下・右下の各隅の周辺が高周波数成分に対応する。

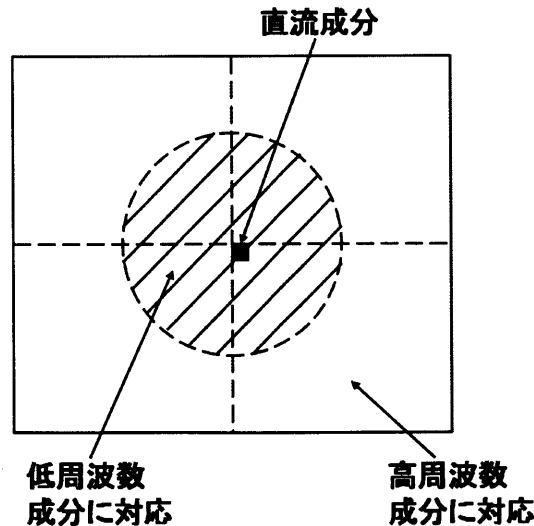


図 3.4: 2 次元 DFT 結果における周波数成分の配置

フィルタ処理

画像に対するフィルタ操作は、一般に画像とフィルタ関数との畳み込み処理として実行できる。一方、2次元DFTを介して空間周波数領域でのフィルタ関数を用いて処理することもできる。例えば、低域通過フィルタや高域通過フィルタに対しては、空間周波数領域において所定の空間周波数成分を一定の規則に従って変更（例えば強制的に0とする）したうえで、2次元フーリエ逆変換を計算することにより、低域通過画像と高域通過画像が得られる。

高域通過フィルタの適用

本研究で扱う画像は、鋼材や背景の輝度値のばらつきがあり、画像2値化のしきい値をうまく設定することは難しい。画像を見ると、鋼材とその背景という大局的な輝度値の変化と、鋼材と文字部分という局所的な輝度値の変化がある。つまり、大局的な輝度値の変化を抑えることができれば、文字列部分を強調させる効果が期待できる。そこで、ごく小さい周波数成分を取り除くために、以下の3種類の高域通過フィルタを空間周波数領域に適用し、画像2値化を行った。

- 直流成分とその4近傍を0とする
- 直流成分とその8近傍を0とする
- 直流成分から4近傍での距離2以内を0とする

高域通過フィルタの適用結果

入力画像及び入力画像に高域通過フィルタを適用した画像に対して画像2値化を行い、得られた2値画像の文字列が判読可能かどうかを目視評価し、高域通過フィルタの効果を比較した。入力画像は55枚で、文字列総数は365である。実験結果を表3.1に、得られた画像例を図3.5、3.6に示す。図3.5は、中央付近に位置している文字列部分の2値化結果について高域通過フィルタを適用した場合の方が文字列部分は多くノイズは少なく2値化されているので、期待した結果が得られたといえる。一方図3.6では、高域通過フィルタを適用した方が文字列部分の2値化精度が悪く、期待した結果が得られなかった。全体としては図3.5のような結果よりも図3.6のような結果の方が多く、表3.1に示したように高域通過フィルタなしが一番2値化成功率が高い結果となってしまった。

原因として、鋼材の大きさや位置は画像ごとにばらつきがあり、その違いに対応できて

いないことが考えられる。高域通過フィルタで文字列部分の強調を行うには、画像ごとに高域通過フィルタの特性を変化させるか、どの画像にも期待した結果が得られるようなフィルタを考案する必要がある。

以上の結果から以下の実験では 3.2.1 節の手法を用いる。

	2 値化成功数	成功率 (%)
高域通過フィルタなし	334	91.5%
4 近傍を 0	291	79.7%
8 近傍を 0	298	81.6%
4 近傍距離 2 以内を 0	293	80.3%

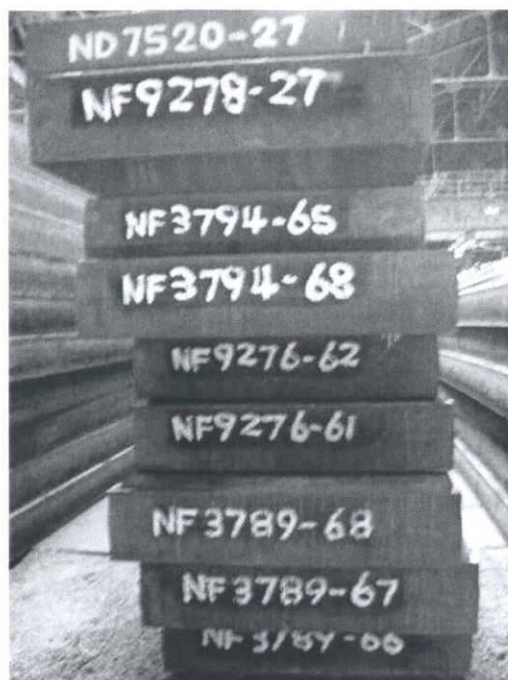
表 3.1: 高域フィルタの適用結果

3.2.3 ノイズ除去

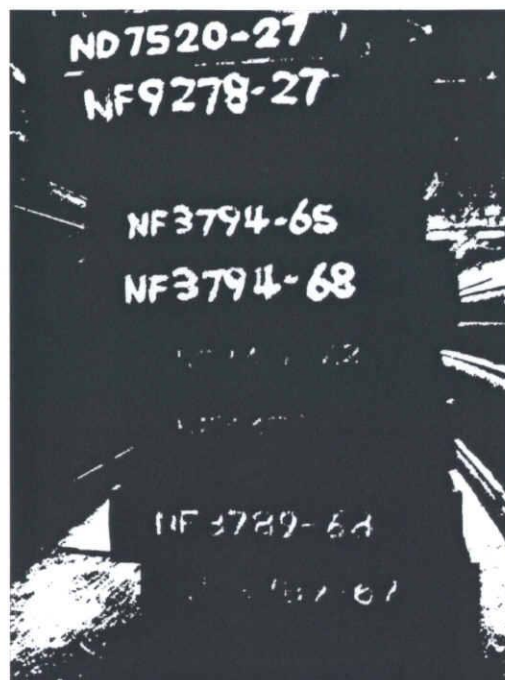
得られた 2 値画像にラベリングを施し、ノイズ除去を行う。ラベリングされた画像を図 3.7(a) に示す。図 3.7(a) では連結成分の外接矩形を白い長方形で表示している。ラベリングされた画像を見ると、文字部分の外接矩形は正方形に近く、もっとも縦長のものは数字 '1' でもっとも横長のものはハイフン '-' である。さらに文字部分は外接矩形の面積に対して連結成分の画素数が多い特徴がある。これらの特徴から文字部分以外をノイズとして除去することを目指す。まず、文字部分の連結成分の画素数や外接矩形の大きさは一定以内であると考えられるので、極端に画素数が大きい連結成分や極端に画素数が小さい連結成分、外接矩形の縦もしくは横が一定値より長い連結成分を除去する。(図 3.7(b)) 次に、連結成分の外接矩形の縦横比が極端な連結成分を除去する。(図 3.7(c)) ここで横長の線状の連結成分と縦長の線状の連結成分を除去できる。そして、連結成分の外接矩形の面積に対して外接矩形の画素数が少ない連結成分を除去する。(図 3.7(d)) ここでは斜めに走る線状の連結成分を除去できる。

次に、ノイズ除去を行った画像を水平方向に膨張させる。膨張させた画像を図 3.8(a) に示す。文字列はほぼ水平に書かれているので、横方向に膨張させることで文字と文字が結合し、文字列がひとつの連結成分に結合される。ここで、膨張は水平方向にのみ行うことで、上下に隣接する文字とは結合しないようにしている。この処理で文字列部分に対応する連結成分の外接矩形は横長の長方形となる。よって外接矩形が横長でない連結成分をノイズとして除去する。ノイズ除去を行った画像を図 3.8(b) に示す。

図 3.8(b) を見ると文字列部分に対応する連結成分以外のノイズの多くは除去できていることが確認できる。



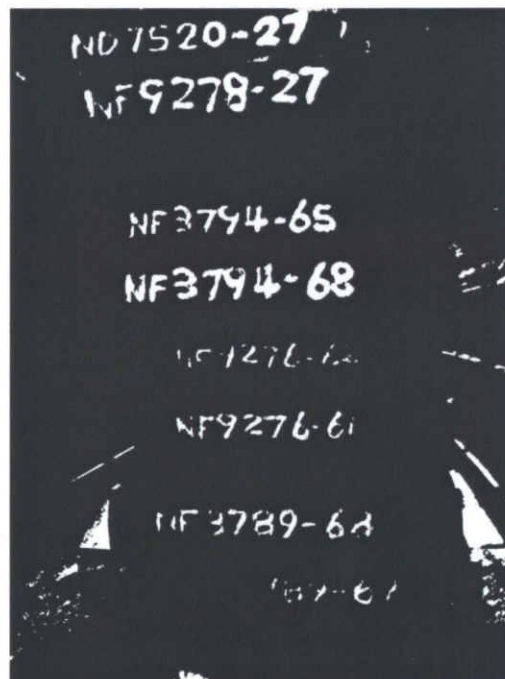
(a) 原画像



(b) 3.5(a) を 2 値化した画像



(c) 3.5(a) に高域通過フィルタ 3 を適用した画像

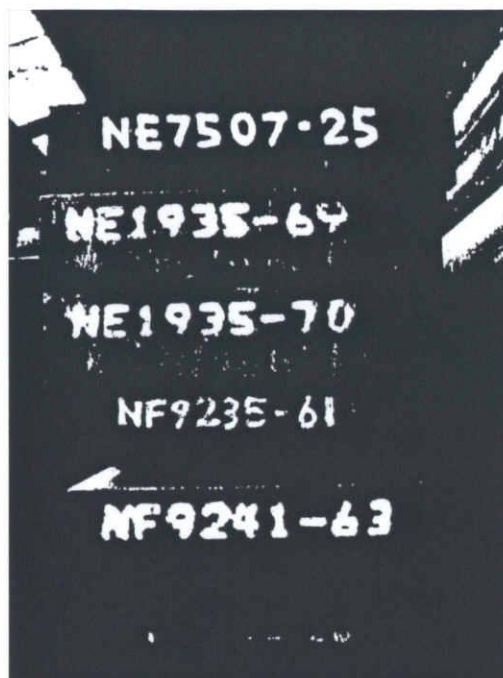


(d) 3.5(c) を 2 値化した画像

図 3.5: 期待した結果が得られた例



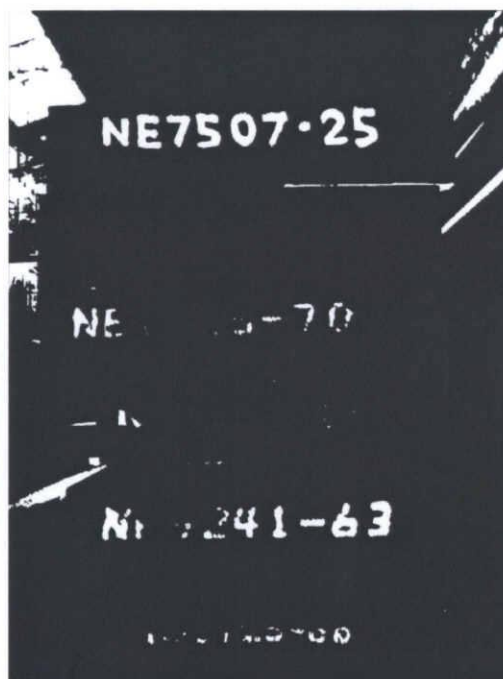
(a) 原画像



(b) 3.6(a) を 2 値化した画像

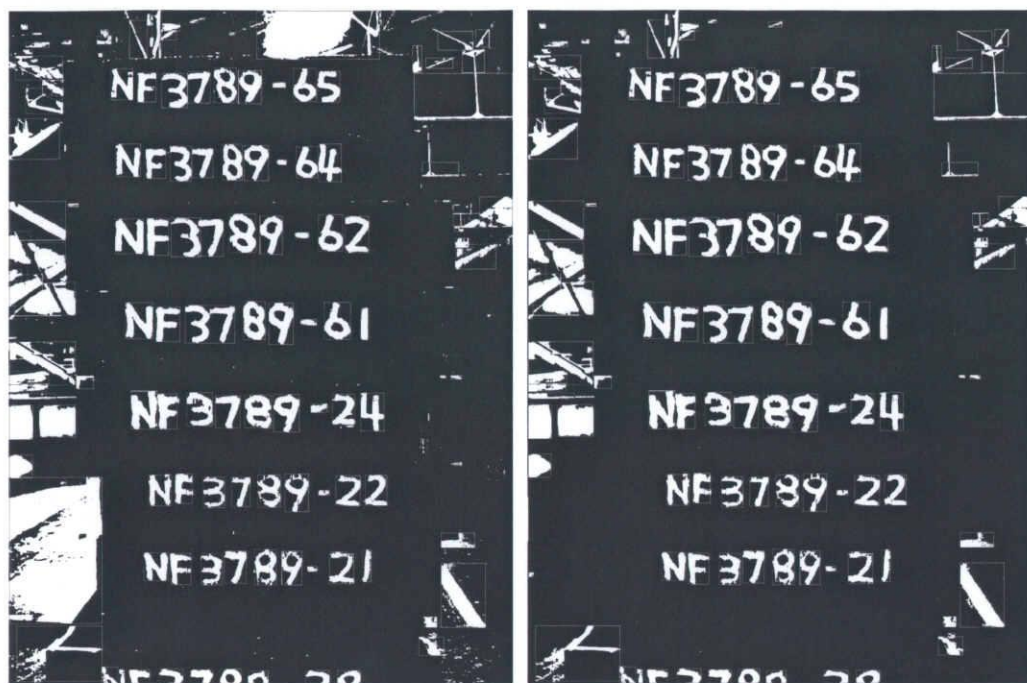


(c) 3.6(a) に高域通過フィルタ 3 を適用した画像



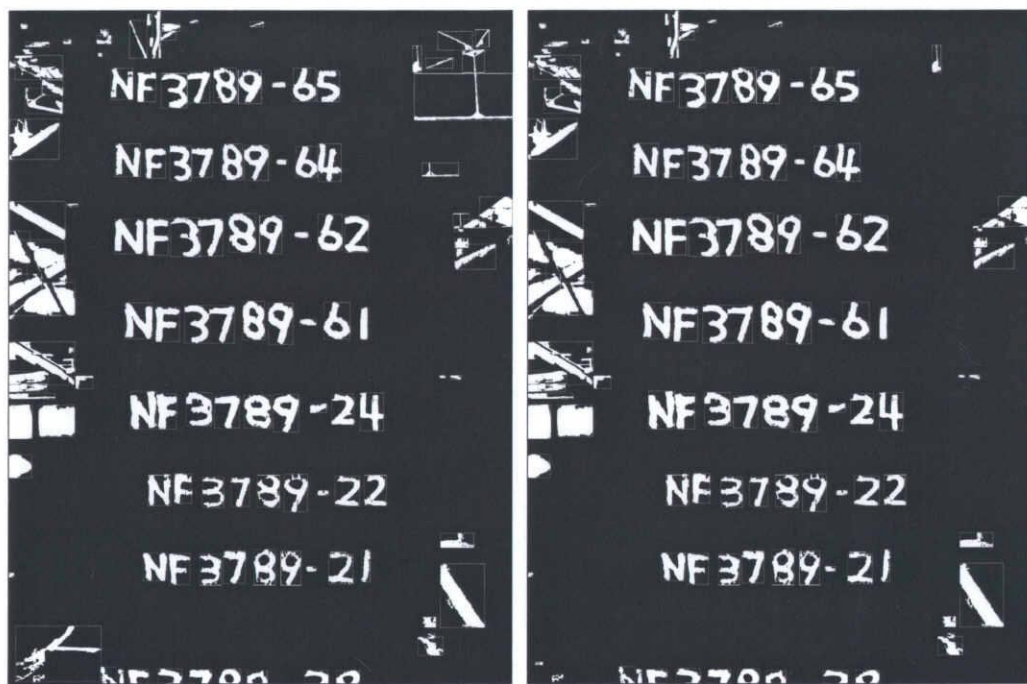
(d) 3.6(c) を 2 値化した画像

図 3.6: 期待した結果が得られなかった例



(a) ラベリングした画像

(b) 画素数や外接矩形の縦幅, 横幅でノイズ除去



(c) 外接矩形の縦横比を用いてノイズ除去

(d) 外接矩形の面積に対する連結成分の面積を用いてノイズ除去

図 3.7: ラベリング・ノイズ除去の様子

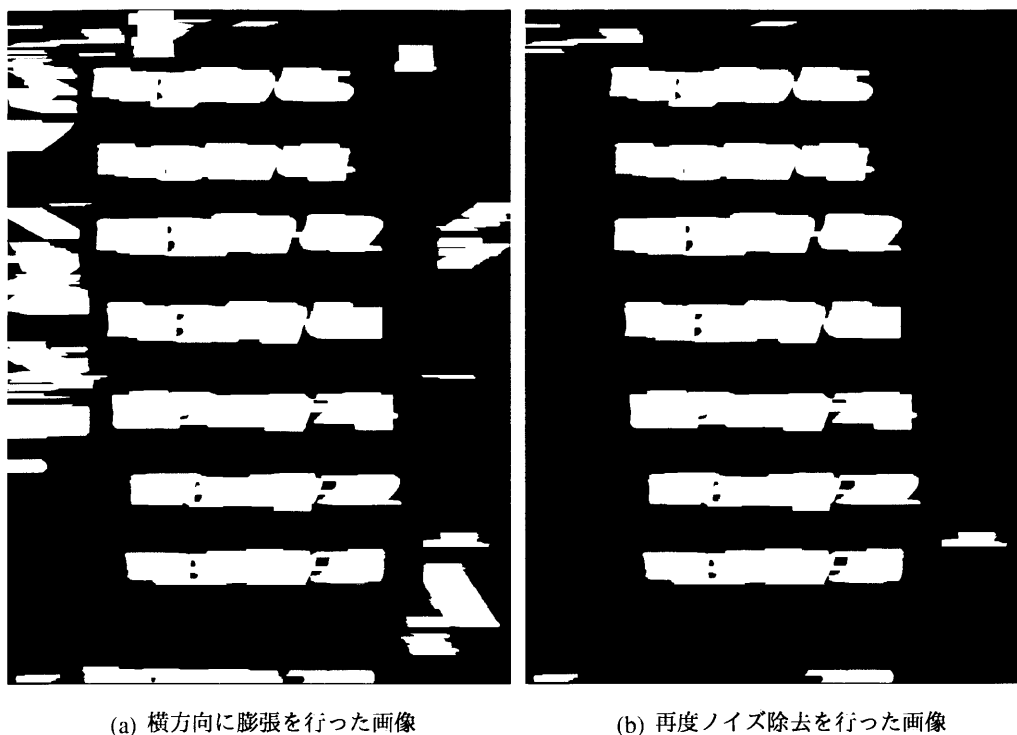


図 3.8: 膨張後, 再度ノイズ除去の様子

3.2.4 文字列領域の抽出

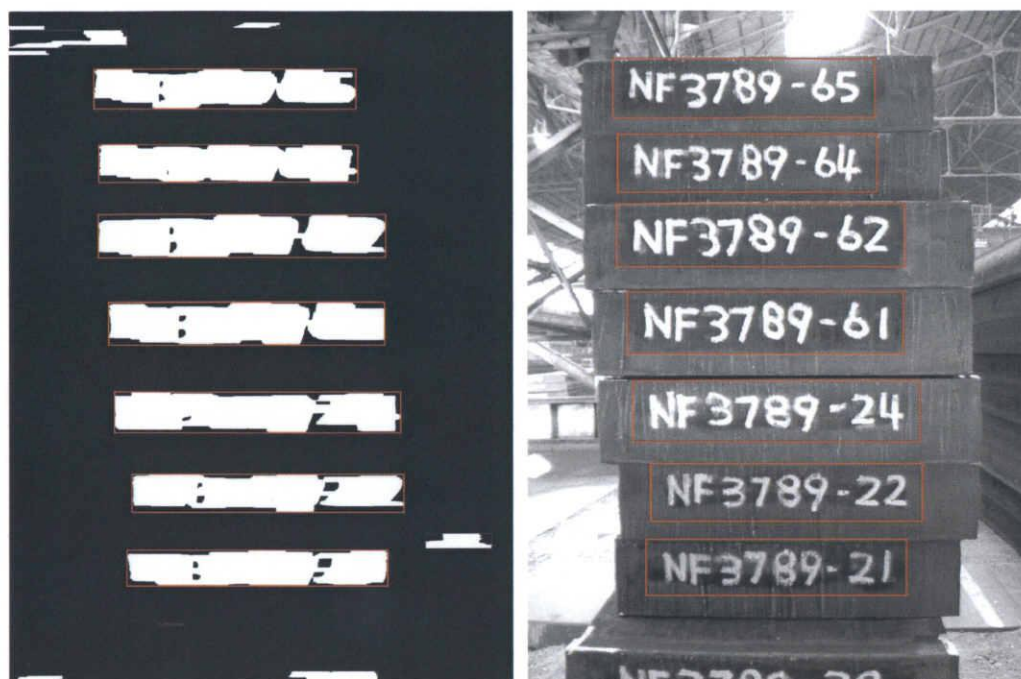
図 3.8(b) を見ると, 文字列部分に相当する連結成分は画素数が大きく, かつ文字列部分同士は同じような大きさであることが確認できる. このような特徴から, 以下の手順で文字列部分の抽出を行う.

連結成分の画素数で連結成分をソートする. S_i ($i = 1, 2, \dots$) を i 番目に画素数が大きい連結成分とすると, 次の条件を満たす S_i を文字列として抽出する.

1. S_i は文字列とする.
2. もし S_i が文字列ならば, $S_{i-1}/S_i > \alpha$ を満たす S_{i-1} は文字列である.

ここでは, $\alpha = 0.7$ とした. 文字列の抽出の様子を図 3.9(a) に示す. 図 3.9(a) は, 赤い線で囲まれた連結成分が文字列部分として抽出されたことを表している. この赤い線で示された外接矩形は水平方向に膨張した後の外接矩形であるため, 本来の文字列領域に対して水平方向には余裕があると考えられる. しかし, 垂直方向には文字列部分ギリギリであると考えられ, 画像 2 値化の段階で若干文字部分が欠落してしまった可能性を考慮し, 垂直方向にもマージンを持たせて最終的な文字列の切り出し枠を決定する. 切り出し枠を入力

画像に反映させた画像を図 3.9(b) に示す。入力画像にある文字列に対して適切に文字列の切り出し枠が抽出できていることが確認できる。



(a) 文字列の抽出の様子

(b) 入力画像に切り出し領域を反映させた画像

図 3.9: 切り出し枠の推定の様子

3.3 文字列画像から文字切り出しと認識

ここでは、1 画像 1 文字列に切り出された画像に対して 2 値化、文字の切り出し、文字認識について述べる。切り出された画像例を図 3.10 に示す。

まず、文字のかすれが少なく、かつノイズも少ない画像 2 値化を目指す。図 3.10 を判別分析法を用いて 2 値化した画像を図 3.11 に示す。鋼材の写り込みがなく文字もしっかりきれいに書かれている画像については判別分析法を用いて 2 値化することで、文字部分のみを白とする良好な 2 値画像が得られた。しかし鋼材が白く写り込んでいる画像については、鋼材ごと白く 2 値化されてしまい、文字が全く判読できない 2 値画像となってしまった。またかすれがひどい文字や液だれがおきている文字については 2 値化すると文字の一部が欠けてしまったり液だれ部分がノイズとして残ってしまったものがある。

このように、画像 2 値化の際、判別分析法のみでは良好な 2 値画像が得られない。そこで、このような画像に対しても良好な 2 値画像が得られる 2 値化手法が必要である。

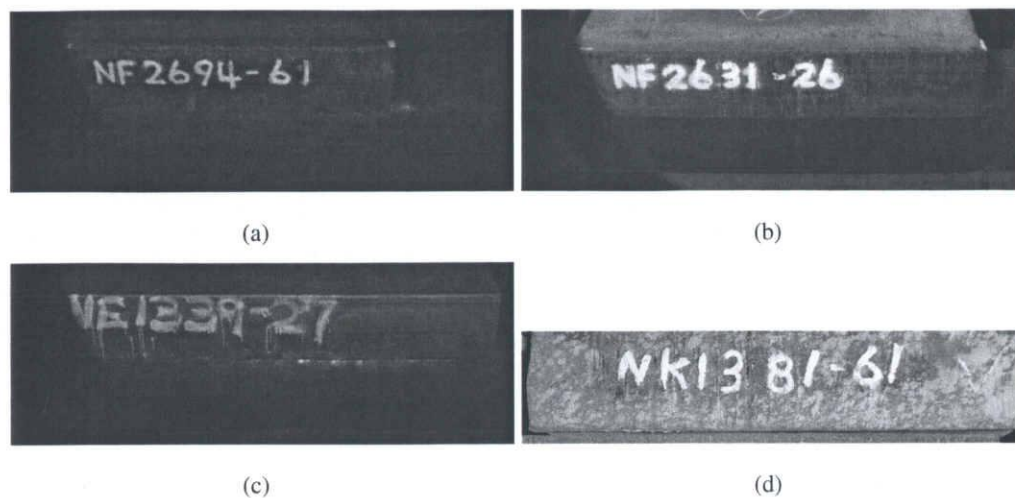


図 3.10: 切り出された画像の例

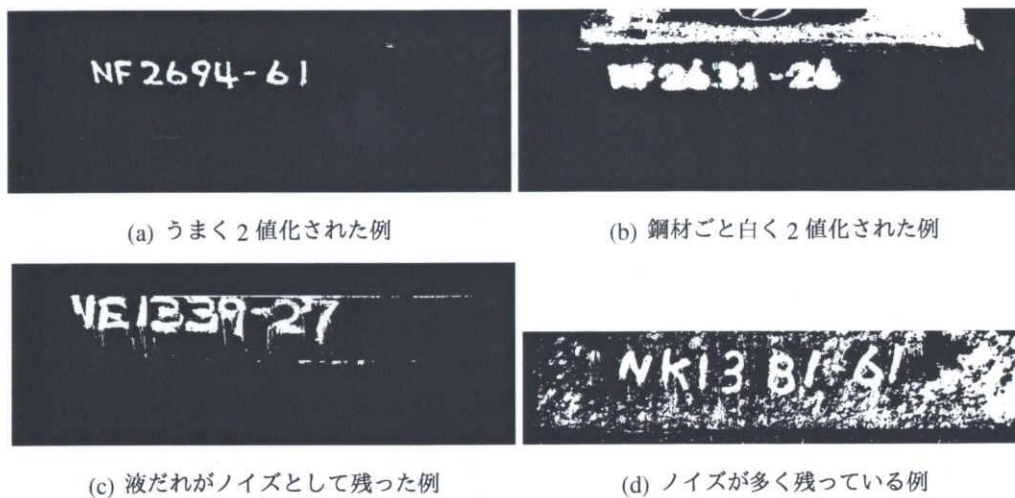


図 3.11: 図 3.10 を判別分析法で 2 値化した画像

3.3.1 ヒストグラム変換を用いた画像 2 値化

ヒストグラムの度数変換

判別分析法のみでうまく 2 値化できなかったものの多くは、図 3.11(b) のように鋼材ごと白く 2 値化されたものであった。これらの原因は、背景に対して対象図形の面積が非常に小さいためであると考えられる。図 3.11(b) のヒストグラムを図 3.12 に示す。

このヒストグラムは、横軸に輝度値、縦軸に度数をとったものである。輝度値 40 前後にある大きなピークが背景である黒色部分（以後、背景部分と呼ぶ）で、輝度値 50 から

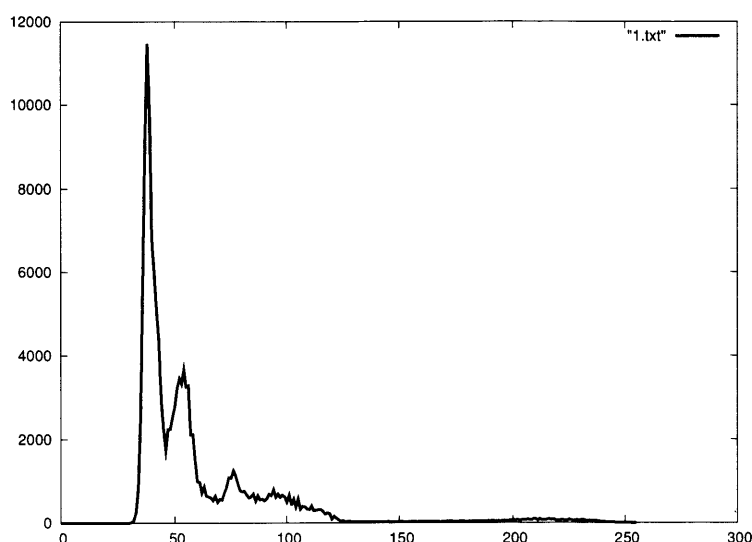


図 3.12: 図 3.11(b) のヒストグラム

130 あたりに分布しているのが鋼材が照明の影響で灰色に写った部分（以後、鋼材部分と呼ぶ）である。輝度値 200 前後にごく小さく分布しているのが白色の文字の部分（以後、文字部分と呼ぶ）である。鋼材部分と文字部分の間にしきい値を設定できれば文字のみを白とするような好ましい 2 値画像が得られるが、文字部分の分布が非常に小さいため、判別分析法で求めたしきい値は背景部分と鋼材部分の間の値となってしまった。

そこで、ヒストグラムの度数変換を行う。具体的には、ヒストグラムの各輝度値の度数を 0.5 乗する。この結果、ヒストグラムの度数が大きい輝度値ほど大きな影響を受け、度数が低いと小さな影響しか受けないので、輝度値間の度数の差が相対的に小さくなり、判別分析法に適したヒストグラムに近づく。図 3.12 を度数変換を行ったヒストグラムを図 3.13 に示す。

ここで、図 3.12 の縦軸の最大目盛は 12000、図 3.13 の縦軸の最大目盛は 120 であることに注意する。度数変換の結果、相対的に背景部分の度数が小さくなり、文字部分の度数が大きくなっていることが確認できる。

ヒストグラムの範囲限定

ヒストグラムの度数変換により背景部分の影響は相対的に小さくはなったが、いまだにヒストグラムの大部分は背景部分と鋼材部分が占めていて、文字部分は小さい。そこでこの性質を利用して、ヒストグラムの範囲を限定して判別分析法でしきい値を求めることを考える。その様子を図 3.14 に示す。

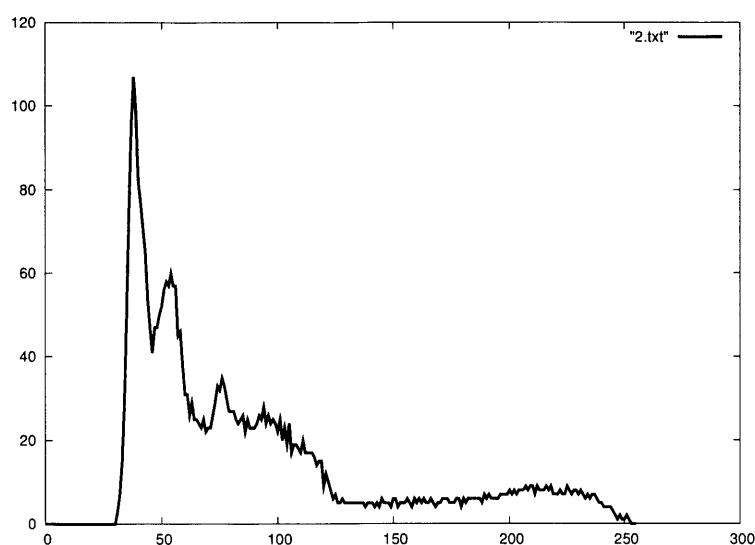


図 3.13: 図 3.12 の度数変換後のヒストグラム

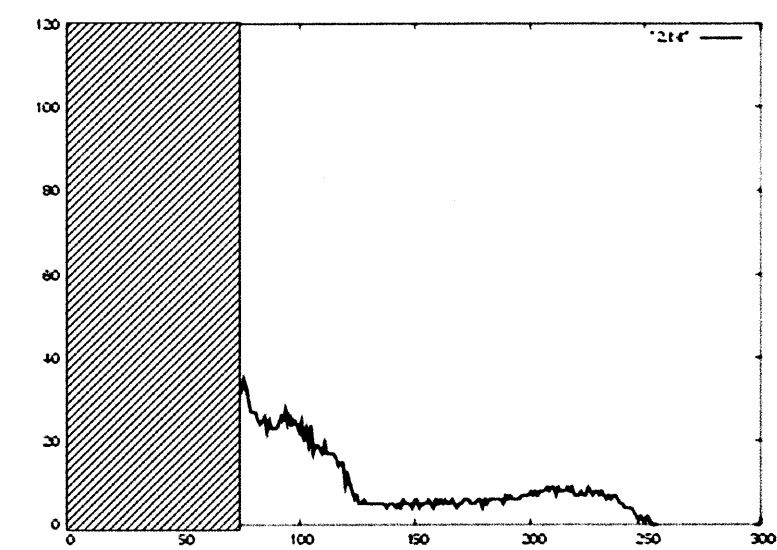


図 3.14: ヒストグラムの範囲限定をする様子

図 3.14 の斜線部はおおよそ背景部分に対応する部分であり、斜線部を除いた領域で判別分析法を行うことで鋼材部分と文字部分の範囲からしきい値を求めることになり、その結果、鋼材部分と文字部分を一番分離するような値をしきい値として出力することになるので、全体として最適なしきい値が求まる。図 3.14 では斜線部は全体の面積の 50% とした。

ヒストグラム変換を行った後、判別分析法により 2 値化を行う。得られた 2 値画像に膨張

収縮処理を施し、2 値画像中の小さな孔を埋め、収縮膨張処理を施して小さなノイズを除去する。得られた画像を図 3.15 に示す。



図 3.15: ヒストグラム変換を適用して 2 値化した画像

ヒストグラム変換を行うことで多くの場合適切な 2 値化結果が得られたのが確認できる。ただし図 3.15(c) のように 2 値化しきい値の工夫ではうまく 2 値化できないものもあった。

3.3.2 文字領域の推定

文字周辺にまだノイズが残っている可能性があるので文字領域の推定を行う。文字領域の推定の様子を図 3.16 に示す。

まず水平方向に画素値を投影する（図 3.16 右）。得られたグラフに一定しきい値を設け（グラフの緑線）、しきい値以上の区間を水平方向の文字領域とする（黄色の水平線）。次に垂直方向に画素値を投影する（図 3.16 中段）。垂直方向には文字と文字に間隔があるので、これを平滑化する（図 3.16 下段）。これにしきい値を設定し、しきい値以上となる最大の区間を文字領域と推定する。ただし、しきい値以下の区間が一定以下である箇所は文字と文字の間とした。

水平投影、垂直投影から推定した黄色で囲まれた区間を文字領域と推定し、それ以外の区間をノイズとして除去する。

3.3.3 文字認識

文字切り出しと認識

文字の切り出しと認識には、2 値画像を過分割して得られる複数セグメントに対して文字認識を行いながら動的計画法によって最適な文字の切り出しを行う [8, 9]。

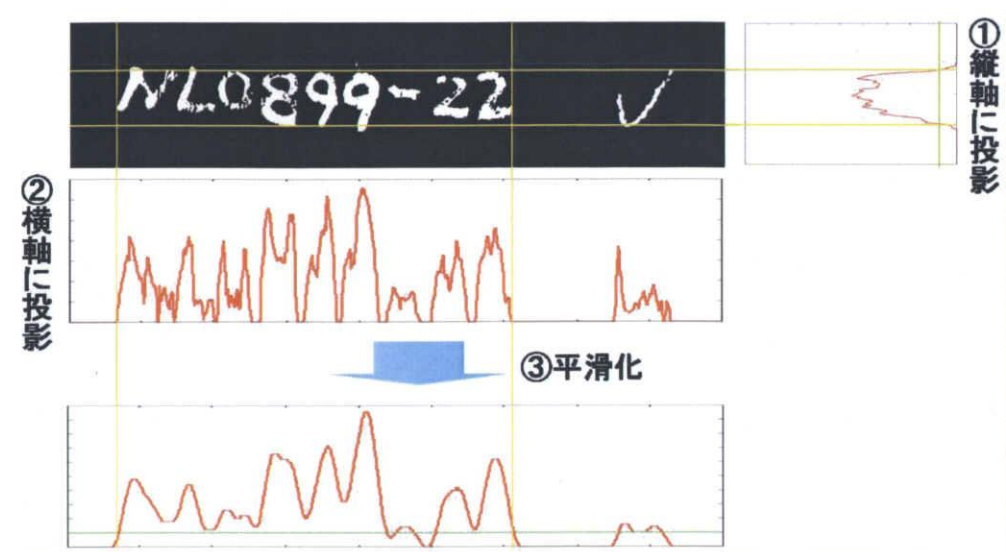


図 3.16: 文字領域の推定

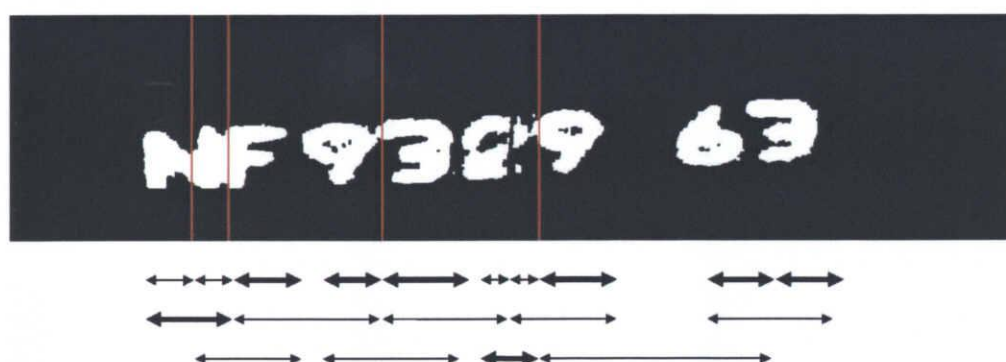


図 3.17: 動的計画法による文字切り出し

文字の切り出しでは、2 値化された文字列画像において、連結成分の輪郭線の垂直方向の極小点（谷の部分）でセグメントに分割する。得られたセグメントを 1 つ、または、2, 3 セグメントずつまとめた成分を文字認識して文字の尤度（文字らしさ）を求め、文字列の平均尤度が最大となるよう動的計画法を用いてセグメントの組み合わせを選ぶ。文字の切り出しの例を図 3.17 に示す。図 3.17 は、赤線で過分割した後、下部矢印の太線の組み合わせでセグメントを組み合わせたことを示している。この文字の切り出しにおける文字認識には加重方向指数ヒストグラム法 [10] による 64 次元の特徴ベクトルを使用する。また、文字列の認識においては、使用する文字は数字 '0' から '9' と '・' や特定の英字に限られるため、図 3.18 に示すようなテンプレートを用いた。ここで、% には 0 から 9 の数字が入る。

NE%%%%%%%%
 NE%%%%%-%%
 NE%%%%%-%%
 NE%%%%%-%%-%%
 NE%%%%%-%%-%%-%%
 NF%%%%%%%%
 NF%%%%%-%%
 NF%%%%%-%%-%%
 NF%%%%%-%%-%%-%%
 NL%%%%%%%%-%%
 NM%%%%%%%%-%%
 NK%%%%%%%%-%%
 ZE%%%%%%%%
 ZE%%%%%-%%

図 3.18: テンプレートの例

最初のテンプレートは英字 'NE' につづき数字 6 文字が入ることを示しており、2 番目のテンプレートは英字 'NE' につづき数字 4 文字が入り、ハイフン '-' の後、数字 2 文字が入ることを示す [11]. 個々のテンプレートに対して、テンプレートで指定される条件を満たす文字の組み合わせの中から動的計画法により最適解が求められる. さらに、各テンプレートに対する最適解の中から、文字の平均尤度が最大の最適解を最終的な認識結果とする. 文字の尤度計算には疑似ベイズ識別関数を用いた [12].

輪郭点チェインコードからの特徴抽出

文字切り出し、および、文字認識に用いる特徴ベクトルとしては、輪郭点のチェインコードから求められる加重方向指数ヒストグラム特徴を用いる. 文字の 2 値画像に対して文字部分の輪郭線を 8 連結で追跡し、各輪郭線の方角を 45° おきの 4 方向に量子化する. 2 値文字画像を縦 $(2m-1)$ 、横 $(2m-1)$ の計 $(2m-1)^2$ ブロックに分割し、ブロック別、方向別に輪郭画素数を数え、 $4(2m-1)^2$ 次元 (ブロック数 \times 4 方向) の方向指数ヒストグラムを得る. 5×5 の 2 次元ガウスフィルタを縦横 1 ブロックおきに施して、ブロック数を $(2m-1) \times (2m-1)$ から $m \times m$ に削減し、 $4m^2$ 次元 (縦 $m \times$ 横 $m \times$ 4 方向) の特徴ベクトル (加重方向指数ヒストグラム) を求める. 2 値文字画像に対する空間分割数 m を 4, 5, ...10 とすることで、次元数が 64, 100, ...400 の加重方向指数ヒストグラムを

作成することができる．図 3.19 に $m = 4$ の場合の特徴抽出の例を示す．この例では，ブロック数は 7×7 ，特徴ベクトルの次元数は 64 次元となる．

識別関数

特徴抽出手法によって得られる特徴ベクトルの識別・分類には疑似ベイズ識別関数を用いる．疑似ベイズ識別関数は，共分散行列が未知の正規分布に対するベイズ識別関数から導かれる．ベイズ識別関数は次式で与えられる．

$$g_l(X) = (N_N + 1) \ln \left\{ 1 + \frac{(X - M_l)^T \sum_{N_l}^{-1} (X - M_l)}{N_N} \right\} + \ln |\sum_{N_l}| - 2 \ln P(\omega_l) \quad (3.3)$$

ただし，

$$N_N = N + N_0$$

$$\sum_{N_l} = (1 - \alpha) \sum_l + \alpha \sum_0$$

$$\alpha = \frac{N_0}{N + N_0}$$

ここで， l はクラスに関する添字， N は各クラスの学習サンプル数， M_l は母集団の平均ベクトル， \sum_l は標本共分散行列， \sum_0 は事前分布の共分散行列で， N_0 は \sum_0 の信頼度を表す定数で信頼度定数と呼ぶべきものである．なお，サンプル数 N および \sum_0 ， N_0 は各クラスに共通とする．また，以後，クラスに関する添字 l は省略する．特徴ベクトル X の事前分布として球状分布を仮定して $\sum_0 = \sigma^2 I$ とし，計算量・記憶容量を削減するため式 (3.3) を変形すると次式の疑似ベイズ識別関数が得られる．

$$g(X) = (N + N_0 + 1) \ln \left[1 + \frac{1}{N_0 \sigma^2} \left\{ \|X - M\|^2 \sum_{i=1}^k \frac{\lambda_i}{\lambda_i + \frac{N_0}{N} \sigma^2} \Phi_i^T (X - M)^2 \right\} \right] + \sum_{i=1}^k \ln \left(\lambda_i + \frac{N_0}{N} \sigma^2 \right) - 2 \ln P(\omega) \quad (3.4)$$

ここで， λ_i ， Φ_i はそれぞれ共分散行列 \sum_l の第 i 固有値と第 i 固有ベクトル， k は識別に用いる固有ベクトルの数である．また，実際の計算では M は標本平均ベクトルで代用する． σ^2 は X の事前分布を球状分布と仮定した場合の分散で，その値として，全字種，全固有値の平均を用いる．加重方向指数ヒストグラムは物理的に同一種類の特徴によって構成されているため，事前分布を球状分布と仮定する．

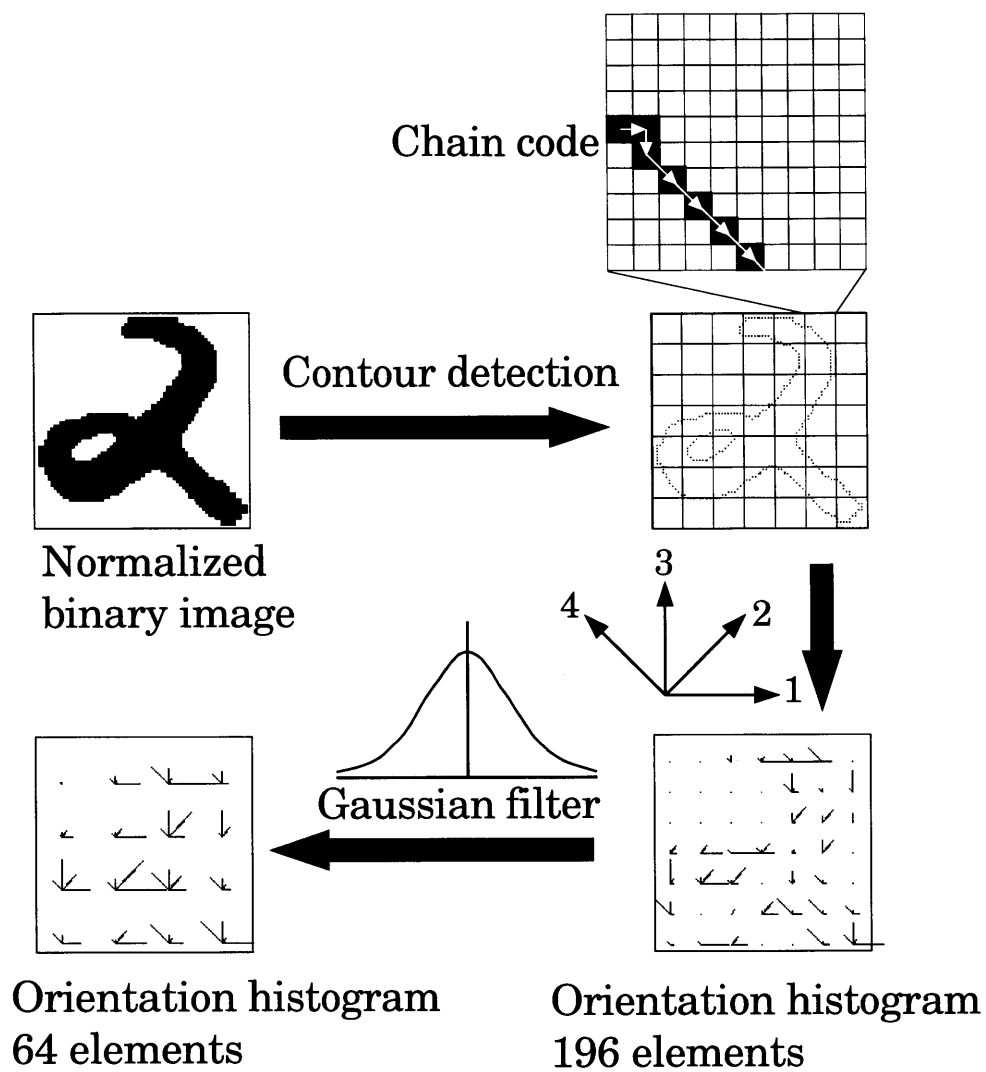


図 3.19: 輪郭点チェーンコードからの特徴抽出

第 4 章

実験

ここでは，文字列切り出し実験，および認識実験の実験方法，実験結果，考察を述べる．

4.1 文字列切り出し

4.1.1 実験と結果

入力画像 55 枚に対して，文字列切り出しの実験を行う．ここでの入力画像は図 3.2 のような 1 画像につき複数の文字列が写っている画像で，入力画像 55 枚に含まれる文字列総数は 365 である．入力画像に対して， 10×10 ， 20×20 ， 30×30 の領域分割 2 値化を行って切り出し，切り出された文字列画像に対して，適切な切り出しが行えたかどうかを目視評価する．

ここで，正解は“完全な文字列を 1 つだけ含んでいる画像”と定義し，それ以外は全て不正解とした．

実験結果を表 4.1 に示す．検出数とは切り出された画像数で，正解数とは切り出された画像が正解であった数を表している．ここで，検出率は文字列総数 (365) に含まれる正解数の割合，精度は検出数に含まれる正解数の割合を表している．

	検出数	正解数	検出率 (%)	精度 (%)
10 × 10 分割	343	317	86.8%	92.4%
20 × 20 分割	357	323	88.5%	90.5%
30 × 30 分割	369	302	82.7%	81.8%

表 4.1: 文字列切り出しの実験結果

4.1.2 考察

文字列の切り出しにおいて、最も重要なのはなるべく多くの文字列を検出することである。この段階で検出できなければ後の文字認識では認識すべき文字列がないので認識し得ない。逆に、この段階で文字列とは全く関係ない部分を検出してしまったとしても、後の文字認識で棄却できる可能性は高い。

ここで、精度は、システムが検出したものが本当に文字列である割合である。つまり、どの領域分割数においても検出したもののうち 80% 以上は本当に文字列であることを表している。これは後の文字認識での棄却の可能性を考えると十分な値であると考えられる。よってここでは検出率が一番高い 20×20 分割が最も良い結果であったと考えられる。

また、ここで検出できなかった文字列の多くは

1. 文字と文字を結合する際の膨張が足りず、文字列の一部を検出してしまったもの
2. 文字列領域抽出の際、適切な領域選択ができなかったもの

であった。これらは膨張や領域選択の際のパラメータをより適切に設定することで多少の改善は見込める。しかし、1 については、膨張させすぎるとノイズと結合しやすくなってしまう問題があり、2 については、大きな（文字列領域の大きさに近い、またはそれより大きい）ノイズがある場合や文字列の大きさに違いがある場合は適切なパラメータの設定だけでは対応できない問題があるので根本的な解決にはならない。検出率向上のためにはこれらを解決する方法を考案する必要があると考えられる。

4.2 2 値化・認識実験

4.2.1 実験と結果

文字列画像 399 枚に対してヒストグラム変換を用いた 2 値化を行い、得られた 2 値画像が判読可能であるかどうかを目視評価する。また得られた 2 値画像に対して認識実験を行う。ここで、文字列認識実験には全てのサンプルを学習に用い、文字認識実験は全サンプルを 5 分割して学習用と評価用に分けて実験を行った。

実験の結果、

- 文字列の 2 値化成功率は 95.2%
- 文字列認識率は 72.7%
 - 文字認識率は 98.99%

であった。

ただし、文字列認識率とは、文字列に含まれる全ての文字の認識に成功した割合である。

4.2.2 考察

実験結果から文字列の 2 値化には成功しているが、文字列認識に失敗している例が多いことがわかる。文字列認識の失敗の原因として、文字認識の失敗と文字切り出しの失敗がある。文字認識率は 98.99% であるので、平均文字数を 10 文字とすると、90% 以上の割合で 10 文字全てを文字認識できることが見込める。よって文字列認識の失敗の主な原因は文字切り出しの失敗であると考えられる。文字切り出し失敗の原因は、除去しきれなかったノイズの影響や文字のかすれなどが考えられるので、ノイズ除去や文字の整形の精度向上が文字列認識率の向上につながると考えられる。

第 5 章

むすび

5.1 まとめ

本研究は複数の文字列が写っている鋼材画像から文字列を切り出す手法、及び鋼材上の手書き文字列を認識する手法を開発した。文字列切り出しには、

1. 画像 2 値化の際の領域分割
2. 判別分析法を適用する前にヒストグラム変換を行うこと

が有効であることがわかり、88.5% の検出率を実現した。

また、手書き文字列の認識にもヒストグラム変換が有効であることがわかり、72.7% の文字列認識率を得た。

5.2 今後の課題

文字列画像切り出しの成功率や文字列認識率の向上のために以下の問題が今後の課題として挙げられる。

1. 文字列切り出しの際の、文字認識を用いた棄却の導入

現在は文字列画像の切り出しに文字認識は用いていないが、図 5.1 のように切り出し結果を文字認識し、文字認識の結果を切り出しにフィードバックさせることで切り出しの高精度化を図る。具体的には、切り出し結果が文字列らしくない場合はパラメータを変えて再度文字列候補を検出、というループ構造にする。この変更を行えば画像のサイズや文字の大きさに強く依存する固定パラメータを可変にできるのではないかと考える。

2. 文字列認識精度の向上

4.2.2 節で述べたように、文字列認識率の向上のためには文字の整形、ノイズ除去

精度の向上が必要である。具体的には、多値画像の段階でエッジ強調などを前処理を追加しインクのむらによる文字のかすれを減らす処理や、2 値化の際に、(a) 高めのしきい値と (b) 低めのしきい値の 2 枚の 2 値画像を作成し、(a) 上の白画素を含む (b) 上の連結成分のみを残し、文字かすれやノイズを軽減させる処理が有効ではないかと考える。

3. より多くの鋼材画像への適用

現在の画像数では十分であると言えないので、より多くの画像で提案手法の有効性を検証する必要がある。また、他の鋼材画像に提案手法を適用し、有効であるかどうかを確認し、どんな問題点があるのかを検証する必要がある。

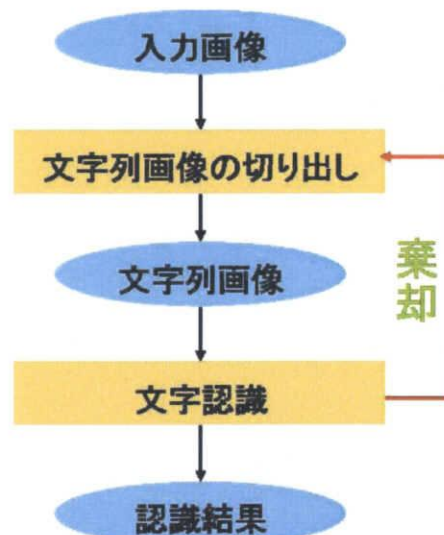


図 5.1: 棄却の導入のイメージ

謝辞

本研究に関し，研究に対する多くのアイデア，理論的裏付けや，計算機の使用方法などの指導をいただきました木村文隆教授，若林哲史助教授に深く感謝します．研究の進め方から発表のコツ，プレゼンテーションの作り方などを夜遅くまでご指導いただきました大山航助手に深く感謝します．また，日頃いろいろとお世話になった田中みゆき事務官に感謝します．日頃の研究生生活において多くの有益なアドバイスを下さった研究室の先輩方，時に討論し，時に励ましあった同期のみなさん，わきあいあいとした雰囲気楽しく過ごした研究室仲間に感謝します．

最後に，私の学生生活を支えてくれた両親に今一度の感謝を表して，本論文の結びといたします．

参考文献

- [1] HITACHI: ニュースリリース 2002 年 8 月 30 日
<http://www.hitachi.co.jp/New/cnews/2002/0830b/>
- [2] 内藤貴志, 塚田敏彦, 山田啓一, 小塚一宏, 山本新: “屋外環境下における走行中の車両のナンバープレート認識” 映像情報メディア学会誌, 映像情報メディア vol.53, no.5, pp.730-737 (1999.05)
- [3] 高橋裕子, 田中久子, 鈴木章, 塩昭夫, 大塚作一: “多値テンプレートマッチングを用いたナンバープレート認識法: 雑音抑制フィルタと文字配置規則の効果的利用”, 電子情報通信学会論文誌 (D-II), vol.J87-D-II, no.7, pp.1451-1461 (2004.07)
- [4] 熊野信太郎, 宮本一正, 玉川光明, 池田弘昭, 菅弘二: “コンテナ番号認識装置の開発”, 電子情報通信学会論文誌 (D-II), vol.J84-D-II no.6, pp.1073-1083 (2001.06)
- [5] 澤和弘, 若林哲史, 鶴岡信治, 木村文隆, 三宅康二: “FA における低品質文字列の認識別”, 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, vol.97 Num.501, pp.51-56 (1998.01)
- [6] 編著 田村秀之: “コンピュータ画像処理”, オーム社
- [7] 大津展之: “判別および最小 2 乗基準に基づく自動しきい値選定法” 電子情報通信学会論文誌 (D), vol.J63-D no.4, pp.349-356 (1980.4)
- [8] F.Kimura, Y.Miyake, M.Shridhar, “Handwritten ZIP Code Recognition Using Lexicon Free Word Recognition Algorithm”, ICDAR'95, pp.906-910, Aug.1995.
- [9] F.Kimura, M.Shridhar, Z.Chen, “Improvements of a Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words”, ICDAR'93, pp.18-22, Oct.1993.
- [10] 鶴岡, 栗田, 原田, 木村, 三宅, “加重方向指数ヒストグラム法による手書き漢字・ひらがな認識”, 電子情報通信学会論文誌 (D), vol.J70-D no.7, pp.1390-1397, July 1987.
- [11] F.Kimura, M.Shridhar, “Handwritten Address Interpretation Using Extended Lexicon Word Matching”, IWFHR'96, pp.396-372, Sep.1996.
- [12] 若林, 鶴岡, 木村, 三宅, “特徴量の次元数増加による手書き数字認識の高精度化”, 電子情報通信学会論文誌 (D-II), vol.J77-D-II, no.10, pp.2046-2057, Oct.1994.