画像ファイルの入出力処理法に関する研究

平成3年4月

岡田　守

概　要

ＬＳＩやローカルエリアネットワークなどのハードウェア技術や通信技術の発展に支えられて，計算機システムは大きな変貌を遂げつつある。その大きな特徴は処理の分散化と扱う情報メディアの多様化にある。処理の分散化についていえば，ソフトウェア技術の発展にも支えられて，既に，計算機システムの利用者は物理的に離れた計算機資源をネットワークを介して有効に活用することが可能になっている。情報メディアの多様化については，計算機システムで扱う情報メディアがデータ，画像，音声といったように多様化し大容量化の一途を辿っている。このため，これらを統合化して通信したり相互間での通信を可能とするマルチメディア通信の実現が期待されている。その実現のためには解決すべき課題が多い。その中でも，多様化する情報メディアのうち利用者に最も適した情報メディアに変換して情報の伝達を可能にすることが大きな課題となっている。これを解決するためには，特に，膨大な情報量を持つ画像を認識して伝送に必要な情報を抽出し編集や加工が容易なデータに変換する入力処理技術と，データを逆変換して画像を再構成する出力処理技術の確立が重要な課題となる。

本論文では，以上のような課題認識に立ち，画像ファイルの入出力処理の高度化に資するため，画像情報の検索サービスにおいて大量に利用される彩色図形を対象とした入出力処理法について議論する。

まず，第１章では，研究の背景としてマルチメディア通信においてパターン情報とコード情報を相互に変換するメディア変換処理の必要性を明らかにしたのち，彩色図形の入出力処理の位置づけについて述べ，研究の目的と意義を明らかにする。また，研究の狙いと課題を示し，研究内容の位置づけを明らかにする。

第２章では，彩色図形の輪郭線を構成する線図形の入力処理法について検討する。具体的には，線図形の制御を可能とし線図形入力過程における図形品質の向上を目的として，線図形の基本入力処理として細線化処理を取り上げ，処理の経済化と高速化について議論し，線順次型細線化処理法を提案する。

第３章では，彩色図形の符号化処理法として，ラスタ走査型表示装置に表示する条件で，彩色図形の輪郭情報を線分と円弧で表現しコード情報に変換する図形コマンド符号化法について検討する。技術的課題として点列で表現された輪郭情

報から線分と円弧の特徴を有する部分を抽出する方法について議論し，テンプレートマッチング法に基づいた手法を提案する。

第４章では，線順次型細線化処理と図形コマンド符号化処理の応用として，彩色図形を作成し符号化して画像ファイルに入力するための画像ファイル作成装置の構成法について検討する。線図形をファクシミリから入力し，彩色図形を会話的に作成する方式を取り上げ，装置の実現法と機能分担法を議論する。また，彩色図形の入力実験結果から装置の評価を行う。

第５章では，画像ファイルの出力処理法として，漢字パターンと図形パターンの発生法について検討する。漢字パターン発生法ではサイズ変換処理を取り上げて，高速化を達成するため比例変換法を基本原理にする条件で，その品質劣化要因を抑制する方法について議論し，品質劣化要因を抑制するためのパターン処理を比例変換の前処理として付加した任意サイズ漢字パターン発生法を提案する。図形パターン発生法では，２値表示の線分と円弧の発生法における表示品質上の問題を解決するため，線分と円弧のアンチエリアシング法について検討する。技術的課題として，高速化と円弧発生への拡張性を取り上げ，面積計算や距離計算によらない，変位を用いた輝度の変調原理によるアンチエリアシング法を提案する。

最後に，第６章では，本研究で得られた成果をまとめると共に，今後に残された課題について述べる。

目　　次

記号略号一覧

記号略号	意　味
$\wedge$	論理積.
$\vee$	論理和.
ϕ	空集合.
$b \in a$	bは集合aの要素.
$[x]$	xを越えない最大の整数.
$x \leftarrow y$	yをxに代入.
$\| r \|$	ランrの長さ.
$\| f(x,y) \|$	f（x，y）の絶対値.
A	画素文字パターン.
a_i	第i走査線上の画素情報.
B（P）	図形発生のための輝度変調関数.
$D_{[\,]}$	画素の座標値列.
F（T）	縦横線要素の識別しきい値Tの評価関数.
$\| f(P) \|$	座標値Pにおける曲線f＝0の変位.
$\| f_N(P) \|$	座標値Pにおける曲線f＝0の正規化変位.
H_L	線分の判別式.
H_A	円弧の判別式.
H_T	しきい値Tで類別された横線成分.
K	変位正規化のための基本しきい値.
mrp（r）	ランrの右端画素の位置.
mlp（r）	ランrの左端画素の位置.
max	最大値.
min	最小値.
N_h	横線成分の個数.
n_h	横線成分に含まれる要素（ラン）の個数.

記号略号	意　味
n_i	画素の座標値列D_{ij}の要素の個数.
Pl	漢字パターンの左ピッチ.
Pr	漢字パターンの右ピッチ.
R_i	漢字パターン拡大縮小のための縦方向倍率.
R_j	漢字パターン拡大縮小のための横方向倍率.
r_i	第ｉ走査線上の任意のラン.
$r_i(j)$	第ｉ走査線上のｊ番目のラン.
r_i^L	第ｉ走査線上の左端ラン.
r_i^R	第ｉ走査線上の右端ラン.
T	横線要素と縦線要素の識別しきい値.
V_T	しきい値Ｔで類別された縦線成分.
W_h	横線成分の平均線幅.
W_v	横線成分の平均線幅.
z_j	補助メモリの第ｊ番目の要素.
$\alpha(i,j)$, α_{ij}	ｉ行，ｊ列の画素の値.
η	データ圧縮率.
ρ	計算処理量の相対比.
ＣＲＴ	Ｃａｔｈｏｄｅ　Ｒａｙ　Ｔｕｂｅ.
ＪＩＳ	日本工業規格.
ＭＡＣＳ	任意倍率拡大スムージング法.
ＲＡＬＴＨ	任意倍率細線化縮小法.

図　一　覧

表　一　覧

第１章　序　論

１．１　研究の背景と目的

ＬＳＩやローカルエリアネットワーク（LAN： Local Area Network）などに代表されるハードウェア技術や通信技術の発展に支えられて，処理装置や端末装置の高性能化と通信の大容量化が可能となり，計算機システムにおいては処理の分散化と共に，処理対象となる情報メディアの大容量化が進展している。

また，事業所においては定型業務の事務処理の効率化を目的にしたＥＤＰ（Electronic Data Processing）が広く普及している。更に，近年のオフィスオートメーション気運の高まりに伴って，非定型業務の事務処理の効率化を目的に，文書をイメージ情報として通信するファクシミリ通信をはじめとして，文書の作成や管理・交換などを行う文書処理[1]，遠隔地相互間で動画像を通信し会議を行う画像会議[2]，イメージ情報の保管・検索[3]，ＣＡＤ（Computer Aided Design）による設計業務の支援[4]，音声による電子メールサービス[5]など，広い業務分野でシステム導入が活発に進められている。このように技術の応用的側面を概観すると，応用形態の多様化と共にシステムの中で扱われる情報メディアが，文字コードや数値などのデータのみならず，ファクシミリ，図形，映像などの画像や音声と多様化していることが大きな特徴であるといえる。

このようなシステムは，扱う情報メディアに応じて音声は電話網，データはパケット交換網，ファクシミリはファクシミリ交換網，映像は広帯域交換網で伝送し交換するといったように，情報メディアに特化した形態で構築されてきたため，同一の情報メディア間での通信しかできなかった。しかし，通信網のディジタル化や端末の知能化の進展と共に，異なる情報メディアを相互に関連づけて通信したり，一つの端末で複数の情報メディアを同時に扱うことが求められるようになった。このため，システムの経済化や応用形態の多様化への対応の観点から，データ，画像，音声などの情報メディアを混在させ統合化して通信したり，情報メディア相互間での通信を可能とするマルチメディア通信システム[6]の実現が期待されるようになった。

マルチメディア通信システムは，情報量が比較的少ないデータから膨大な情報

量を持つ画像まで，可変長の情報を統合化して通信したり蓄積することや，文字，画像，音声などの入出力を伴った高度なヒューマンインタフェースを持ち，利用者に最も適した情報メディアに変換して情報を伝達することが大きな特徴といえる。特に，人間が視覚を通じて獲得する情報量の多さから，画像の通信，蓄積，入出力がヒューマンインタフェースに果たす役割は大きいといえる。しかし，画像は膨大なデータ量を持つことから，文字などのデータを扱う場合と同程度の応答性能で，かつ，利用者に適した情報メディアでもってヒューマンインターフェースを提供するには，解決しなければならない多くの課題がある。このため，多くの分野において画像の扱いを考慮した研究が精力的に行われている。

通信技術については，マルチメディア通信を目的としたＩＳＤＮ（Integrated Services Digital Network）の建設や，通信網の高速広帯域化を目的とした高速回線交換技術や高速パケット交換技術[7]，可変長データの高速伝送を可能とする非同期転送モード通信技術[8]，および，データ，音声，画像，ファクシミリデータなどの混在伝送を可能とする通信プロトコル[9]などの研究・開発が活発に行われている。

情報の蓄積技術については，視覚情報処理システムや画像通信システムの発展に伴って，文書の構造に着目した文書データベース[10]，設計・製造データを管理するＣＡＤ／ＣＡＭ用データベース[11]，地図・図面データベース[12]，画像情報データベース[13]など，視覚情報を対象としたデータベースの研究が活発化している。

入出力技術のうち入力技術については，ファクシミリ，撮像装置，各種スキャナなどのハードウェアが高性能化し，画像を画素の集合として精度よく入力することが可能になった。出力技術については，ワークステーションの進歩の中に見られるように，ハードウェア技術に支えられて画面上に複数個のウィンドウと呼ばれる表示領域を配置し，個々のウィンドウがあたかも独立した表示装置のように利用できるマルチウィンドウ表示方式[14]が普及し，文字，図形，画像などの混在表示が可能になっている。

このような技術の進展に伴って，マルチメディア通信の特徴の一つである情報の通信，蓄積，入出力における情報メディアの統合化の問題は，今後，克服されて行くものと考えられる。これに対し，重要なのはマルチメディア通信のもう一

方の特徴である情報メディア相互間の変換に関する課題である。利用者に最も適した情報メディアでもってヒューマンインタフェースを提供するためには，データ（コード情報），画像，音声などを相互に変換するメディア変換機能を備える必要がある。メディア変換は，入出力処理の関係で捉えると，文字，画像，音声などのパターン情報を，文字コードや数値情報からなるコード情報に変換するパターン・コード変換と，コード情報を文字，画像，音声などのパターン情報に変換するコード・パターン変換で構成できる。パターン・コード変換の基本になるのは，文字認識技術，画像認識技術，音声認識技術などの認識技術であり，コード・パターン変換の基本になるのは，文字パターン発生技術，画像発生技術，音声合成技術などの合成技術である。

人間が視覚を通じて摂取する情報の多さに対応して，非電話系サービスとして考えられている多くがビデオテックス，ファクシミリ，描画像通信，静止画像通信などの画像通信サービスであり[15]，マルチメディア通信への発展が期待されている。このことから，画像を対象としたメディア変換機能が特に重要であるといえる。他方，ハードウェア技術の進歩に支えられて画像入出力装置の高精度化は更に進展することが考えられ，システムで扱う画像データは飛躍的に増大して行く傾向にある。これに応えるため，画像の通信・蓄積におけるデータ量の圧縮を目的とした画像符号化技術の研究も活発に行われ，画素間の相関性やフレーム間の相関性などの統計的性質を利用したファクシミリ画像，自然静止画像，動画像などの符号化技術が実用化され[16]，通信時間の短縮や記憶容量の削減を可能にしてきた。

しかし，人間の情報処理能力は1秒間当たり数十ビットであるといわれているように，真に必要とする情報量は更に少ないと考えられている[17]。このことから，伝送路や記憶媒体などの大容量化が進展する中にあっても，画像から必要な情報を抽出し，更に再構成する問題は，ヒューマンインタフェース高度化のため解決しなければならない問題であるといえる。

したがって，画像を認識して必要な情報を抽出し，編集や加工が容易なコード情報に変換する入力処理技術と，コード情報を逆変換して画像を再構成する出力処理技術の確立が，マルチメディア通信を実現する上で重要な技術的課題であると考えられる。

本論文では，以上のような背景の下で，非電話系サービスとして注目されている画像情報の検索サービス[18],[19]において大量に利用される，輪郭線で囲まれた領域が同一色で塗り潰された彩色図形を対象とした入出力処理法について検討し，画像ファイルの入出力処理の高度化に資するため，以下のことを調べることを目的とする。

（１）彩色図形の輪郭線を制御可能な形式に変換し，品質の良い彩色図形を得ることを目的に，２値線図形を１画素幅に細めるための線図形入力処理法として細線化処理について検討する。技術的課題として，テレビ画面以上の比較的画素数が多い画像を扱う場合の処理の高速化とメモリ削減による経済化を取り上げる。具体的には，線順次型細線化アルゴリズムを考察し，処理時間と所要メモリを評価し高速化と経済化が図れることを明らかにする。

（２）彩色図形をコード情報に変換することを目的に，彩色図形の輪郭形状を線分・円弧などの図形要素で表現する図形コマンド符号化法について検討する。技術的課題として，画素列から線分成分と円弧成分を認識する方法について考察し，実験結果により図形コマンド符号化の実現性を示す。

（３）線順次型細線化処理と図形コマンド符号化処理の応用として，線図形を入力して彩色図形を会話的に作成し，コード情報に変換して画像ファイルに入力するための装置を取り上げ，実現方式や機能分担などについて考察する。また，彩色図形の作成実験を行い，装置の実現性を示すと共に線順次型細線化処理と図形コマンド符号化処理の有効性を示す。

（４）文字コードや図形コマンドなどのコード情報を変換し，品質の良い画像パターンを得ることを目的に，文字・図形パターンの発生法について検討する。具体的には，比例法における品質劣化要因を抑制する前処理を付加した画素型漢字パターンのサイズ変換法と，線分・円弧のアンチエリアシング法について考察し，実験結果によりそれらの有効性を示す。

１．２　研究の位置づけ

本論文で対象とする画像（彩色図形）の入力処理と出力処理は，互いに密接に関連している。１．１節で述べたように，入力処理は伝送すべき情報を抽出する

過程，出力処理は受信した情報を解釈して画像を再合成する過程とみなせる。このような認識に立って，画像から抽出されコード情報への変換対象になる知識の内容に応じて，入出力処理の世代区分が考えられている[20]。第１世代は，画素間相関などの統計的性質を利用した方式であり，現在実用化されている動画像や静止画像の符号化方式[18]の大部分が該当する。第２世代は，画像の輪郭情報などの特徴をコード情報に変換する方式であり，入力処理において輪郭線抽出処理[21]や芯線抽出処理[22]などを施して得られた特徴を図形要素で表現し，出力処理において図形要素をパターンとして発生させ，画像を再合成する方式である。第３世代は，送信側と受信側で画像のモデルに関する知識を共有し，モデルを記述するためのパラメータをコード情報に変換する方式である。これは，入力処理において画像を解析しパラメータの値を計測し，出力処理においてパラメータの値からモデルを変形させて画像を再合成する方式であり，スプライン関数やベジェ関数などの高次曲線・曲面を用いたモデル化技術[23]や，コンピュータグラフィックスに代表される画像創成技術[24]が基盤になると考えられる。第４世代は，画像の構造を認識・理解した結果をコード情報に変換して伝送する方式であり，この実現のためには，特に，近年精力的に研究が行われているコンピュータビジョンシステムなどに見られる画像理解技術[25]への期待が大きい。

本研究で対象としている彩色図形の入出力処理は，彩色図形の輪郭の形状を線分や円弧などの図形要素で表現して扱うものであり，第２世代に位置づけられるといえる。以下に，本研究で扱う線図形入力処理，図形コマンド符号化処理，任意サイズ漢字発生法，および図形発生法の位置づけについて述べる。図１．１にそれぞれの研究経過を示す。

（１）線図形入力処理法の研究

彩色図形の入力処理の目的の一つは，彩色図形を品質良く入力することにある。彩色図形の入力は，線図形を入力する過程と，線図形で囲まれた領域を彩色する過程に分けられ，前者の処理過程における図形品質の向上対策として，線幅の変更機能や不要輪郭線の除去機能などの線図形の制御機能を実現するため，２値線図形を１画素幅に細める細線化処理法を取り上げる。

２値図形の細線化処理の研究は，パターン認識におけるパターン構造の特徴抽出の前処理法の研究として開始され，１９６０年代後半から１９７０年代前半に

かけて処理アルゴリズムの原理に関する多くの研究[26]-[31]が行われた。

国外においては，１９６６年，A.RosenfeldとJ.L.Pfaltz[26]はディジタル画像の処理アルゴリズムを逐次処理と並列処理に体系づけ，逐次処理による連結成分の抽出，距離変換，距離変換画像からのスケルトン抽出のアルゴリズムを示した。１９６７年，C.J.Hilditch[27]は３×３画素マトリクスの局所操作によるディジタル画像の細線化アルゴリズムの原理を提案した。また，１９７０年には，A.Rosenfeld[28]は画素の連結性の概念を導入することにより，ディジタル画像のトポロジカルな性質を体系化した。これにより，ディジタル画像の縮退，芯線化，細線化，連結成分のラベリングに関するアルゴリズムの妥当性が示された。更に，１９７１年，R.SteffanelliとA.Rosenfeld[29]はハードウェア向きの並列処理による細線化アルゴリズムを提案した。

国内においては，１９７３年，横井，鳥脇，福村[30]は連結数の概念によりディジタル画像の性質を体系化し，アルゴリズムを定式化した。更に，１９７５年，田村[31]は細線化アルゴリズムの比較結果を発表した。

細線化アルゴリズムの基本的な考え方は，マスクと呼ばれる３×３の画素マトリクスで表される局所領域に着目し，ラスタ走査を繰り返しながら中心の黒画素の消去可能性を周囲の８隣接画素の論理演算で判定するものであった。このため，画像の画素数の増大に伴って処理計算量と所要メモリが飛躍的に増加することから，１９７０年代後半から１９８０年代前半にかけて，処理の高速化とメモリ削減による経済化に関する研究[32]-[36]が見られるようになった。

国外においては，１９７８年，G.Woetzel[32]は線図形の最大幅をｋとしたとき（ｋ＋３）走査線分のメモリを使用して細線化する方法を提案した。原理的には３×３の画素マトリクスの局所操作に基づいた方法であり，画素の状態を端点，内部点，節点に分類しそれらの連結関係から芯線を抽出する方法である。１９８０年，１９８１年には，T.Pavlidis[33]とC.Arcelli[34]は，それぞれ，図形の輪郭線を追跡して細線化処理を高速化する方法を提案した。その原理は，輪郭追跡を行うことにより，非図形部分の画素の参照を不要にする考え方に基づいている。これらの方法は処理計算量が対象図形の太さに依存していたため，１９８５年，C.ArcelliとG.S.di Baja[35]は対象図形の太さに依存せず，ほぼ６回のラスタ走査で細線化できる方法を提案した。その原理は，距離変換画像の対称性に着

目し，等距離を有する画素の連結パスを抽出する考え方に基づいている。

国内においては，１９８４年，中山，木村，吉田，福村[36]は処理の経済化を目的に，図形の最大幅の２倍に等しい数の走査線メモリを用いて細線化する方法を示した。この方法は，原理的には３×３の画素マトリクスの局所操作に基づいた方法であるが，走査線メモリ内の走査が完了する度に，処理結果を走査メモリ間で移送させるパイプライン方式を採用している点に特徴がある。

これに対し，筆者は，１９７８年に２値図形の細線化処理の高速化と経済化に関する研究を開始し[37]-[40]，１９８１年に図形の読み取り走査に同期させて細線化する線順次型細線化法を提案した[41]。この方法は，簡易な図形情報入力装置と考えられるファクシミリに適用することを前提条件として開発したものであり，３走査線分のメモリのみを用い，かつ，走査線間の黒画素ランの連結関係を調べてパイプライン的に処理することにより，経済化と高速化の両条件を満足させた。

（２）図形コマンド符号化処理法の研究

彩色図形の入力処理のもう一方の目的は，入力した彩色図形から伝送すべき情報を抽出することである。図形コマンド符号化処理は，伝送すべき情報として彩色図形の輪郭情報を抽出し，線分や円弧などの図形要素で表現しコード情報に変換する処理である。このため，輪郭情報から線分や円弧の特徴を有する部分を認識することを課題として取り上げる。

図形を表現する手法は，大きく二分類されると考えられる。一つは，画素の集合で表現されたパターンを解析して，近似曲線の導関数が不連続となる屈折点や分岐点などの特徴点を抽出することに重点を置いて，幾何学的な特徴を捉えることが容易な線分や円弧で図形を近似する解析的手法である。他方は，スプライン関数[42]やベジェ関数[43]などの高次関数を用いて，離散点列を通り導関数が連続の滑らかな曲線を創成することに重点を置いた合成的手法であり，今日のコンピュータグラフィックスにおけるモデル化技術[23]の基礎になっている手法である。本研究は，彩色図形をパターン情報として入力し輪郭形状の特徴を表現する観点から，前者の解析的手法を対象としている。

解析的手法に基づいた図形表現法の研究は，パターン認識における図形構造の特徴抽出法の研究として行われ，１９７０年初頭より１９８０年代前半にかけて，

多くの研究報告[44]-[50]がある。

国外においては，１９７０年，U.Montanari[44]は，周長を最小化する多角形近似法を，輪郭を近似する多角形の周長をコスト関数とし，各点の許容誤差を制約条件とする非線形計画問題として一般化した。１９７２年には，C.T.ZahnとZ.Roskies[45]は，図形の輪郭形状を周波数解析し，フーリエ展開係数の組で表現する方法を提案した。１９７４年には，T.PavlidisとS.L.Holowitz[46]は，近似線分の個数を最小にする近似法を提案し，近似許容誤差を制約条件とした線分列の最小化問題として定式化した。また，１９７８年と１９８１年には，C.M.Williams[47],[48]は，点列で表現されたディジタル曲線から，各点の許容誤差範囲を通過する最大長の線分を探索する問題として定式化した近似法を発表した。

国内においては，１９８１年に，名倉[49]は，曲線上の点Ｐを中心として曲線に沿ってＮ画素前後した２点で形成される弦に対して，点Ｐから下ろした垂線の長さを用いて，屈折点，直線部，曲線部を識別する方法を提案した。この方法は，曲線の曲率変化の度合いを垂線の長さの変化で捉えた方法であり，原理的には輪郭形状の周波数解析を根源とする方法といえる。１９８２年には，佐藤[50]はディジタル曲線の最適折れ線近似を，周長を目的関数としてこれを最大化する点列の選択問題として定式化した。また，１９８５年には，安居院，飯塚，中嶋[51]は，ディジタル曲線と近似直線との平均誤差を最小化する問題として捉えた直線近似法を発表した。

また，１９８０年代には，これら図形近似法を２値図形や彩色図形の符号化に応用した，図形コマンド符号化の研究報告[52]-[54]が見られるようになった。

１９８０年には，K.Ramachandran[52]は，ファクシミリ信号のラン端点を線分近似しベクトル表現する符号化法を提案し，設計図面のデータが４０分の１近くまで圧縮できることを示した。１９８５年には，星野，小倉，河久保[53]は，ラン端点を線近似する符号化法を彩色図形の符号化に拡張した。また，１９８８年には，秦，富田，大西，中田[54]は，ラン端点を線近似する符号化法の発展型として，彩色図形を塗り潰し多角形と塗り潰し円弧の集合で表現し符号化する方法を提案した。

これに対し，筆者は，１９７３年に，線分と円弧を用いた図形表現による図形コマンド符号化法の考え方を発表した[55]。１９７６年には，変位比較を利用し

てディジタル曲線を線分と円弧で近似する方法を提案した[56]。更に，１９７９年，１９８１年には，この近似法を利用した彩色図形のコマンド符号化法を提案した[57],[58]。提案した線分と円弧による近似法は，ディジタル曲線に対して線分または円弧を仮定して，各画素からの距離が許容範囲内となる最大の線分または円弧を探索する問題を扱っており，原理的にはテンプレートマッチング法に基づいた手法といえる。また，近似処理に要する計算量を削減させるため，近似曲線を曲面とｘｙ平面との交線として捉えて，画素と近似曲線との距離をその画素から曲面に至るｚ軸方向の高さ（すなわち，変位）の大きさで判定している点に特徴がある。

（３）任意サイズ漢字発生法の研究

漢字パターンは彩色図形を構成する要素の一つである。漢字を主要な意志伝達手段とするわが国においては，漢字パターンの出力処理技術が表示画面の表現能力や品質の決定要因の一つとなる。すなわち，図や表を含んだ画面表示などソフトウェア機能の充実を背景として，漢字パターンのサイズをきめ細かく制御したいという要求がある。漢字パターンのサイズ変換処理は，画素型パターンを対象とした手法[59]-[64]と，輪郭表現されたパターンを対象とした手法[65]に分けられる。画素型パターンを対象とした手法の研究は，１９７０年代後半から１９８０年代前半にかけて，漢字パターン発生器の経済化や解像度が異なる種々の出力装置への日本語出力を目的として行われ，現在の表示装置における漢字表示方式の主流を占めている。一方，輪郭表現されたパターンを対象とした手法の研究は，１９８０年代後半から見られるようになり，現在では印刷装置への適用例が多く見られるようになってきた。本研究では，このうち，画素型漢字パターンのサイズ変換処理について議論する。

１９７５年に，黒崎[59]は，画素形漢字パターンのサイズ変換の必要性とその方法を示した。この方法は，画素のサイズを単純に整数倍する考え方に基づいた方法であり，原理的に整数倍の拡大に限られていた。そこで，縮小まで含めた任意倍率のサイズ変換処理の研究が開始され，１９７７年には，森[60]は，線分比例配置法を発表した。この方法は，漢字の主要構成要素である縦線分と横線分を抽出し，倍率に応じて線分の位置と幅を制御する考え方に基づいた方法である。この方法は，更に，１９７９年には，渡部等[61]により高速化へと改良されたが，

文字品質が縦線分と横線分を識別するためのしきい値に依存していた。

１９７９年には，井上，木村，武川[62]は，サブパターン変換法を提案した。この方法は，漢字パターンをサブパターンに分割し，変換倍率に対する各サブパターンの変換規則をテーブル化する方法である。この方法は，文字品質の劣化を抑制するために，変換倍率毎に異なるテーブルを必要とするので，テーブル容量の制約から変換倍率が制限されるという問題があった。同じく１９７９年には，斉藤，松葉，杉山[63]は，画素間論理演算法を提案した。この方法は，画素単位に周囲画素との論理演算を行い，画素の挿入や削除を行う方法である。この方法も，文字品質の劣化を抑制するために，変換倍率に応じて論理演算式を定義する必要があり，変換倍率に制約があった。サブパターンの変換規則を論理演算で表現したものとみなすと，サブパターン変換法と画素間論理演算法は，原理的は同様の手法といえる。

また，１９８２年には，小川，中根，池沢[64]は，差分パターン法を発表した。この方法は，原漢字パターンを単純比例変換した中間のパターンと，サイズ変換後のパターンとの差分パターンを記憶しておき，サイズ変換時に中間のパターンと差分パターンの排他的論理和をとる方法である。この方法は，文字品質の劣化を抑制するために，漢字パターン毎，変換倍率毎に変換規則を差分パターンとして持たせるという考え方に基づく手法であり，原理的にはサブパターン変換法や画素間論理演算法に近い手法といえる。

これに対し，筆者は，１９８６年に，漢字パターンサイズ変換法として任意倍率細線化縮小法と任意倍率拡大スムージング法を提案した[68]。この方法は，原理的には，黒崎[59]が示した漢字パターンを比例的に変換する比例法に基づいた手法である。その特徴は，明朝体飾りの種類，付加条件，斜曲線や縦・横線分の幅などに関する漢字パターンの設計規則を一般化し，比例変換の前処理として飾り除去処理，細線化処理，輪郭線補間処理などのパターン処理を設計規則に応じて適用することにより，文字品質の劣化要因を抑制している点にある。

（４）図形発生法の研究

彩色図形の輪郭は線図形で表現される。このため，線図形の品質が彩色図形の品質を支配する要因の一つになるといえる。本論文では，線図形を品質良く表示するための図形発生法を明らかにすることを目的として，線図形の最も基本的な

要素である線分と円弧の発生法を取り上げる。線分と円弧の発生法については，１９６０年代後半から１９７０年代後半にかけて，線分発生の原理に関する研究報告[67]，その円弧発生への拡張に関する研究報告[68]，発生精度や発生速度の改良に関する研究報告[69]-[71]など多く見られる。

１９６５年に，J.E.Bresenham[67]は，画素選択方式による線分発生法の原理を示した。その原理は，２個の選択候補画素のうち実際の線分との距離を評価尺度とし，距離が小さい方の画素を順次選択する考え方に基づいている。１９７７年には，J.E.Bresenham[68]は，この線分発生の原理を円弧発生に拡張した。また，１９７３年には，釜江，小杉，星野[69]は，距離の計算を変位の計算に置き換えることにより，線分と円弧の発生精度を最適化し，かつ，画素選択に要する計算量を削減し高速化した変位比較法を発表した。更に，１９７８年，１９７９年には，末永，釜江，小林[70],[71]は，変位比較法における変位の計算回数が半減できることを示し，変位比較法を高速化へと改良した。しかし，これらの方法は，線図形を白画素と黒画素からなる２値図形として発生する方法だったので，表示図形にジャギと呼ばれる段差が発生し，表示品質が劣化するという問題は残されていた。このため，滑らかな図形を発生させるためのアンチエリアシング法が注目され，１９７０年代後半から１９８０年代初頭にかけて，基本手法[72]やそれらの比較[73]，画素を内挿する方法[74]，線分と重なる画素の面積で輝度を変調する方法[75]-[77]，画素と線分との距離の大きさで画素の輝度を変調する方法[78]などに関する多くの研究報告が見られるようになった。

最初に，１９７７年に，F.C.Crow[72]は，アンチエリアシングの基本手法として，画素密度を高める方法，生成画像に対して輪郭を平滑化する後置フィルタによる方法，画素に面積を表現させ画像生成時に面積値に応じて輝度を変調する前置フィルタによる方法を示した。更に，１９８１年には，F.C.Crow[73]は，これらの比較結果を発表した。

１９７８年には，F.Crow[74]が，中間調の画素を内挿する方法を示した。１９７９年には，H.FuchsとJ.Barros[75]が，画素を正方形の素片とし，発生する線分を面積を持つ多角形とみなして，線分と重なる画素の面積で画素の輝度を変調する方法を発表した。更に，１９８１年には，S.Gupta[76]は，画素を小円形の素片とみなし，面積計算の計算量を削減する方法を提案した。同じく１９８１年

には，西田と中前[77]は，画素と線分との重なり部分の形状の変化を，状態遷移の形で一般化することにより面積計算を簡略化した方法を提案した。また，面積計算によらない方法として，１９８０年に，M.L.V.PittewayとD.J.Watkinson[78]が，Bresenhamの線分発生アルゴリズム[67]における線分と画素との距離的な誤差に応じて，画素の輝度を変調する線分発生法を提案した。このように，図形発生におけるアンチエリアシング法の研究では，性能劣化要因となる面積計算を効率化する試みが多くなされ，線分発生の性能改善が図られてきたが，円弧発生への拡張には制限があった。

これに対し，１９７７年，１９７８年に，筆者は，面積計算や距離計算によらない，線分と円弧のアンチエリアシング法の可能性を示し[79],[80]，１９８０年には，滑らかな線分と円弧を発生する方法として，多値ドット表示法[81]を提案した。この方法は，変位の大きさで画素の輝度を変調する方法であり，これにより計算量を削減させ，高速化と円弧発生への拡張という両条件を同時に満足させた。

１．３　論文の構成

以下に本論文の構成を示す。

２章では，線図形の入力処理法について述べる。彩色図形を対象とした画像ファイルの入力処理は，線図形を入力する過程，線図形を会話的に彩色することにより彩色図形を作成する過程，および，彩色図形を符号化する過程に分けられる。ここでは，線図形の制御を可能とし線図形入力過程における図形品質の向上を目的として，線図形の基本入力処理として細線化処理を取り上げ，処理の経済化と高速化について議論する。処理の経済化については，連続する３走査線分の画素情報を保持させ，画素情報の入力に同期させて冗長な画素を削除する，線順次型処理アルゴリズムについて検討する。処理の高速化については，細線化処理の性能劣化要因となっていた３×３画素マトリクスによる局所操作を，縦線分と横線分のランの連結関係を調べる操作に置き換えた，処理アルゴリズムの構成法を明らかにする。次に，細線化図形の品質を支配する要因の一つである，縦線分と横線分のラン分離パラメータについて定量的に考察する。

3章では，彩色図形の符号化処理法について述べる。ここでは，ラスタ走査型表示装置に表示する条件で，彩色図形の輪郭情報を線分と円弧で表現し，コード情報に変換する図形コマンド符号化法を取り上げ，点列として表現された輪郭情報（ディジタル曲線）から，線分と円弧の特徴を有する部分を抽出する方法について議論する。具体的には，ディジタル曲線に対して線分または円弧を仮定的に当てはめて，各画素からの距離が許容誤差範囲内となる最大の点列を探索する問題として扱った，テンプレートマッチング法に基づいた手法を提案する。また，処理計算量を削減させるため，許容誤差の評価尺度として変位の考え方を導入し，処理計算量の定量的な考察を行う。

4章では，線順次型細線化処理と図形コマンド符号化処理の応用として，彩色図形を作成し符号化して画像ファイルに入力するための画像ファイル作成装置の構成法について検討する。ここでは，線図形をファクシミリから入力し，彩色図形を会話的に作成する方式を取り上げ，線図形入力処理機能の実現法，ハードウェアとソフトウェアの機能分担法を検討し，装置構成法を明らかにする。また，彩色図形の作成時間と彩色図形を表現するために必要となる符号量を評価し，線順次型細線化処理と図形コマンド符号化処理の有効性を示す。

5章では，画像ファイルの出力処理法として，漢字パターンと図形パターンの発生法について述べる。漢字パターン発生法ではサイズ変換処理について検討する。技術的課題として，高速化を達成するため基本原理である比例変換法を基本にする条件で，その品質劣化要因を抑制する方法について議論する。具体的には，漢字パターンの設計規則を集約し，品質劣化要因を抑制するための飾り除去処理，細線化処理，輪郭線補間処理などのパターン処理について考察し，これらのパターン処理を比例変換の前処理として付加した任意サイズ漢字パターン発生法を提案する。図形パターン発生法では，2値表示の線分と円弧の発生法における表示品質上の問題を解決するため，線分と円弧のアンチエリアシング法について検討する。技術的課題として，高速化と円弧発生への拡張性を取り上げ，面積計算や距離計算によらない，変位を用いた輝度の変調原理を示し，線分と円弧が同じ原理で発生できることを明らかにする。また，処理計算量や輝度変調パラメータと処理結果の関係などについて考察する。

6章では，本研究の成果をまとめると共に，今後に残された課題を整理する。

第２章　線図形入力処理法

２．１　入力処理方式

線図形を入力し彩色して彩色図形を作成する場合，線図形の線幅の不揃いの修正や表罫線の修正，着色した彩色図形から不要な縁どり線の除去など，線図形の制御機能が必要となる。線図形を１画素幅に細める細線化処理により，これらの制御機能を実現することを考える。細線化処理を彩色図形作成に適用する場合には，（１）処理の高速化，（２）彩色処理などの会話処理との並列処理，（３）図形の連結性の保証，などの条件を満足する必要がある[82]。（１）と（２）の条件は，線図形入力時間を削減し画像ファイル作成時間を短縮するための条件である。（３）の条件は，画素同士の連結関係が欠落することによる着色時の色漏れ現象を防止するための条件である。

図形の細線化法としては，１画面分の画素データを一旦メモリに蓄積したのち，ラスタ走査を繰り返し線縁から順次画素を削除し，図形の中心線を４連結図形または８連結図形[83]として抽出する方法が一般的である。しかし，この方法は処理に１画面分のメモリを要するので，経済的でない。また，３×３の画素間で論理演算を行うマスク走査により削除画素を決定していたので，画素データの参照回数が膨大となり，処理の高速化を阻害していた。これらの問題を解決するため，ファクシミリなどの順次走査型装置から走査データを得て実時間的に細線化を行う線順次型処理方式について考察する。

線順次型細線化法としては，連続する走査線の画素情報を保持する走査メモリを，線図形の最大線幅を越える限られた数だけ持たせ，３×３画素マスク走査に基づいて細線化する方法[32],[36]がある。しかし，これらの方法は，３×３画素間の論理演算により削除画素を決定していたので，処理時間がかかるという問題があった。本研究では，連続する３走査線の画素情報のみを保持する走査メモリを持たせ，黒画素ランの連結関係から削除画素を決定し，４連結の細線化図形に変換する線順次型処理法[41],[82]を提案し，任意形状の線図形を対象とした線順次型細線化処理法と，表罫線を対象とした線順次型水平垂直線化処理法の処理アルゴリズムの構成法について考察する。

2.2 基本的な考え方

（1）問題点の所在

線図形入力では図形の大きさ，線の長さといった幾何学的特徴を保つことが重要である。このためには，輪郭から線の中心に向かって不要な画素を削除することが必要となる。縦線と横線では細める方向が90°異なるので，縦線と横線を誤認識すると図形の縮退現象が起こる。このため，線順型次処理方式では走査毎に処理が完結するので，各走査の画素列から実時間的に縦線と横線を識別し，削除画素を決定することが課題となる。

（2）処理方式とラインメモリ

線順次処理を考えるため，図形の1画面分を蓄積するための画面メモリは設けない。ある画素とその隣接画素との連結関係を調べることが必要となるので，3走査線分のメモリを設け，これらをラインメモリと呼ぶ。ラインメモリの画素情報を以下のように表す。

$\mathbf{a}_{i-1}$：処理された画素情報

$\mathbf{a}_{i}$　：処理しようとする走査線の画素情報

$\mathbf{a}_{i+1}$：最新の走査線の画素情報

なお，処理の進行に従い，$\mathbf{a}_{i-1}$の内容が出力され，1時刻後には，

$\mathbf{a}_{i} \rightarrow \mathbf{a}_{i-1}$

$\mathbf{a}_{i+1} \rightarrow \mathbf{a}_{i}$

のようにラインメモリの内容が更新され，$\mathbf{a}_{i+1}$が新たな走査線の画素情報となる。

（3）諸定義

ファクシミリなどのラスタ走査型装置の入力形式に対応し，第i番目の走査線上の第j番目の画素をα_{ij}と表し，

黒画素のとき，$\alpha_{ij}=$"1"

白画素のとき，$\alpha_{ij}=$"0"

とし，以下の定義を行う。

［定義2.1］ 黒画素の連なりをランと呼び，第i番目の走査線上の1つのランを，

$$r_i = \{\alpha_{ij}\},$$

と表し，ランの右端画素位置と左端画素位置を，それぞれ，

$$mrp(r_i) = max\{j \mid \alpha_{ij} \in r_i\},$$

$$mlp(r_i) = min\{j \mid \alpha_{ij} \in r_i\},$$

と定義する。■

［定義2.2］ 2つのランr，r′に対して，

$$|i - i'| + |j - j'| = 1 \qquad (2.1)$$

を満足する黒画素$\alpha_{ij} \in r$，$\alpha_{i'j'} \in r'$が存在するとき，rとr′は連結しているという（図2.1）。■

［定義2.3］ ランr_iとr_k（$k = i \pm 1$）の論理積（∧）と論理和（∨）を，

$$r_i \wedge r_k = \{\alpha_{ij} \wedge \alpha_{kj} \mid \alpha_{ij} \in r_i,\ \alpha_{kj} \in r_k\},$$

$$r_i \vee r_k = \{\alpha_{ij} \vee \alpha_{kj} \mid \alpha_{ij} \in r_i,\ \alpha_{kj} \in r_k\},$$

と定義する。■

［定義2.4］ 同一画面内に存在する互いに連結するラン群を類別することができ，各類の各々を連結成分と呼ぶ。このうち，連結成分が閉ループを形成しない場合には単連結成分と呼び，閉ループを少なくとも1つ形成する場合には多重連結成分[83]と呼ぶ。■

また，線図形が垂直方向（副走査方向）に近づくにつれ，その図形を横切る走査線上のランの長さは減少するので，線幅がほぼ一様な図形に対しては，ランの長さによって縦線か横線の識別が可能になる。そこで，ランの種別と連結関係に関して，以下の定義を行う。

［定義2.5］ 1つのランrの長さ（$\|r\|$と表す）を，あるしきい値Tと比較し，

$\|r\| < T$のとき，縦線要素，

$\|r\| \geq T$のとき，横線要素，

と呼び，それぞれの集合をV_T，H_Tと表し，それぞれを縦線成分，横線成分と呼ぶ。■

［定義2.6］ ランr_i（$\in a_i$）に対するランの連結関係を，次の4種の状態に分類し，連結状態1～4と呼ぶ（図2.2）。

（連結状態1）ランr_iに連結するランが存在しない場合。

（連結状態2）ランr_iに連結するランがa_{i+1}にのみ存在する場合。

（連結状態３）ランr_iに連結するランがa_{i-1}とa_{i+1}に存在する場合。

（連結状態４）ランr_iに連結するランがa_{i+1}にのみ存在する場合。 ■

２．３　線順次型細線化処理法

２．３．１ 予備的考察

細線化処理は，図形のトポロジカルな性質を保存しつつ黒画素を消去し図形を１画素幅の線図形に変換する処理である。図形のトポロジカルな性質を表す量としてオイラー数があり，連結成分の個数と閉ループの個数の差で定義される[84]。細線化処理において黒画素の過剰消去を行うと，線の途切れや穴の発生という現象を引き起こし，その結果として連結成分や閉ループの個数が変化し，図形のトポロジカルな性質が変わってしまう。このことから，消去しても図形のトポロジカルな性質を変えない黒画素を，消去可能画素と呼ぶ[83]。

すなわち，図形を細線化する場合，消去可能画素のみを消去することにより，線の途切れや穴の発生を防止して図形のトポロジカルな性質を保存することが重要となる。そこで，ランを単位とした処理を可能にするため，定義２．６で示した連結状態におけるランの消去可能性について考察する。

（１）連結状態１

連結状態１のランr_iは，それ自身で１つの連結成分をなしている。このため，r_iを消去すると連結成分の個数が減少するので，r_iは消去可能でない。

（２）連結状態２（連結状態４）

連結状態２（連結状態４）のランr_iの消去可能性は，r_iに連結するa_{i+1}（a_{i-1}）のランの個数Ｎに依存する。

Ｎ＝１の場合：r_iは１つの連結成分のエッジに位置するランであり，消去しても連結成分と閉ループの個数に変化を与えないので，消去可能である。

Ｎ＞１の場合：r_iを消去すると明らかに線の途切れが生じ，連結成分と閉ループの個数が変動するので，消去可能でない。

（３）連結状態３

連結状態３のランr_iは，r_iに連結するランがa_{i-1}とa_{i+1}に存在するので，明らかに消去可能でない。

２．３．２　アルゴリズム

図形を１画素幅に細線化するためには，ランの消去可能性に基づいた処理では十分でなく，ランを構成する画素に着目したきめ細かい処理が必要となる。ここでは，ランの種別と連結状態により，画素単位に消去する処理アルゴリズムについて考察する。

以下，連結状態１～４に対応する処理手順を手順１～４と呼ぶこととする。

（１）連結状態１

連結状態１のランは消去可能でないが，孤立した縦線要素は雑音とみなす。

［手順１］

if $\| r_i \| < T$, then {雑音とみなして消去する}

else {横線とみなして残す} ■

（２）連結状態２

連結状態２は図形の開始部で出現する。そこで，縦線成分と横線成分の開始部を，以下のように定義する。

［定義2.7］ 連結状態２において，連結するラン（r_iとr_{i+1}）が共に縦線要素（または，横線要素）のとき，ランr_iを縦線成分（または，横線成分）の開始部と呼ぶ。 ■

［手順２］

if {$\| r_i \| \geq T$ and $\| r_{i+1} \| \geq T$},

then {r_iを消去する}

else if {$\| r_i \| \geq T$ and $\| r_{i+1} \| < T$},

then {r_iを横線とみなして残す}

else if {$\| r_i \| < T$ and $\| r_{i+1} \| \geq T$},

then {r_iは横線要素r_{i+1}の輪郭雑音とみなして消去する}

else if {$\| r_i \| < T$ and $\| r_{i+1} \| < T$},

then {r_iは縦線成分の開始部とみなし r_{i+1}に連結する中央部の１画素x_{ij}を残し，他の画素を消去する。

但し，

$j = [\{mlp(r_i \wedge r_{i+1}) + mrp(r_i \wedge r_{i+1})\} / 2]$. } ■

但し，［　］はガウス記号（以下同様）。

（３）連結状態３

連結状態３は図形の中間部に出現する状態である。ここで，ラインメモリの内容a_{i-1}，a_{i+1}のランのうち，ランr_iに連結するランをそれぞれr_{i-1}，r_{i+1}とし，$\|r_i\| < T$の場合の処理手順を手順３．１，$\|r_i\| \geq T$の場合の処理手順を手順３．２と呼ぶ。

［手順３．１］（$\|r_i\| < T$の場合）

$r = r_i \wedge r_{i-1} \wedge r_{i+1}$とする。

if ｛$r \neq \phi$｝，

then ｛共通部の中央の１画素α_{ij}のみを残す。

但し，

$j = [\{mlp(r) + mrp(r)\} / 2]$．｝（図２．３（ａ））

else if ｛$r = \phi$｝，then ｛r_{i-1}の画素のうち r_iに連結する端部の画素位置をＰ，r_iとr_{i+1}の共通部中央の画素の位置をＱとし，Ｐ～Ｑの範囲内の画素だけを残す。｝（図２．３（ｂ））　■

［手順３．２］（$\|r_i\| \geq T$の場合）

$r = r_i \wedge r_{i-1} \wedge r_{i+1}$とする。

if ｛$\|r_{i-1}\| < T$ and $\|r_{i+1}\| \geq T$｝，

then ｛r_{i-1}との連結性を保つため，$r_i \leftarrow r_{i-1} \wedge r_i$とする。

但し，←は代入を表す。｝（図２．４（ａ））

else if ｛$\|r_{i-1}\| < T$ and $\|r_{i+1}\| < T$｝，

then ｛連結する横線要素がないので 横線成分の最下端の要素とみなし消去しない。｝（図２．４（ｂ））

else if ｛$r = \phi$｝，then ｛r_{i-1}の画素のうち r_iに連結する端部の画素位置をＰ，r_iとr_{i+1}の共通部中央の画素の位置をＱとし，Ｐ～Ｑの範囲内の画素だけを残す。｝（図２．４（ｃ））　■

手順３．１と手順３．２は連結状態３における基本的な処理手順である。連結状態３は，図形の中間部に現れ最も出現頻度が高い連結状態であるので，一般的な接続関係の場合の処理について考察する。ここで，以下の定義をする。

［定義2.7］ ラインメモリ $\mathbf{a}_j$（$j = i \pm 1$）に存在し，かつ，ラン $\mathbf{r}_i$ に連結するランのうち，左端画素位置（または右端画素位置）が最小のランを左端ラン，最大のランを右端ランと呼び，それぞれ，$\mathbf{r}_j^L$，$\mathbf{r}_j^R$ と表す。 ■

連結状態3において，ラン $\mathbf{r}_{i-1}^L$ と $\mathbf{r}_{i-1}^R$ の間に存在するラン $\mathbf{r}_{i-1}$ は，手順2，手順3．1および手順3．2により，共通部の中央画素以外は消去されて1画素長になっているので，次の手順を得る（図2．5）。

［手順3．3］

$P_0 = \mathrm{mrp}(\mathbf{r}_{i-1}^L)$

$Q_0 = \mathrm{mlp}(\mathbf{r}_{i-1}^R)$

$P_1 = \mathrm{mrp}(\mathbf{r}_{i+1}^L)$

$Q_1 = \mathrm{mlp}(\mathbf{r}_{i+1}^R)$

$P \leftarrow P_0$

$Q \leftarrow Q_0$

if $\{P_1 < P_0$ and $\| \mathbf{r}_{i-1}^L \| = 1\}$, then $\{P \leftarrow P_1\}$

if $\{P_1 < P_0$ and $\| \mathbf{r}_{i-1}^L \| \geq 2\}$, then $\{X \leftarrow P_1\}$

if $\{Q_1 > Q_0$ and $\| \mathbf{r}_{i-1}^R \| = 1\}$, then $\{Q \leftarrow Q_1\}$

if $\{Q_1 > Q_0$ and $\| \mathbf{r}_{i-1}^R \| \geq 2\}$, then $\{Y \leftarrow Q_1\}$

以上の計算で得られたP，Q，X，Yに対して，$\mathbf{r}_i$ の画素のうち，$P \leq j \leq Q$ または $j = X$ または $j = Y$ を満足する画素 $\alpha_{i,j}$ 以外の画素を消去する。 ■

（4）連結状態4

連結状態4は図形の最下端に出現する状態であり，ラン $\mathbf{r}_i$ に連結するランがラインメモリ $\mathbf{a}_{i+1}$ に存在しないので，次の手順を得る。（図2．6）

［手順4］

if $\{\| \mathbf{r}_i \| \geq T\}$,

then {横線成分の最下端要素とみなし消去しない。}

else if $\{\| \mathbf{r}_i \| < T$ and $\| \mathbf{r}_{i-1} \| \geq T\}$,

then {$\mathbf{r}_i$ を $\mathbf{r}_{i-1}$ の輪郭雑音とみなして消去する。}

else if $\{\| \mathbf{r}_i \| < T$ and $\| \mathbf{r}_{i-1} \| < T\}$,

then {縦線成分の最下端要素とみなし，$\mathbf{r}_{i-1}$ に連結する1画

素以外を消去する。}　■

2．3．3　処理計算量の評価

細線化処理は，消去可能画素を1画素ずつ削除して図形を細める処理である。このため，図形パターンを格納したメモリの画素データへのアクセス回数が，計算量を支配する最も大きな要因となる。ここでは，画素データへのアクセス回数の観点から，3×3画素のマスク走査を基本とする手法[37]と比較することにより，提案アルゴリズムの計算量を評価する。

いま，3×3画素のマスク走査を基本とする手法と提案手法における画素データへの走査線当りのアクセス回数を，それぞれS_M，S_Lとすると，計算量の相対比ρは，

$$\rho = \frac{S_L}{S_M} \qquad (2.2)$$

と表される。

ここで，図形パターンを（M行×N列）の2値画素パターンとし，行を走査線に対応させ，第i行に存在する黒画素ランを$r_i(1), r_i(2), \cdots, r_i(N_i)$と仮定して，$S_M$と$S_L$を求める。3×3画素のマスク走査を基本とする手法では，各行（走査線）における画素データへのアクセス回数は，黒画素か白画素かを判定するためのアクセス回数と，各黒画素においてマスクを作用させるためのアクセス回数の合計で近似できる。また，1回のマスクの作用に5回のアクセスを要する[37]。　したがって，S_Mは式（2．3）により表すことができる。

$$S_M = k \cdot \{ N + 5 \cdot \sum_{j=1}^{N_i} \| r_i(j) \| \} \qquad (2.3)$$

但し，kは消去可能画素がなくなるまでの走査回数を表す。

また，提案アルゴリズムでの走査線当りの画素データへのアクセス回数は，黒画素を判定しランを検出するためのアクセス回数と，前後の走査線上の連結ランを検出するためのアクセス回数の合計で近似できる。

したがって，S_Lは式（2．4）で表すことができる。

$$S_L = N + \sum_{j=1}^{N_i} \{ \omega_{i-1}(r_i(j)) + \omega_{i+1}(r_i(j)) \} \qquad (2.4)$$

但し，$\omega_{i-1}(\mathbf{r}_i(j))$ と $\omega_{i+1}(\mathbf{r}_i(j))$ は，それぞれ，第（ｉ－１）走査線と第（ｉ＋１）走査線においてラン $\mathbf{r}_i(j)$ に連結するランを検出するために要する画素データへのアクセス回数を表す。

故に，計算量の相対比 ρ は式（２．５）となる。

$$\rho = \frac{1}{k} \cdot \frac{N + \sum_{j=1}^{N_i} \{\omega_{i-1}(\mathbf{r}_i(j)) + \omega_{i+1}(\mathbf{r}_i(j))\}}{N + 5 \cdot \sum_{j=1}^{N_i} \| \mathbf{r}_i(j) \|} \qquad (2.5)$$

ここで，簡単のため，走査線に直交するＬ本の線幅Ｗ［画素］の縦線からなる図形パターンのモデルを考えると，

$$N_i = L$$

$$\| \mathbf{r}_i(j) \| = W$$

$$\omega_{i-1}(\mathbf{r}_i(j)) = \omega_{i+1}(\mathbf{r}_i(j)) = W$$

となり，マスク走査による手法では１回の走査で左右の輪郭を１画素だけ削除するので，走査の繰り返し回数は，ほぼＷ／２に等しくなる。したがって，計算量の相対比 ρ は，

$$\rho = \frac{1}{(W/2)} \cdot \frac{N + 2LW}{N + 5LW} \qquad (2.6)$$

となる。また，線図形に対しては，黒画素が占める比率は極めて小さいことから，$N \gg LW$ と考えてよく，計算量の相対比 ρ はほぼ２／Ｗとなることが分かる。

Ｍ＝Ｎ＝２０４８［画素］，線幅が約５画素の任意線図形に対して，ＦＯＲＴＲＡＮプログラムでシミュレーションを行ったところ，マスク走査による手法と提案手法の１走査線当りの処理時間は，それぞれ，２．５[sec]，１．３[sec]であり，ほぼ理論値に近い結果が得られた。

なお，プログラム規模は，マスク走査による手法で約１５０[step]に対して，提案手法では約１．４[Kstep]であり，約１０倍を要した。

２．３．４　処理結果と評価

図２．７に，提案アルゴリズムによる図形（５１２×５１２画素）の細線化例

を示す。処理時間は，ＦＯＲＴＲＡＮプログラムで平均２００ｍｓ／ラインであった。

以下に，アルゴリズムを評価するため，図形の縮退，短線分の発生，歪現象について考察する。

（１）図形の縮退

提案手法では，ランの連結関係と消去可能性を考慮して消去画素を決定しているので，線の切断は起こらない。しかし，パラメータＴによって縦線要素と横線要素を識別しているので，Ｔが縦線の幅より小さい場合には縦線要素が横線要素に判定され，手順２において消去可能ラン（横線要素）が順次消去されることになり，図形が縮退する。これに対しては，２．５節においてパラメータＴの決定法について考察し，図形の縮退を防止できることを示す。

（２）短線分の発生

図２．８に示すように，鋭角図形を細線化すると図形の角部に短線分ＡＢが発生する。この原因は，パラメータＴとの関係から角部が縦線成分として識別されるためである。この現象はパラメータＴを小さくすることで抑制できるが，Ｔを小さくし過ぎると図形の縮退を引き起こすという問題がある。短線分の発生現象は，他の細線化アルゴリズムにおいても持っている一般的な問題であり，今後の課題といえる。

（３）歪現象

細線化図形の歪現象は，抽出される画素の位置の歪により引き起こされる。提案手法では，縦線成分に対してはほぼ中央の画素が抽出されるが，手順２および手順３．２における横線要素に対する処理の特徴から，図形の分岐部と横線成分において歪が発生し易い。

すなわち，図２．９に示すように，分岐部Ａでは線順次処理のため分岐の状態を認識することができず，輪郭の分岐部Ｂではじめて分岐の状態を認識できることから，分岐部が押しつぶされた図形となる。この現象は図形の交差部においても起こり得るが，線幅が線画像サイズに比べて比較的小さい図形を対象としているので，大きな問題とはならない。また，手順２において横線成分の最下端の画素を残すこととしているため，図２．１０に示すように処理結果が副走査方向に細長くなるが，その歪の度合は原画の線幅以上になることはないので，細線化結

果の自然さに大きな影響を与えるものではない。

2.4 線順次型水平垂直線化処理法

2.4.1 予備的考察

画像情報の検索サービスなどで多く使用される表やブロック図を作成するには，水平線や垂直線をきれいに入力する必要がある。細線化処理は図形の輪郭雑音や図形の変形に影響を受け易いため，この目的には適さない。ここでは，正確な水平線と垂直線を自動的に入力することを目的とした，線順次型水平垂直線化処理法について考察する。

水平線と垂直線を正確に入力するためには，縦線成分に対しては縦線要素を同一走査線上の1画素に置き換え，横線成分に対しては連結する横線要素を統合して1つの横線要素に置き換える処理が基本となる。この処理を線順次型処理で実現するためには，縦線要素を1画素に置き換えることから，垂直線の位置，垂直線と縦線要素および横線要素との連結関係を保存することが必要となる。

例えば，図2.11に示すようにランr_iの連結状態に着目するとランr_{i-1}の処理前後で連結状態3から連結状態2に変化しており，r_iとr_{i-1}の連結関係は保存されていない。

そこで，垂直線の位置と連結関係を保存するバッファメモリを補助メモリと呼び，走査線上の画素位置jに対応させた各要素（z_jと記す）に以下の意味をもたせる（図2.12）。

① $z_j = 2$のとき，jは垂直線の位置。

② $z_j = 1$のとき，jからjが増加する方向（主走査方向）に垂直線が存在する。

③ $z_j = -1$のとき，jからjが減少する方向（主走査の逆方向）に垂直線が存在する。

ここで，定義2.2におけるランの連結関係を，以下のように拡張する。

［定義2.8］ 任意のランr_iに対して，r_iが$z_j \neq 0$となる画素α_{ij}（＝1）を含むとき，r_iは垂直線に連結しているという。 ■

水平垂直線化処理は，縦線成分と横線成分を線幅が1画素の垂直線と水平線に

変換する処理である。 このため，定義２．８より３走査線分のラインメモリと補助メモリによりランの連結関係を保存でき，線順次型処理により実現できる。

２．４．２　アルゴリズム

細線化処理と同様，連結状態１～４に対応する処理手順を手順１～手順４と呼ぶ。

（１）連結状態１

孤立した縦線要素は，消去可能でないが輪郭雑音とみなして消去する。横線要素に対しては，垂直線との連結性から画素の削除と追加を制御する。

［手順１］

if ｛‖ r_i ‖＜Ｔ｝，

then ｛輪郭雑音とみなして消去する。｝

else if ｛r_iが垂直線に連結していない｝，

then ｛r_iは水平線とみなし消去しないで残す。｝

else if ｛r_iが垂直線に連結している｝，

then ｛r_iが連結している垂直線の位置の最小値と最大値をそれぞれ v_{min}，v_{max}とし，$P_i = mlp(r_i)$，$Q_i = mrp(r_i)$とする。

このとき，

$P'_i = min(v_{min}, P_i)$

$Q'_i = max(v_{max}, Q_i)$

とし，$P'_i \leq j \leq Q'_i$なる j に対して，$\alpha_{ij} = 1$とする。｝ ■

（２）連結状態２

縦線要素と横線要素に対する処理手順を，それぞれ手順２．１と手順２．２と呼ぶ。

［手順２．１］（‖ r_i ‖＜Ｔの場合）

if ｛（r_iが垂直線に連結していない）and（‖ r_{i+1} ‖≧Ｔ）｝，

then ｛r_iは横線要素 r_{i+1}の輪郭雑音とみなして消去する。｝

if ｛（r_iが垂直線に連結していない）and（‖ r_{i+1} ‖＜Ｔ）｝，

then ｛垂直線の開始部とみなし，r_iの中央の画素 α_{iv}のみを

残し，それ以外の画素を消去する。このとき $\mathbf{r}_i$ に対して，補助メモリの内容を，以下のように設定する。

$z_j = 2$ for $j = v$,

$z_j = 1$ for $mlp(\mathbf{r}_i) \leqq j < v$,

$z_j = -1$ for $v < j \leqq mrp(\mathbf{r}_i)$.

但し，$v = [\ \{mlp(\mathbf{r}_i) + mrp(\mathbf{r}_i)\} / 2\ ]$ }

if {$\mathbf{r}_i$ が垂直線に連結している}，

then {$\alpha_{iv} = 1$ とし，α_{iv} を除く $\mathbf{r}_i$ の画素を消去する。但し，v は垂直線の位置を表す。このとき，

$P'_i = \min(P_i, v)$，$Q'_i = \max(Q_i, v)$ とし，補助メモリの内容を以下のように設定する。

$z_j = 2$ for $j = v$,

$z_j = 1$ for $P'_i \leqq j < v$,

$z_j = -1$ for $v < j \leqq Q'_i$.

但し，$P_i = mlp(\mathbf{r}_i)$，$Q_i = mrp(\mathbf{r}_i)$。} ■

［手順２．２］（$\| \mathbf{r}_i \| \geqq T$ の場合）

if {$\| \mathbf{r}_{i+1} \| \geqq T$ となる連結ランが少なくとも一つ存在する}，

then {$\alpha_{i+1,j} = 1$ for $P'_{i+1} \leqq j \leqq Q'_{i+1}$. 但し，

$P'_{i+1} = mlp(\mathbf{r}_i \vee \mathbf{r}_{i+1})$，

$Q'_{i+1} = mrp(\mathbf{r}_i \vee \mathbf{r}_{i+1})$.

次に，$\mathbf{r}_i$ を消去すると共に，垂直線との連結性を保つため，

$z_j = 2$ ($P_i \leqq j \leqq Q_i$) なる位置の画素 $\alpha_{ij} = 1$ とする。

但し，$P_i = mlp(\mathbf{r}_i)$，$Q_i = mrp(\mathbf{r}_i)$ である。}

else {ラン $\mathbf{r}_i$ に連結する垂直線の位置の最小値と最大値をそれぞれ，

v_{min}，v_{max} とし，$\mathbf{a}_{i+1}$ のランのうち $\mathbf{r}_i$ に連結するランの中央画素位置の最小値と最大値を v'_{min}，v'_{max} とする。

このとき，

$\alpha_{ij} = 1$ for $P'_i \leqq j \leqq Q'_i$.

但し，

$P'_i = \min(P_i, v_{min}, v'_{min})$，

$$Q'_t = \max(Q_t, V_{max}, V'_{max}).\ \}$$ ■

（３）連結状態３および連結状態４

a_{t-1}のランr_{t-1}とa_tのランr_tとの連結関係は，補助メモリに保存されていることから，手順３は手順２．１または手順２．２と，手順４は手順１と同様のアルゴリズムとなる。

２．４．３　処理結果と評価

図２．１３に，提案手法による手書き線図形の水平垂直線化処理例を示す。画像サイズは（５１２×５１２）画素である。本処理では，ラインメモリのほかに１走査線に相当する補助メモリを必要とするが，手順に示したようにラインメモリa_{t-1}の処理を補助メモリの処理で置き換えているので，画素データへのアクセス回数は細線化に比べて大きく増加せず，処理時間は細線化処理とほぼ同様の結果が得られた。

以下に，アルゴリズムを評価するため，図形の縮退，歪現象，および斜線処理の考察結果を述べる。

（１）図形の縮退

本手法では，ランの連結関係を補助メモリ内の情報で保存しているので，図形の切断現象は発生しない。一方，パラメータＴの設定値によっては細線化処理と同様，図形の縮退現象を引き起こす。パラメータＴの決定法は，２．５節で考察する。また，本処理は手書き図形の自動清書を目的としているため，手順１において線のはみ出し部分を意識的に消去し，短線分の発生を防止している。

（２）歪現象

本手法では，図形を水平線と垂直線に規格化するので，細線化で問題になった分岐部での歪現象は発生しない。一方，入力用紙の傾きにより水平方向と垂直方向の歪が大きくなるが，傾き補正処理と組み合わせることにより実用上の問題とはならない。

（３）斜線処理

本手法による処理結果には斜線が現れない。しかし，表の枠組みや流れ図などでは斜線使用の要求が当然考えられ，斜線処理への拡張は今後の課題である。

2.5 線分識別しきい値

2.5.1 評価関数

1次元ランレングスの出現分布の性質については，ファクシミリ信号の統計的性質としてよく論じられている[85]。文字や図形からなる通常文書に対するランの出現分布には，

（1）長いランの出現頻度は，短いランの出現頻度に比べて極めて小さい。

（2）局所的なピークを有し，そのピークはランの短い領域に偏る。

という定性的特徴があり，ピーク近傍の長さを持つランが縦線要素であることが推察できる。

ここでは，より正確にしきい値Tを推定するための評価関数について考察する。

細線化処理と水平垂直線化処理では線幅がほぼ一様な線図形を対象にしているので，しきい値Tの選び方が良ければ識別された縦線成分と横線成分の平均的な線幅は等しい。そこで，両成分の線幅が最も接近するように，しきい値Tを求めるため，評価関数を以下のように定義する。

［定義2.9］ しきい値Tで識別された縦線成分と横線成分の線幅の平均値を，それぞれ$W_v(T)$，$W_h(T)$と表し，評価関数$F(T)$を，

$$F(T)=|W_v(T)-W_h(T)| \qquad (2.7)$$

と定義する。 ■

定義より，$F(T)$が0に近いほどよいことになる。すなわち，$F(T)$を最小化するT求めればよい。

ところで，縦線成分と横線成分の線幅の平均値$W_v(T)$と$W_h(T)$は，それぞれ式（2.8）と（2.9）のように表すことができる。

$$W_v(T)=\text{（縦線要素の平均長）}$$

$$=\{\Sigma_{r\in V_T}\|r\|\}/n_v(T) \qquad (2.8)$$

但し，$n_v(T)$は縦線成分V_Tの要素の総数を表し，$V_T=\phi$の場合は，$W_v(T)=0$とする。

$W_v(T)$＝（横線成分に含まれる平均要素数）

$$= n_h(T) / N_h(T) \qquad (2.9)$$

但し，$n_h(T)$は横線成分H_Tの横線要素の総数，$N_h(T)$は横線成分の個数を表し，$H_T = \phi$のとき$W_h(T) = 0$とする。

したがって，評価関数として

$$F(T) = \left| \sum_{r \in V_T} \| r \| / n_v(T) - n_h(T) / N_h(T) \right| \qquad (2.10)$$

を得る。

2.5.2 評価関数の計算法

しきい値Tに対する評価関数の値を算出する場合，横線成分の個数$N_h(T)$を求める必要がある。

2値図形のトポロジカルな性質を表す量としてオイラー数[84]がある。その性質として，単連結成分に対してはオイラー数を成分の個数を表す量とすることができるが，しきい値が小さいためすべてのランが横線要素になり，横線成分が多重連結になる場合には，オイラー数は0以下になることがあるので，オイラー数を成分の個数を表す量として用いることは適当でない。

そこで，図形を走査し第i番目の走査で得られるランの個数をN_i，このうち第（i－1）番目の走査線上に連結するランを有するランの個数をN^*_iとし，

$$C = \Sigma (N_i - N^*_i)$$

とする。

Cは直前の走査線上に連結するランを有しないランの総数（すなわち，連結状態1と2のランの総数）を意味しているので，1つの連結成分に対して$C \geqq 1$となる。

これより，連結成分の個数NとCの間に，$N \leqq C$が成立することから，しきい値Tで識別された横線成分の個数$N_h(T)$をCの値（C_Hと表す）で近似することにより，

$$F(T) = \left| \sum_{r \in V_T} \| r \| / n_v(T) - n_h(T) / C_H \right| \qquad (2.11)$$

を得る。

２．５．３　線分識別結果としきい値の推定

図２．１４に，評価関数の算出例と評価関数を最小にするしきい値で縦線成分と横線成分を分類した結果を示す。"Ｖ"は縦線成分を，"Ｈ"は横線成分を表し，良好な分類結果が得られることが分かった。

図２．１５は，約３０種の手書きパターン（３２画素×３２画素）を入力し，各パターンについて評価関数を求め，評価関数を最小にするしきい値とそれに対する縦線成分と横線成分の平均幅との関係を求めたものである。これより，最適なしきい値T_{opt}は縦線成分の１．２倍～２．５倍の間に存在することが分かる。

以上のことから，走査線密度をｄ（本／ｍｍ），入力図形の線幅をｗ（ｍｍ）としたとき，しきい値Ｔを１．２ｄｗ～２．５ｄｗに設定すればよいことを示している。

２．６　むすび

線図形の入力を目的にした，線順次型細線化法と線順次型水平垂直線化法を提案した。本手法は，ファクシミリなどの順次走査形装置から走査データを得て，縦線分と横線分のランを識別し，それらの連結関係を調べて処理を順次進行させる実時間入力処理である。その特長は，処理に要するメモリを節減でき，ラン単位で処理を行い走査線単位で処理が完結するので，処理時間の短縮を図れる点にある。

実験で得た走査線当りの処理時間は，ＦＯＲＴＲＡＮプログラムで平均２００ｍｓであり，ファクシミリへ十分適用可能である。また，本手法による処理結果は横線要素と縦線要素の分類しきい値に依存する。図形の切断や縮退がない良好な処理結果を得るための，しきい値の推定法を示し，彩色図形の入力処理に必要な線図形入力処理に対する要求条件を満足させた。

第３章　図形コマンド符号化法

３．１　符号化方式

符号化処理は，パターン情報として入力された彩色図形から，通信や蓄積において真に必要となる情報を抽出する過程とみなすことができる。本研究では，彩色図形から輪郭情報を画素列として抽出し，線分と円弧で表現する図形コマンド符号化法[57],[58]について検討する。このため，画素列から線分や円弧の特徴を有する部分を抽出する方法を，技術的課題として取り上げて議論する。

画素列から線分成分や円弧成分を抽出する問題は，パターンの構造解析のため画素列の近似問題として多くの研究がなされ，近似多角形の周長，線分個数，線分長などの最小化あるいは最大化問題として扱った方法[44],[46]-[48],[50]や，図形形状の周波数解析に根源を持つ方法[49]などが提案されている。前者の方法は，原理的に線分成分を抽出する問題を扱った方法であるといえる。また，後者の方法は，直流成分を線分成分としそれ以外を曲線成分とする考え方に基づいた方法であるといえる。

本研究では，画素列から線分や円弧の特徴を有する最大長の点列を探索する方法[56]を提案する。本手法は，パターン認識の最も基本的な手法であるテンプレートマッチング法に基づいた手法であり，線分あるいは円弧に対するテンプレートの表現法，画素列とテンプレートとの一致度の判定法，最大長点列の探索法について考察する。テンプレートの表現法については，線分または円弧を中心線とし一様な幅を持つ，チューブと呼ぶ可変長の帯状図形をテンプレートとして定義する。テンプレートとの一致度判定法は，画素がテンプレートの領域に含まれるかどうかを判定する，変位比較に基づいた判別式として定式化する。また，最大長点列の探索法については，画素列に対してテンプレートを２分探索の手法に基づいて適用することにより，探索が効率的に行えることを示し，画素列から線分や円弧の特徴を有する最大長の点列を抽出するアルゴリズムを明らかにする。更に，動的ステップ数と図形データ圧縮効果の評価を行い，提案手法の有効性を示すと共に，図形コマンド符号化法の実現性を明らかにする。

3.2 符号化手順[58]

彩色図形の符号化処理は，符号化結果を解釈し表示する復号化処理と独立ではない。すなわち，同一色の領域を閉曲線で符号化する方法では，復号化時に閉曲線を発生しその内側を指定色で埋める処理が必要となる。また，ラスタ走査方向の色の変化点を曲線（開曲線となる）で符号化する方法では，色の変化点と変化点の間をラスタ走査方向に同一色で埋める処理が必要となる。後者の方法はランレングス符号[86]の復号化処理と等価となり，前者の方法に比べ高速な復号化が可能となるという利点がある[87]。このため，ランレングス符号から色の変化点の画素列を得て，それを線分と円弧で表現することにより符号化する方法[58]について検討する。

図形コマンド符号化の処理手順は，図3.1に示すように，色の変化点の画素列を抽出する段階，抽出した画素列を線分と円弧で近似する段階，近似結果を図形コマンドに変換する段階に分けて考えることができる。各段階の概要を以下に述べる。

（1）画素列の抽出

第i走査線上のn番目のランを$r_i(n)$と表すと，［定義2.1］からその開始アドレスは，$mlp(r_i(n))$と表される。

まず，$r_i(n)$と同一色符号を有し，次の条件，

$$mlp(r_{i+1}(m)) \leqq mlp(r_i(n)) < mlp(r_{i+1}(m+1))$$

または，

$$mlp(r_i(n)) \leqq mlp(r_{i+1}(m)) < mlp(r_i(n+1))$$

を満足する第（i＋1）番目の走査線のラン$r_{i+1}(m)$を検出する。次に，これらのうち開始アドレスが最小のもの，すなわち

$$j' = min\{ mlp(r_{i+1}(m)) \}$$

となるランを求め，画素（i＋1，j′）を次の変化点アドレスとする。

以上の操作を繰り返すことにより，一連の色変化点画素を抽出できる。

（2）線分と円弧の認識

3.3節で詳細に考察する。

（3）図形コマンドへの変換

線分と円弧の認識結果は，（２）において始点，終点，および円弧の中心点として保存されているので，これに色符号と制御符号を付加することにより図形コマンドに変換する。図形コマンドの構成を図３．２に示す。

３．３　線分と円弧の認識法[55],[56]

３．３．１　線分と円弧の判別式

画素列からテンプレートマッチングに基づいて線分と円弧を抽出するために，線分と円弧に対するテンプレートを定義し，画素列とテンプレートとの一致度判定法について考察する。ここで，画素列D_{ij}を，

$$D_{ij} = \{P_i, P_{i+1}, \cdots, P_j\}$$
$$P_k = (x_k, y_k),\ i \leq k \leq j \qquad (3.1)$$

と表し，以下の定義を行う。

［定義3.1］　ｘｙ平面上の線分または円弧を中心線として，その両側に幅ｄの帯を有する帯状図形をテンプレートと考え，これをチューブと呼ぶ。また，画素列D_{ij}のすべての画素がチューブに含まれるとき，D_{ij}は誤差ｄで線分または円弧で近似できるという。■

［定義3.2］　線分または円弧に対して，関数Ｈ（ｘ，ｙ）を作ることができ，チューブ内のすべての画素に対しＨ≦Ｃとなり，チューブ外の画素に対しＨ＞Ｃとなるとき，関数Ｈを線分または円弧の判別式という。但し，Ｃは線分または円弧から定まる定数である。■

［定義3.3］　曲面ｚ＝Ｈ（ｘ，ｙ）に対して，ｘｙ平面上の任意の点(x_i, y_i)におけるｚの絶対値$|H(x_i, y_i)|$を変位と呼ぶ。■

定義３．２の判別式は，次のようにして作ることができる。

（１）線分の判別式H_L

画素列をD_{ij}，直線の方程式を式（３．２）とする。

$$f(x, y) = ax + by + c \qquad (3.2)$$

図３．３から，チューブの端点Q_1，Q_2での平面ｚ＝ｆ（ｘ，ｙ）の変位は，次のようになる。

$$|f(Q_1)| = |f(Q_2)| = d\{a^2 + b^2\}^{1/2} \qquad (3.3)$$

したがって，画素Pが$f(x,y)=0$のチューブに含まれるためには，

$$|f(P)| \leqq d\{a^2+b^2\}^{1/2}=C \qquad (3.4)$$

が成立すればよい。

一般に，線分は2点が与えられると決まるので，画素列D_{ij}の始点P_iと終点P_jを用いて表すと，次のようになる。

$$H_L(x,y)=|(y_i-y_j)(x-x_i)-(x_i-x_j)(y-y_i)|$$

$$C=d\{(x_i-x_j)^2+(y_i-y_j)^2\}^{1/2} \qquad (3.5)$$

但し，$P_i=(x_i,y_i)$，$P_j=(x_j,y_j)$である。

（2）円弧の判別式H_A

円弧の方程式を式（3.6）とする。

$$g(x,y)=(x-a)^2+(y-b)^2-r^2=0 \qquad (3.6)$$

図3.4から，画素Pがチューブに含まれるためには，式（3.7）が成立しなければならないことが分かる。

$$g(Q_2) \leqq g(P) \leqq g(Q_1) \qquad (3.7)$$

但し，$g(Q_1)=2rd+d^2$

$g(Q_2)=-2rd+d^2$

いま，$d/r \ll 1$と仮定すると，点Q_1とQ_2における変位は，

$$|g(Q_1)| \fallingdotseq |g(Q_2)| \fallingdotseq 2rd \qquad (3.8)$$

となり，円弧の判別式（3.9）を得る。

$$H_A(P)=|g(P)| \leqq 2rd \qquad (3.9)$$

画素列D_{ij}の始点P_i，終点P_jおよび中央の点P_m（$m=[i+j]/2$）を用いると，円弧の判別式として式（3.10）を得る。

$$H_A(x,y)=|g(x,y)|$$

$$=|(x-a)^2+(y-b)^2-r^2| \leqq 2rd \qquad (3.10)$$

但し，

$$a=(1/2\Delta)\{w_i^2(y_j-y_m)+w_j^2(y_m-y_i)+w_m^2(y_i-y_j)\},$$

$$b=(1/2\Delta)\{w_i^2(x_m-x_j)+w_j^2(x_i-x_m)+w_m^2(x_j-x_i)\},$$

$$\Delta=\begin{vmatrix} x_i-x_m & y_i-y_m \\ x_j-x_m & y_j-y_m \end{vmatrix}\neq 0,$$

$$w_k^2=x_k^2+y_k^2 \quad (k=i,j,m),$$

$$r=\{(x_i-a)^2+(y_i-b)^2\}^{1/2}.$$

また，円弧の始点と終点が決まってもその回転方向の違いから2個の円弧が考えられる。このため，ベクトル$\overrightarrow{P_iP_m}$と$\overrightarrow{P_mP_j}$の外積を考えると，

$$\overrightarrow{P_iP_m}\times\overrightarrow{P_mP_j}=-\vec{k}\cdot\Delta$$

となるので，$\Delta>0$の場合は時計方向回転の円弧，$\Delta<0$の場合は反時計方向の円弧と判定できる。但し，$\vec{k}$はz軸方向の単位ベクトルを表す。

（3）判別式の適用法

画素列D_{ij}が線分であるか円弧であるかを判定する場合，D_{ij}に含まれるすべての画素に対して双方の判別式を適用することは能率的でない。

このため，始点P_i，終点P_j及び中央点P_mが同一直線上にあれば線分の判別式を適用し，そうでなければ円弧の判別式を適用する。

すなわち，線分の判別式（3.5）において，

$H_L(x_m,y_m)\leqq C$ならば，線分の判別式，

$H_L(x_m,y_m)>C$ならば，円弧の判別式，

を適用する。

また，P_i，P_j，P_mが同一直線上にあり，かつ，D_{ij}が1つの線分でない場合は1つの円弧ともなり得ないので，D_{ij}に再び円弧の判別式を適用する必要はない。

3.3.2 認識アルゴリズム

ここでは，テンプレートマッチングにより画素列から最大の線分成分と円弧成分を探索する方法について考察する。画素列D_{ij}が多くの線分や円弧から構成されている場合には，画素列を分割しそれぞれを別個に認識し図形コマンドに変換する必要がある。その場合，図形コマンド数を少なくし符号化効率を向上させるため，1つの図形成分として抽出する部分画素列に含まれる画素数をできるだけ多くすることが要求される。

そこで，最大の図形成分を次のように定義する。

［定義3.4］ 画素列$D_{i,j}$に対し，部分画素列$D_{i,M}$は1つの図形成分として認識できるが，$D_{i,M}$に次の画素を付加した部分画素列$D_{i,(M+1)}$は，1つの図形成分として認識できない場合，部分画素列$D_{i,M}$を最大の図形成分という。 ■

画素列$D_{i,n}$から最大の図形成分$D_{i,M}$（$M \leq n$）を求める方法に関し，次の補題が成立する。

［補題3.1］ 画素列$D_{i,k}$（$i < k \leq n$）が1つの図形成分として認識できないとき，少なくとも1つのm（$i < m < k$）が存在し，$D_{i,m}$は1つの図形成分として認識できる。 ■

（証明） 少なくともk＝i＋1とすれば，$D_{i,k}$は2個の画素しか含まないので線分である。 （証明終り）

補題３．１より，

$$Dit_1, \cdots, Dit_h \quad (t_1 < \cdots < t_h)$$

はすべて1つの図形成分として認識でき，

$$Dit_1', \cdots, Dit_h' \quad (t_1' > \cdots > t_h')$$

はすべて1つの図形成分として認識できない画素列が存在し，

$$i < t_1 < \cdots < t_h < M < t_h' < \cdots < t_1' \leq n$$

となることが分かる。

画素列$D_{i,n}$に対し，その最大の図形成分を求めるためには，$D_{i,n}$を適当な画素Pt_kで分割する。Dit_kが1つの図形成分として認識できなければ，$M < t_k$なるMが存在することを意味しているので，画素Pt_{k+1}（$i < t_{k+1} < t_k$）で画素列$D_{i,n}$を分割する。もし，Dit_kが1つの図形成分として認識できれば，$M \geq t_k$なるMが存在することを意味するので，画素Pt_{k+1}（$t_k \leq t_{k+1} < n$）で$D_{i,n}$を分割すればよいことが分かる（図３．５）。

この分割の方法として，画素列$D_{i,n}$を一定の比（＝p：q），すなわち，

$$t_{k+1} = \frac{p \cdot t_k' + q \cdot t_k}{p + q} \qquad (3.11)$$

なる内分点で分割するとき，次の補題が成立する。

［補題3.2］　画素列D_{in}（画素数をn_iとする）の最大の図形成分を求めるため，D_{in}を一定の比（p：q）で内分する場合，分割回数を最も少なくするためには，p：q＝1：1で分割すればよい。また，その分割総回数は，たかだか$\log_2 n_i$である。 ■

（証明）付録1に示す。

以上のことから，極大の図形成分を抽出する次の手順を得る。

［手順］

step1:　{k＝1，t_0＝i，t_0'＝nとする。}

step2:　{画素列$Dt_{k-1}t'_{k-1}$を中間の画素P_Lで1：1に内分する。}

if {部分画素列D_{iL}は1つの図形成分}，

then {t_k＝L，t_k'＝t'_{k-1}}

else {t_k＝t_{k-1}，t_k'＝L}

step3:　if {$t_k' - t_k \leqq 1$}，

then {画素列Dit_kは極大の図形成分}

else {k←k＋1としてstep2へもどる。} ■

3.3.3　計算量の評価

画素列から線分と円弧を認識するために要する計算量は，判別式を決定するために要する総計算量と，画素の座標値を判別式に代入するための総計算量との和で表される。ここでは，判別式を1回決定すること，または，画素の座標値を判別式に1回代入することを1ステップと定義し，1つの画素列から1つの最大図形成分を認識するために要する総ステップ数を求める。なお，画素列D_{in}の画素総数をn_iとする。

まず，判別式を決定するために要する総ステップ数S_Hを求める。S_Hは画素列の分割回数に等しいので，補題3.2から，

$$S_H \leqq \log_2 n_i \qquad (3.12)$$

となる。

次に，画素の座標値を判別式に代入するために要するステップ数S_Dを求める。

画素列D_{in}が1つの図形成分として認識できる場合は，分割の必要がないので，$S_D = n_i$となる。 1つの図形成分として認識できない場合は，式（3.13）が成り立つ。

$$S_D \leq n_i + \sum_{j=1}^{S_H} m_j \qquad (3.13)$$

但し，m_jはj回目の分割で定義された部分画素列に含まれる画素の総数を表す。

ここでは，画素列を1：1に内分しているので，m_jは式（3.14）を満足する。

$$m_j \leq \sum_{k=1}^{j} (n_i / 2^k) \qquad (3.14)$$

したがって，式（3.12）より，S_Dは次式を満足する。

$$S_D \leq n_i \log_2 n_i + 1 \qquad (3.15)$$

以上のことから，総ステップ数Sは，式（3.12）と式（3.15）から，

$$\begin{aligned} S &= S_H + S_D \\ &\leq (n_i + 1) \log_2 n_i + 1 \end{aligned} \qquad (3.16)$$

となり，画素数n_iの画素列から最大の図形成分を認識するための計算量のオーダは，（$n_i \log_2 n_i$）程度であることが分かる。

なお，本認識アルゴリズムでは，判別式を満足しない画素があった時点で次の画素列分割を行うので，実際上は，式（3.16）の上限値より少ないステップ数となる。

次に，ミニコンピュータ（平均命令実行時間=約4.5［μs］）を用いて認識時間を評価した結果について述べる。ここで，判別式の決定に要する時間をt_H，

1画素当りの判別式の変位計算に要する時間をt_Dとすると，1つの極大図形成分の認識に要する時間t_Aは，式（3.12）と式（3.15）から，

$$t_A = t_H \cdot S_H + t_D \cdot S_D$$
$$\leqq t_H \cdot \log_2 n_I + t_D (n_I \log_2 n_I + 1) \qquad (3.17)$$

を満足する。プログラムの動的ステップ数は，線分と円弧の判別式決定に，それぞれ，約140[step]，約1700[step]を要し，判別式の変位計算には，線分と円弧の場合で大差がなく，約250[step]を要しているので，

$$0.53\,[ms] \leqq t_H \leqq 7.7\,[ms]$$
$$t_D \fallingdotseq 1.1\,[ms]$$

となる。

したがって，認識時間t_Aは，式（3.17）より

$$t_A \leqq 7.7 \log_2 n_I + 1.1 (n_I \log_2 n_I + 1)\,[ms] \qquad (3.18)$$

を満足する。

3.4 実験結果

3.4.1 認識例

線分と円弧の認識例を図3.6に示す。（a）は原図形，（b）～（f）は原図形の輪郭画素列を線分と円弧で近似した結果を示す。dはテンプレートの幅の2分の1に対応するものであり，本手法による近似誤差を表していると考えることができる。なお，画素総数が約1,000画素で，認識に要した処理時間はミニコンピュータで約10秒であった。この結果から分かるように，近似精度を決めるパラメータであるチューブの幅を大きくすると図形の細かな変化が表現できなくなるので，チューブの幅は最大でも2ないし4画素にするのが望ましい。

3.4.2　データ圧縮効果

図形をコマンド化することによるデータ圧縮効果は，１画面分の画素数とその中に含まれる線分成分と円弧成分の数に依存する。

いま，Ｎ×Ｎ画素の画面でＰ個の線分成分とＱ個の円弧成分からなる図形を考える。このとき，線分と円弧のコマンドに必要なビット数をそれぞれα，βとし，データ圧縮率を

$$\eta = (\alpha P + \beta Q) / N^2$$

で定義する。但し，図形は２値図形とする。

ここで，一般的なデータ圧縮率の目安を得るため，テレビ画面程度の画面を想定すると，ほぼＮ＝５１２（画素），α＝５０（ビット），β＝７０（ビット）程度である。したがって，線分のみからなる図形の圧縮率をη_P，円弧のみからなる図形の圧縮率をη_Qとすると，

$$\eta_P \approx 1.9 \times 10^{-4} P$$

$$\eta_Q \approx 2.7 \times 10^{-4} Q$$

となる。

図３．７にデータ圧縮率とＰ，Ｑの関係を示す。これより，２００個程度の線分または円弧からなる図形では，１／２６ないし１／１９という高い圧縮率が期待できる。

3.5　むすび

彩色図形の符号化法として，彩色図形の輪郭を線分と円弧で表現し，それぞれをコード情報である図形コマンドに変換する図形コマンド符号化法について考察した。この中で，画素列から線分成分と円弧成分を認識する方法として，テンプレートマッチングに基づいた手法を提案した。本手法の特徴は，線分と円弧に対するテンプレートとの一致度を判定する方法を，画素列の各画素におけるテンプレートに対する変位を計算し比較することで実現した点にある。

定量的考察ならびに実験の結果から，以下のことを示した。

（１）画素総数ｎ個の画素列から１つの図形成分を認識しコマンド化するために要する時間はほぼ$n \log_2 n$に比例する。

（２）画像情報の検索サービスで用いる画面を想定すると，２００個程度の線分または円弧成分からなる図形に対して１／２６ないし１／１９という高いデータ圧縮率が期待できる。

（３）直感的に理解し易いチューブの幅を変更することにより，線分と円弧の近似精度を容易に制御できる。

以上のことから，提案した図形コマンド符号化法は，画像情報の検索サービスなどで多く用いられる彩色図形の有効な符号化法として期待できる。

第４章　画像ファイル作成装置

４．１　彩色図形作成方式[88]

非電話系サービスの中で考えられている画像情報の検索サービス[18],[19]で大量に用いられる文字が混在した彩色図形は，多くが手作業を中心として作成されていた。そのため，作成に時間がかかることや均一な画質を得にくいことなどの問題があった。これを解決するため，２章と３章で考察した線図形入力処理法と図形コマンド符号化法を応用した，彩色図形の入力処理方式の有効性を明らかにするため，入力した線図形を電子的に彩色し符号化する彩色図形作成方式について考察する。以下に，背景と人手による彩色図形作成法について述べたのち，それと対比させて彩色図形作成方式が備えるべき要求条件と装置化のための基本構想を明らかにする。

４．１．１　背景

画像情報の検索サービスを実現するにあたっては，情報センタに大量の画像情報をファイルしておくことが必要である。このファイルの質および容量がサービスの良否を決定する要因の一つと考えられる。しかし，画像を効率的に蓄積するための静止画像の符号化技術[16]や，高速・大容量の画像ファイル装置[89],[90]の研究は多く行われてきたが，大量の彩色図形を作成し画像ファイル装置へ入力する技術については，あまり研究がなされていなかった。

彩色図形は各種概念の視覚的表現を行うためによく用いられるが，その作成は多くが手作業によっていたので，彩色などに要する画像作成時間が長く，撮影時に色むらが生じるなどの画質上の問題が多く存在していた。また，彩色図形の修正や変更が必要になった場合，再び始めから図形を作り直さなければならなかったので，作成効率の面でも問題があった。そのため，大量の彩色図形を効率良く作成し，画像ファイルに入力する入力処理技術の実現が望まれていた。このような背景から，彩色図形の作成と画像ファイルへの入力を目的とした画像ファイル作成装置の実現方式について検討する。

４．１．２　手作業による彩色図形作成法

手作業による彩色図形の作成法は，図４．１に示すような手順で透明フィルムに描画や彩色などを行うものであった。

以下に，各作成段階の概要を述べる。

（１）下絵制作段階

彩色画像は構成面から文字と図形に分けて考えることができる。この段階は１枚の彩色図形に入れる文字と図形のおおまかな形，位置，色などについて検討し下絵を作成する段階であり，人間の創造性が多く要求される。

（２）版下制作段階

下絵を清書する段階である。文字については，写植により適当な大きさでフィルムに印字する。図形については，紙に線画を作成し必要部分のみを切り出し，文字を写植した印画紙にはりつけて版下とする。この作業内容は比較的単純であるが，作業に細心の注意と技術を必要とするため，時間がかかるという特徴がある。特に，文字単位の修正や線画の一部修正ができないため，版下の作り直しも度々発生する。

（３）画稿制作段階

版下を透明フィルムに転写しカラーインクで彩色を行う段階である。作業内容は版下制作に比べ単純であるが，絵筆による着色は多大の時間と労力を必要とし，色むらをなくすことは技術的に極めて困難である。また，色の一部修正ができないために画稿の作り直しが発生する。

（４）撮影

作成した複数枚の画稿を重ね合わせ所望の彩色図形を得たのち，カメラなどで撮影して蓄積する媒体に適した形に変換する段階である。

（５）符号化

撮影した彩色図形は画像ファイル装置に蓄積するため，画像入力装置で読みとり符号化する。画質をほとんど劣化させず符号化するためには，彩色図形あたり１３０Ｋバイト程度の符号量を必要とし，蓄積コストや画像ファイル装置への書き込み・読み出し時間において問題となる。

4.1.3　要求条件と基本構想

手作業による彩色図形の作成法にはいくつかの問題点が残されている。これらも含め，画像ファイル作成装置として考えられる要求条件を表4.1に示す。これら要求条件をすべて満足させることは，手作業中心の方式では不可能である。このため，作業の電子化により問題解決を図ることを考える。すなわち，既に述べたように，手作業による彩色図形の作成工程は，創造性を必要とする部分と作業性を中心とする部分に分けることができるので，極力，後者の自動化を図り作業時間の短縮を達成する。また，作業の自動化にあたっては，彩色図形作成に必要な画像処理や文字処理を実現し，手作業による方法では満足させることができなかった機能の実現や性能の向上を図る。

図4.2に画像ファイル作成のための各制作段階の構想をまとめる。本方式は各制作段階において，それぞれ，ファクシミリを適用した線図形入力処理，彩色図形処理，画面の電気的合成演算処理，図形コマンド符号化処理を適用することにより，画像ファイル作成の条件を満足させている点に特徴がある。

4.2　装置構成[91]

画像ファイル作成装置の基本構成を図4.3に示す。本装置は，処理部として汎用ミニコンピュータを用い，線画入力部，線画入力処理部，編集メモリ部，表示処理部，および文字入力部と文字情報蓄積部からなる構成としている。本装置の諸元を表4.2に示す。

以下に，主な構成部の特徴について述べる。

（1）線画入力部

線画の入力方法としては，操作者が入力機器をを用いて直接描画したりデータを投入して線画を作成する会話形入力法と，予め紙面に描画した線画を画像読み取り装置で入力する自動入力法がある。ここでは，ファクシミリ送信機を適用した自動入力法を実現し，大量の線画を短時間で入力できるようにした。

（2）線画入力処理部

入力された線画データに雑音や用紙の傾きなどに起因する歪が生じるのは回避困難なため，これらを除去したり補正する機能を実現し入力データの品質を高め

た。また，線の幅を均一にしたり線画の一部を削除するなど，線画データ自体を自動的に制御できる形式に変換する機能を実現し，操作者の作業負荷を軽減できるようにした。更に，線画入力部と表示部の解像度を一致させるための密度変換処理を実現している。このように，線画入力段階で前処理を積極的に行う構成にすることにより，線画入力段階での線図形の品質を高めている。

（３）編集メモリ部

入力した線画や作成中の彩色図形を一時的に蓄積するメモリ部である。これは，彩色などの処理用メモリとして，あるいは，表示部のリフレッシュ信号を作成するためのメモリとして用いる。編集メモリは，彩色図形をランレングス符号[86]で記憶するランレングス符号メモリと，画素単位に記憶するＰＣＭ符号メモリで構成する。このように編集メモリを２画面分持たせることにより，文字画面と図形画面あるいは図形画面同士を電気的に合成することにより，作成済みの画面を有効に再利用できる構成にしている。また，編集メモリ間でのデータの相互変換を実時間で実現するため，ランレングス符号化処理と復号化処理をハードウェアで実現し，１フレーム時間での処理を可能にしている。

（４）表示処理部

編集メモリ部の内容を読出し，表示部のリフレッシュ信号を作成する部分である。編集メモリには２画面分の容量を持たせているので，それぞれの読み出し内容を表示信号の段階で合成することにより，編集メモリの内容を変更することなく実時間で複数画面の合成信号が得られる構成としている。また，複数画面を合成する場合には，画像の位置を細かく制御する機能が不可欠となるので，編集メモリの読み出しタイミングを制御することにより，実時間で画面の平行移動ができる構成としている。

（５）文字入力部ならびに文字情報蓄積部

彩色図形には，その意図を伝えるために文字がよく使われる。文書作成と際立って異なる点は，モザイクパターンのような記号も多く用いられ，また，画面当りの文字数が数十文字と少なく，画面の見やすさから文字入力速度よりむしろ文字位置の細かな制御が要求されることである。このため，文字情報蓄積部には文字コード，表示位置，文字色の情報を蓄積する構成としている。

4.3　図形画面作成処理[88],[91]

4.3.1　線画入力処理

線画入力処理で実現すべき機能は，彩色図形作成処理の機能条件に依存する。たとえば，入力された線画は入力後には修正したい部分や，彩色後の画面では削除したい冗長な部分などを含んでいるので，それらを識別するための制御情報を付加して線画情報を編集メモリへ送ることが必要となる。このため，彩色図形の作成を，以下の4つの処理モードに類型化し，これらを実現するための基本機能を線画入力処理として実現する方法を採用する。

（1）汎用モード

線画を忠実に入力し彩色後もほぼそのままの状態で残すモード。

（2）線幅均一化モード

最終形態の彩色図形として線画部分を均一な線幅とするモード。

（3）水平垂直化モード

線画を水平線と垂直線に変換するモード。

（4）領域指定モード

彩色する領域や文字を入力する領域を指定する線と線画として残す線の太さを変えて入力し，彩色時に領域を指定する線を削除するモード。

図4.4は彩色図形作成モードと線画入力処理の関係を示しており，線画入力処理機能を，密度変換処理，細線化処理，水平垂直線化処理，線幅識別処理，太め処理から構成している。

密度変換処理は，ファクシミリの解像度（主走査方向1455画素，副走査方向1706画素）をテレビ受像機の解像度（副走査方向485画素，主走査方向512画素）に変換する処理である。細線化処理と水平垂直線化処理は，2章で考察したアルゴリズムを処理部のソフトウェア処理で実現している。線幅識別処理は，主走査方向と副走査方向の黒画素ランの長さから識別する方法[92]を採用し，2種類の線幅（太線は1㎜以上，細線は0.3㎜以下）を識別する。太め処理は，細線化した線画データに対して画素を付加し一定幅の線画に変換する。また，線画入力時間を短縮化するため，これらの機能は逐次的に実時間で処理ができる線順次処理方式で実現している。

4.3.2 彩色処理

彩色図形を得るためには，線画入力処理により作成した線画データに色符号を与え着色することが必要となる。線画を着色する作業には，どの領域をどの色符号で彩色すればよいかを判断する創造的な部分と，領域を指定された色符号で埋める機械的な部分があるので，彩色処理は操作者が判断しながら処理を進める会話形処理で実現している。

線画処理が2値データを対象とした処理であるのに対し，彩色処理は多値データを対象とする処理であるため符号量が増大する。このため，処理時間の短縮や処理用のメモリ削減の点から，彩色処理の実現方法において工夫が必要となる。このため，彩色図形に対し符号量があまり増加しないランレングス符号を利用した彩色処理法[93]を処理部のソフトウェアで実現した。

表4.3は，彩色処理の処理概要を示しており，彩色すべき領域を検出する処理として，操作者が指定した領域を検出する機能と未彩色の領域を自動的に検出する機能を実現し，それらを併用することにより操作性を向上させている。

4.3.3 編集処理

作成した彩色図形を加工して新たな彩色図形を得る場合に必要となる処理で，画面を単位とした処理を特徴とする。

編集処理の概要を表4.4に示す。編集処理では，画面単位の平行移動処理と合成処理を実現し，作成済みの複数枚の画面を重ね合わせることにより，より複雑な画像作成を可能とした。平行移動処理と合成処理は，画素単位での座標変換や論理演算を行うことから，処理は簡単な反面，多くの処理量を必要とする。このため，画素単位の処理をハードウェアで実現し，画面単位のデータ入出力処理をソフトウェアで実現することにより，編集時の実時間処理を可能とし操作性を向上させている。

4.3.4 符号化処理

符号化処理では，作成した彩色図形から画像ファイル装置に蓄積する情報を抽出する。ここでは，3章で考察した図形コマンド符号化処理を処理部のソフトウ

ェアで実現し，編集メモリ部のランレングス符号メモリから彩色図形の輪郭情報を抽出し，輪郭情報を線分と円弧で表現したコード情報に変換することを可能にしている。

4.4　文字画面作成処理[94]

文字が彩色図形に挿入される頻度は非常に高い。このため，文字画面作成の効率化を図ることが重要となる。文字の入力法や文字列の編集法に関しては，ワードプロセッサに代表される文書作成技術の分野で多くの研究がなされてきた。しかし，文書に比べ彩色図形に挿入される文字数は，テレビ受像機への表示を主な目的としているので数十文字と少なく，また，画面上の任意の位置に文字を表示することが必要になるという作成条件上の相違点がある。このため，ここでは文字の表示位置を画素単位にきめ細かく制御することに重点を置いた文字入力と編集処理について考察する。

4.4.1　文字表示位置の制御法

文字パターンとしては文字を画素の集合で表現する画素型文字パターンを用いる。画素型文字パターンはパターンのサイズを表すボディフェースと，文字を構成する画素が配置されるレターフェースからなる。このため，ボディフェースを基準にして画面上に文字を表示すると文字間に冗長な空白部分が生じ，画面上に表示した文字列が不自然になるという問題が起こる。これを防止するための文字表示位置の制御法について述べる。

図4.5に示すように，文字パターンのボディフェースとレターフェースの相違によって生じる左（右）の空白部分を左（右）ピッチと呼び，空白部分の横画素数をPl（Pr）と表す。また，ラスタ走査の座標系と合わせるため，ボディフェースの左上端点を文字パターンの原点とする。

このとき，現在の文字表示位置を（X0, Y0），実際に文字パターンの書き込みを開始する位置を（X1, Y1），文字パターン書き込み後の新たな文字表示位置を（X2, Y2）とし，

$$Y2 = Y1 = Y0$$

$$X1 = X0 - Pl$$

$$X2 = X1 + S + G - Pr$$

$$= X0 + S + G - (Pl + Pr)$$

となるように文字表示位置を制御する（図４．６）。 但し，Ｐｌ，Ｐｒ，Ｓはそれぞれ，（Ｘ０，Ｙ０）に表示する文字パターンの左ピッチ，右ピッチ，横画素数，Ｇは画素数で指定される字間隔を表す。なお，左ピッチと右ピッチの情報は，予め文字対応にテーブル化しておく必要がある。これにより，文字間の冗長な空白部分の発生を防止し，かつ，画素単位で文字表示位置を制御することができる。

４．４．２　表示文字の検索方法

文字列を編集する場合，画面上の任意の位置に表示されている文字を検索することが必要となる。この方法としては，ポインティングデバイスを使用する方法とカーソルキーを使用する方法がある。まず，ポインティングデバイスで指定された画素位置と表示文字の位置との距離を計算し，最も近い文字を検索する。その後は，以下に示す方法によりカーソルキーを用いて文字を検索する。

カーソルが（Ｘ０，Ｙ０）に表示された文字を示し，Ｘ軸方向の移動が指示されたとする。また，文字パターン横ドット画素数（Ｓ）と字間隔（Ｇ）から決まるＸ軸方向の文字表示候補位置を（Ｘ１，Ｙ１）とする。　但し，$X1 = X0 + S + G$，$Y1 = Y0$とする。

このとき，

$$X0 < X2 \leqq X1, \text{ かつ, } Y0 = Y1 = Y2$$

を満足する文字位置（Ｘ２，Ｙ２）を有する文字が存在すれば，その位置を新たなカーソル位置とする。なお，他の方向が指定された場合も同様に扱うことができる。

４．４．３　文字画面編集処理

文字画面の一部修正や変更を行ったり，他の文字画面との合成を行う際に必要となる処理であり，処理の概要を表４．５に示す。 ここでは，文字単位あるいは文字列を対象とした処理として，文字パターンを編集メモリに書き込む方法を制御する論理演算処理，文字列の修正を行う編集処理，および，文字を書き込む範

囲を制御する文字領域処理を実現している。また，画面全体を対象とした処理として，画面の平行移動を行う平行移動処理と２画面を１つの文字画面に合成する合成処理を実現している。これにより，画面の任意位置への文字表示を許容した文字画面を効率的に作成することができる。

４．５　装置のシステム規模

操作者が装置と会話をしながら品質の良い画面情報を短時間で作成するには，画像ファイル作成処理の各段階で性能向上を図り，操作性を向上させることが重要となる。

画像ファイル作成装置が扱う対象は多値画素パターンのため，処理の最小単位は画素単位の処理となる。画素単位の処理は，構成が簡単であるが多くの処理量を必要とするため，性能を阻害する大きな要因となる。このことから，画素単位の処理はハードウェアで実現し，それ以外の処理は，画素単位の処理に比べて処理の構成が複雑になるためソフトウェアで実現し，極力，ランを単位とした処理の構成にすることにより処理性能を向上させる装置設計とした。

以上の設計方針に基づいて開発した，画像ファイル作成装置のシステム規模を表４．６に示す。表４．６はシステムの機能と実現手段の関連，ならびに規模を示しており，プログラムの総規模は約３７[Kｽﾃｯﾌﾟ]，処理に必要な外部バッファは約３２０[KB]である。このうち，きめ細かい会話処理が要求される文字画面作成処理，彩色処理，編集処理において，総プログラム量の約５０％を占めている。また，外部バッファについては，画像データを格納するための編集メモリが全体の約９０％を占めている。

４．６　実験結果

４．６．１　作成例

図４．７に水平垂直化モードで作成した表の作成例を示す。また，図４．８に線幅均一化モードで作成した彩色図形を示す。

以上の作成例から明らかなように，電子的に画像を作成しているので色むらや

色のにじみ，合成による色の劣化がなく，手作業による方法では得ることができなかった良好な画質の彩色図形が作成されていることが分かる。

４．６．２　画面当りの作成時間

画面当りの作成時間は作成する彩色図形の複雑さに依存する。ここでは，複雑さの評価尺度として，線分の数と線分で囲まれた領域の数，および，挿入する文字の数の合計値を用いる。すなわち，線分の数は線画作成時間を，領域の数は彩色処理時間を，文字の数は文字入力時間を支配する要因と考えることができる。

図４．９に，３８種類の彩色図形を対象とした作成時間の測定結果を示す。なお，入力用紙に原画を描画する時間は除いている。

図形種別毎の平均作成時間は，略地図で５５分，間取図で３５．８分，イラスト図とブロック図で３３．１分，幾何学図で２３．３分，その他（軸受図）で２４．７分であった。また，約６８％の彩色図形が４０分以内で作成できた。これにより，人手作業による作成時間（２～４時間）を１／６～１／１０に短縮でき，彩色図形作成の効率化に有効であることが分かった。

４．６．３　画面当りの符号量

彩色図形のコマンド符号化例を図４．１０に示す。画面当りの符号量は図形コマンド符号化時の近似許容誤差に依存する。符号量と近似許容誤差（Ｅ）の関係を，図４．１１に示す。これから分かるように，近似許容誤差を大きくすれば符号量を削減することができるが，近似誤差のため画質が劣化するという問題が起こる。このため，符号化結果を被験者に提示し，十分満足できる「３」，許容できる「２」，許容できない「１」の３段階で主観評価させた。その結果を図４．１２に示す。これより，許容誤差が０．５を越えると画質劣化が著しいので，許容誤差は最大０．５とするのが望ましい。

以上のことから許容誤差を０．２５とした場合，符号量は画面当り平均約９Ｋバイトとなり，画素間相関を用いた方法（ＤＰＣＭ方式で約１３０Ｋバイト／画面）に比べ，平均１／１０以下に符号量を削減できることが分かった。

４．７　むすび

文字が混在した彩色図形を電子的に作成し符号化して画像ファイルへ入力する画像ファイル作成装置の実現法，機能分担法と，この装置を用いた彩色図形の入力実験の結果について述べた。

装置の実現法については，人手作業による彩色図形の作成工程のうち作業性が中心となる部分を，計算機と人間が対話しながら処理を進める会話処理方式で実現し，線図形を入力し，図形画面作成処理，文字画面作成処理，画面単位の編集処理を可能にすることにより，装置が備えるべき要求条件を満足できることを示した。この中で，線画入力処理を構成する基本処理として，線順次型細線化処理を適用することにより，輪郭線に対する制御が可能な彩色図形の作成法が実現できることを示した。機能分担法については，処理は定形的で簡単であるが，画素単位の処理のため処理量が非常に多くなる，線画入力処理おける密度変換処理，線幅識別処理，および，編集処理における合成処理，平行移動処理は，ハードウェア化により高速化する機能分担法を採り，性能の向上を図った。

また，具体化した装置による彩色図形の入力実験結果より，品質が良い彩色図形を手作業による方法の１／１０以下の時間で作成でき，また，図形コマンド符号化法を適用することにより，画素間相関を用いた符号化法に比べて約１／１０の符号量で符号化できることを示した。

以上述べたように，彩色図形の入力処理を可能とする画像ファイル作成装置の構成法を明らかにすると共に，提案した線順次型細線化処理と図形コマンド符号化処理の有効性を確認した。

第５章　画像表示法

５．１　表示方式

表示方式には，画像情報を再生する方法の相違から，ラスタ走査型表示方式とランダム走査型表示方式がある。前者は，表示すべき画像パターンをメモリ上に生成し，それを繰り返し読み出すことにより画像を表示する方式である。後者は，描画命令に従ってＣＲＴ（Cathode Ray Tube）の電子ビームを制御し図形を直接描画し表示する方式である。このような表示方式上の相違から，彩色図形，静止画像，動画像など，画素の集合であるパターンとして表示される画像に対してはラスタ走査型表示方式が用いられる。このため，入力処理でコード情報に変換された漢字や彩色図形をラスタ走査型表示方式で表示するためには，漢字コードを漢字パターンに変換する漢字パターン発生や，図形コマンドを図形パターンに変換する図形パターン発生などの出力処理が必要となる。

本研究では，パターン発生器の経済化や表示品質の向上を目的として，彩色図形の最も基本的な構成要素と考えられる，漢字パターンと線分・円弧などの図形パターンの発生法について検討する。技術的課題として，画素型漢字パターンのサイズ変換法と，滑らかな線分と円弧を発生するためのアンチエリアシング法を取り上げる。

画素型漢字パターンのサイズ変換法としては，画素サイズを比例的に変更する比例法[59]，サイズ変換に伴う品質劣化を抑制するために，変換倍率に対応させてサブパターン毎の変換規則[62]や漢字パターン毎の変換規則[64]を持たせる方法などがある。本研究では，明朝体ＪＩＳフォント[95]を対象にして，高速化のため比例法を基本原理とした任意倍率細線化縮小法と任意倍率拡大スムージング法を提案し，サイズ変換時の品質劣化を抑制する前処理法として飾り除去法，細線化法，および，輪郭線補間法について考察する。これにより，品質劣化を抑制するための変換規則を付加データとして持つ必要がない，経済的で高速化を図ったサイズ変換が実現できることを示す。

線分のアンチエリアシング法としては，線分と画素を面積を持つ領域とみなし線分と重なる画素の面積で輝度を変調する方法[74]-[77]や，線分と画素との誤

差で輝度を変調する方法[78]などがある。本研究では，変位で画素の輝度を変調する図形発生法を提案し，線分と円弧に対する変位による輝度の変調法について考察する。これにより，高速化が図られ同じ原理で線分と円弧のアンチエリアシングが可能となることを示す。

5.2　任意サイズ漢字パターン発生法[66]

5.2.1　予備的考察

漢字パターン上に存在する黒画素の位置座標を倍率に応じて比例的に写像する比例法は，変換結果の文字バランスが良く，かつ，処理に必要な制御情報が少ないため処理が速いという特長を持つ。ここでは，これらの特長を利用した効率の良い文字サイズ変換法を考察する。

比例法には上記の特長がある反面，以下のような問題点がある。

（１）明朝体飾り形状の乱れ
（２）線幅の不揃い
（３）縮小時の飾り付加バランスの劣化
（４）縮小時の線分間隙の消滅
（５）拡大時の斜線部の段差の顕著化

これらを解決するため，縮小時には飾り除去処理と細線化処理を行ったのち比例法により縮小する。すなわち，２４×２４画素未満のパターンサイズでは明朝体を表現することは困難なため，縮小時には明朝体飾りを除去することによって字体を明朝体から細ゴシック体に変換することとする。これにより，問題点（１）～（４）の解決が期待できる。この方法を，任意倍率細線化縮小法（Reduction to Arbitrary size with a Line THinning method：ＲＡＬＴＨ法）と呼ぶ。

また，拡大時には比例拡大処理と併せて輪郭線補間処理を行う。これにより，問題点（１）と（５）の解決が期待できる。この方法を，任意倍率拡大スムージング法（Magnification to Arbitrary size with a Contour Smoothing method：ＭＡＣＳ法）と呼ぶ。図５．１に，ＲＡＬＴＨ法とＭＡＣＳ法の処理の流れを示す。

一般に，ｍ行×ｎ列（ｍ×ｎと略記）の白画素，黒画素の列として表現された

画素型文字パターンAを，

$$A = \{\alpha(i, j);\ i = 1, \ldots, m;\ j = 1, \ldots, n\} \qquad (5.1)$$

但し，$\alpha(i, j) = \{0：白画素，\ 1：黒画素\}$

で表す。

文字サイズ変換は与えられたサイズm×nの文字パターンA（原パターン）を，サイズM×Nの文字パターンA′（変換パターン），すなわち，

$$A' = \Phi(A)$$
$$= \{\alpha'(i', j');\ i' = 1, \ldots, M;\ j' = 1, \ldots, N\} \qquad (5.2)$$

に変換するものとする。

このとき，縦方向倍率R_iと横方向倍率R_jを，それぞれ，

$$R_i = M / m$$
$$R_j = N / n$$

で表すと，比例縮小法と比例拡大法の写像変換式はそれぞれ，以下のようになる。なお，［　］はガウス記号を示す。

（1）比例縮小法

画素$\alpha(i, j)$を，次式によって画素$\alpha'(i', j')$に写像する。

$$i' = [R_i \cdot i + 0.5]$$
$$j' = [R_j \cdot j + 0.5] \qquad (5.3)$$

但し，$R_i < 1$，$R_j < 1$である。

（2）比例拡大法

画素$\alpha(i, j)$を，次式によって画素$\alpha'(i', j')$，画素$\alpha'(i', j'')$，画素$\alpha'(i'', j')$，画素$\alpha'(i'', j'')$に写像する。

$$i' = [R_i \cdot i + 0.5]$$
$$j' = [R_j \cdot j + 0.5]$$
$$i'' = [R_i \cdot (i - 1) + 1.5]$$
$$j'' = [R_j \cdot (j - 1) + 1.5] \qquad (5.4)$$

但し，$R_i > 1$，$R_j > 1$。

ここで，i行の写像であるi′行とi″行がi′≠i″のとき，i行が列方向に拡張されることを意味するので，i行を拡張行と呼ぶ。同様に，j′≠j″のとき，j列を拡張列と呼ぶ。

5.2.2　漢字パターン縮小法（RALTH法）

RALTH法は前処理として飾り除去処理と細線化処理を付加した点に特徴がある。これらの前処理では変換される文字パターンの性質を利用する。すなわち，JIS明朝体漢字パターンには次のような性質がある[95]。

［性質1］

画素マトリクスサイズは24×24画素であり，飾りは16種類存在する（図5.2）。その内訳は，ウロコ3種類，カドウロコ2種類，左上端飾り2種類，左下端飾り3種類，右下端飾り3種類，アタマ1種類，上ハライ1種類，右ハライ1種類，である。■

［性質2］

明朝体で，①縦線分の線幅は1～2画素，②横線分の線幅は1画素，③斜曲線の線幅は1～3画素である。■

［性質3］

アタマ飾りを除く飾りは，ランレングスが4画素以上のランの両端点付近に付加されている。アタマ飾りは，ランレングスが3画素のランの右端点に付加されている。■

次に，これらの性質を利用した明朝体飾り除去処理と細線化処理について述べる。

（1）明朝体飾り除去処理

1つの明朝体の漢字パターンが持つ飾りの種類や数，飾りの付加条件は様々で，同じ飾りでもその形状は一定ではない。また，飾りが付加されるべき位置も一定ではない。このような漢字パターンに対し，飾りを除去する方法として以下の方法が考えられる。

①縦・横線分を除去した後に，残った部分から飾りを抽出する方法。

②マスクパターンを用いて飾りを検出する方法。

①の方法では斜線分と飾りを完全に分離することが困難であるため，飾りを完全に検出除去するためには②の方法が適している。

JIS漢字パターンの［性質1］から，16種類の飾りをすべて検出し除去するためのマスクパターンは，図5.3に示す12種類の飾り除去論理マスクにま

とめることができる。

飾り除去処理では与えられたサイズ２４×２４画素の原パターンを水平方向に走査し，［性質３］からランレングスが４画素以上のランの左端点に対しては左端点用マスクＮｏ．１～４を適用する。右端点に対しては右端点用マスクＮｏ．５～１１を適用する。すなわち，そのうちのいずれかのマスクパターンと文字パターンがマッチすれば飾りを除去する。また，ランレングスが３画素のランの右端点に対しては，アタマ飾り用マスクＮｏ．１２を適用する。

図５．４に飾り除去の例を示す。図中の文字パターンの４隅に対してはそれぞれマスクＮｏ．１，２，９及び１０が適用され，○で示される飾りを構成する黒画素が除去されることになる。

（２）細線化処理

線幅を揃える方法としては線分を細線化して線幅を１画素に統一する方法が考えられる。ところが，文字認識や画像処理の分野で用いられてきた従来の細線化手法を用いると，分岐歪みや縮退を引き起こすという問題があった（図５．５）。

ここでは代表的なHilditchの細線化手法[27]を用いているが，他の方法を用いても同様の現象が起こる。また，従来の細線化手法は処理速度が非常に遅いという欠点があり，高速に文字を出力する際には適さない。

このため，［性質２］を利用し分岐歪みや縮退を引き起こさない，画素型文字パターンに適した細線化法について考察する。ここでは，黒画素ランの長さで画素の状態を判定し，斜曲線部を完全に線幅が１画素になるまで細線化しない方法を採ることとした。これにより，処理の高速化が図れると共に，斜曲線を完全に細線化することにより生じる線分の滑らかさの劣化を防ぐことができる。本方法による処理を，カド部を構成する縦線分の処理とそれ以外の線分の処理に分けて考える。

［ステップ１］

縮小時に起こる線分間隙の消滅を抑制するため，飾り除去処理で得られたパターンから，図５．４に示すように，カド部を構成する縦線分を識別し内側の画素を除去する。■

カド部の検出は，図５．６に示すカド部検出マスクを適用することにより可能となる。すなわち，カド部マスクＮｏ．１で線幅２画素の縦線分の右側の画素を，

Ｎｏ．２で左側の画素を除去することができる。図５．４の例では①で示した画素が除去できる。

しかし，カド部を構成しない縦線分と斜曲線において線幅２画素以上の線分が残存している。このような線分に対してはステップ２で細線化を行う。

［ステップ２］

以下に示すアルゴリズムで細線化を行う。なお，８連結数[30]により消去画素を判定する。

［処理１］　文字パターンを水平方向に走査し，黒ランを検出する。黒ランを検出すると処理２へ進み，文字パターン全体の走査が終了すると全処理の終了とする。

［処理２］

if 黒ラン長＞５　then 処理３へ（ランは横線分）

　else if 左端点の８連結数＝１

　　then 左端点を削除（斜曲線の細線化）

　else if 黒ラン長＝２かつ右端点の８連結数＝１

　　then 右端点を削除

次に処理１へ

［処理３］

if そのランが線幅２画素の横線分の一部分（すなわち，そのランの下に隣接した，６画素以上でほぼ同じラン長を持つ黒ランが存在するならば）

　then その部分の上側の画素をすべて削除（まげハネ，ニョウなどの横線分の細線化）

次に処理４へ

［処理４］

if 左端点の８連結数＝１かつ左端点の右の画素の８連結数＝１

　then 左端点を削除（線分分岐部分の整形）

次に処理１へ　■

図５．４はステップ２の処理の例を示しており②の画素が除去できる。

以上述べたように，細線化処理は線分間隙を広くするように作用するので，文

字の潰れが少ない縮小された漢字パターンを得られるという効果がある。

5.2.3　漢字パターン拡大法（ＭＡＣＳ法）

ＭＡＣＳ法は，拡大変換における最大の問題である斜曲線部の段差の顕著化と飾りの形状変化に対して，文字の輪郭線を平滑化する論理マスクを用いて補間する方法である。

従来の輪郭線補間処理は，拡大後のパターンに対して適用されるために制御用データ量の増大，処理速度の低下，カド部や線分の交差部での過剰補間の発生などの問題があった。

これに対し，本方法は拡大変換とスムージングを同時に行えるように拡大スムージング論理マスクを適用して処理を高速化し，かつ，カド部や線分の交差部分とパターンマッチしないようにマスクを設計することにより過剰補間を防止している点に特徴がある。

比例拡大時に段差が生じ補間を必要とする部分は拡張行または拡張列の点であり，かつ，輪郭線上にある場合に限られる。このことから，拡張行と拡張列および黒画素の隣接関係により，３２種類の拡大スムージング論理マスクにまとめることができる（図５．７）。

このうち，Ｎｏ．６～１０，Ｎｏ．１１～１５，Ｎｏ．１６～２０のマスクは，それぞれ，マスクＮｏ．１～５の左右対称形，点対称形，上下対称形となっている。また，Ｎｏ．２４～２６，Ｎｏ．２７～２９，Ｎｏ．３０～３２のマスクは，それぞれ，マスクＮｏ．２１～２３の左右対称形，点対称形，上下対称形である。

このことから，実質的には論理マスクは８種類にまとめることができるが，処理の簡素化と性能を向上させるため３２種類の論理マスクを用いる。

漢字パターンを拡大するには，まず，与えられたサイズの文字パターンを水平方向に走査し黒画素のランを検出し，その左右両端点の写像画素を式（５．４）から求め黒画素ランを写像する。このとき，端点が拡張行（列）上にあれば拡張行（列）用のマスクとマッチングをとり，マッチすれば黒画素を付加することにより補間を行う。

例えば，図５．８において（ａ）に示す４×４画素の文字パターンの一部を，同図（ｂ）に示す５×５画素のサイズに拡大した場合，点(４，２)の画素①は拡

張列ｊ上にあるため，画素①′と①″に写像されて斜線で示すマスクＮｏ．１の適用によって画素②が付加されスムージングできる。

以上述べたように，本方法は拡大のための写像演算時に同時にスムージングができるので，処理速度の向上に有効と考えられる。

５．２．４　実験結果

提案方法と従来の比例法によるサイズ変換処理結果の比較を図５．９に示す。縮小変換ではＲＡＬＴＨ法によって字体のつぶれの問題が改善され，拡大変換ではＭＡＣＳ法によって斜曲線の段差の顕著化の問題が改善されていることが分かる。

図５．１０に，縮小変換にＲＡＬＴＨ法，拡大変換にＭＡＣＳ法を用いて，文字サイズ１２×１２画素～４８×４８画素に変換した結果を示す。この結果から分かるように，任意サイズへの変換が可能で，かつ，文字バランスも良好で線幅の不揃いも目だたない。これらの結果と，論理マスクに要するデータ量は変換倍率に依存せず一定であり，制御用データ量が少ない変換方法といえることから，提案した手法が文字サイズ変換の条件をバランス良く満足していることが分かる。

また，図の中で２４×２４（細線化）とあるのは，飾り除去処理と細線化処理を用いて細ゴシック体に近い書体へ変換した例である。このように，飾り除去処理と細線化処理は書体変換にも応用可能と考えられる。

図５．１１に，各倍率へのサイズ変換時の処理時間を示す。なお，処理時間は，比例縮小法で２４×２４画素の漢字パターンを２４×２４画素に変換する処理時間を１とした相対時間で表している。このことから，ＲＡＲＴＨ法は比例縮小法の約２．２倍，ＭＡＣＳ法は最大で比例拡大法の約１．５倍の処理時間を必要とすることが分かる。１６ビット５ＭＨｚのＣＰＵによるシミュレーションでは，ＲＡＬＴＨ法で約４０ｍｓ，ＭＡＣＳ法で最大約３５ｍｓであり，ハードウェア化することにより更に高速化が可能である。

5.3　線図形発生法[79]-[81]

5.3.1　予備的考察

ラスタ走査型表示装置に図形を発生する問題は，図形を精度良く近似する画素をいかに効率良く選択するかという問題に帰着できる。画素を選択して線分と円弧を効率良く発生する方法として変位比較法[69],[70]が知られている。しかし，白画素と黒画素といったように2値の画素の集合として図形を表示していたので，商用テレビ受像機程度の画素密度が粗い表示装置に図形を表示した場合，段差が生じて画品質が劣化するという問題があった。

ここでは，変位比較法における画素選択効率の優位性を保ち，かつ，上記の問題を解決する新しい図形発生法について検討する。

このため，基礎となる変位比較法の概要について説明したのち，変位の性質について考察する。

（1）変位比較法による線分発生法

発生すべき線分の方程式を

$$a x + b y + c = 0 \qquad (5.5)$$

とする。なお，直線の傾きθは一般性を失うことなく$0° \leqq \theta < 45°$と考えてよい。

ここで，式（5.5）は

$$z = f(x, y) = a x + b y + c$$

$$z = 0$$

で表される2つの平面の交線とみなせるので，$Q_1(x_0+1, y_0)$と$Q_2(x_0+1, y_0+1)$に対して次式が成立する（図5.12）。

$$f(Q_1) = f(x_0, y_0) + a$$

$$f(Q_2) = f(x_0, y_0) + a + b \qquad (5.6)$$

ここで，$|f(Q_1)|$と$|f(Q_2)|$を変位と呼び，それぞれ距離d_1とd_2に比例している。したがって，変位が小さい方の画素を選択すれば表示すべき線分に最も近い点を得ることができる。

一般に，画素の選択に関し次式が成り立つ。

$$f(x_i \pm 1, y_i \pm 1) = f(x_i, y_i) \pm a \pm b$$
$$f(x_i \pm 1, y_i) = f(x_i, y_i) \pm a \qquad (5.7)$$

この方法は，２つの変位の相互比較でなく変位と一定のしきい値の比較で置換でき，更に画素選択を効率化できる[70]。

［アルゴリズムＩ］

現在点（x_i, y_i）から，ｘ方向に＋１した画素（x_i+1, y_i）における変位｜ｆ（x_i+1, y_i）｜と｜ｂ｜／２を比較して，

｜ｆ（x_i+1, y_i）｜ ≦ ｜ｂ｜／２ならば，（x_i+1, y_i）

｜ｆ（x_i+1, y_i）｜ ＞ ｜ｂ｜／２ならば，（x_i+1, y_i+1）

を選択する。このとき，｜ｂ｜／２を基本しきい値と呼ぶ。 ■

（２）変位比較法による円弧発生法

図５．１３に示すように，各オクタントでの画素の選択方向が異なるが，第１オクタントの反時計回りの円弧で考えても一般性は失われない。

いま，現在点をＰ（x_i, y_i）とすると，次に選択されるべき画素は，

外部点Q_1（x_i, y_i+1）または，

内部点Q_2（x_i-1, y_i+1）

である。そこで，Q_1とQ_2の画素のうち，どちらが円弧に近いかを以下のように判定する（図５．１４）。

円の方程式は中心を原点にとれば，

$$x^2 + y^2 - r^2 = 0 \qquad (5.8)$$

となり，これは

$$z = f(x, y) = x^2 + y^2 - r^2$$
$$z = 0$$

で表される回転放物面と平面との交線である。したがって，

$$f(Q_1) = f(x_i, y_i) + 2y_i + 1$$
$$f(Q_2) = f(x_i, y_i) - 2x_i + 2y_i + 2 \qquad (5.9)$$

が成立する。ここで，変位｜f（Q_1）｜と｜f（Q_2）｜の小さい方の画素を選択すれば表示すべき円弧に最も近い点を得ることができる。

この円弧の発生アルゴリズムは，以下のように変位と一定のしきい値との比較で置換でき，更に画素選択の効率化が図れる[70]。

［アルゴリズムⅡ］

現在点（x_i, y_i）から，y方向に＋1した画素（x_i, y_i+1）における変位｜f（x_i, y_i+1）｜とr（円弧の半径）を比較して，

｜f（x_i, y_i+1）｜ ≦ r ならば，（x_i, y_i+1）

｜f（x_i, y_i+1）｜ ＞ r ならば，（x_i-1, y_i+1）

を選択する。このときrを基本しきい値と呼ぶ。 ■

（3）変位の性質

［定義3.3］の変位の定義から，正規化変位を次のように定義する。

［定義5.1］ 線分または円弧の方程式をf（x，y）＝0，基本しきい値をKとするとき，

$$|f_N(P)| = |f(P)| / K$$

を点Pにおける正規化変位と定義する。 ■

正規化変位は，発生する線分または円弧の線幅に密接に関係する性質を有している。

［性質5.1］ 第1オクタントの直線：f（x，y）＝ax＋by＋c＝0（b＞0）に対し，正規化変位が｜f_N（P）｜＜t（t：非整数）を満足する画素のy方向の個数は，［t］＋1を越えない。但し，［t］はtを越えない最大の整数を表す。

（証明）付録2に示す。 ■

［性質5.2］ 円弧：f（x，y）＝$x^2+y^2-r^2=0$に対し，

正規化変位が ｜f_N（P）｜＜t（0＜t≦r）を満足する画素の半径方向の個数は，［$\sqrt{2}$ t］＋1を越えない。

（証明）付録3に示す。 ■

5．3．2 輝度変調法

テレビカメラによって図形を撮像してCRTモニタに表示する場合を考えると，

ＣＲＴの各画素の輝度はビームスポットと図形との重なり部分の面積Ｉによって決まる。図５．１５に示すように，スポット（画素）を円形とし強度は一様と仮定すると，２値表示の場合Ｉ≧１／２（スポットの面積を１とする）ならば一定輝度を与え，それ以外の場合には輝度を与えないことになり，図形はぎざぎざした感じになる。しかし，実際には図形のエッジ部分は，図形とスポットとの重なり部分の面積Ｉの大きさに応じて輝度が変調され，図形は滑らかに見える。このことから，図形と画素が面積を持っているとみなし，それらの重なり部分の面積で輝度を変調し図形を発生する方法が提案されたが，面積計算に多くの計算量を要し図形発生速度が遅いという問題があった。

ところで，図形の方程式が与えられたとき，実際の図形からの距離が小さい画素には高い輝度を，距離が大きい画素には低い輝度を与えることにより同様の効果が期待できる。しかし，距離を直接求めるためにも比較的多くの計算量を必要とする。

このため，変位の性質から距離の大小関係を変位の大小関係に置き換えて，変位によって輝度を変調することを特徴とする線図形発生法を提案する。

図５．１６は変位による輝度変調法の原理を示した図である。ｆ（ｘ，ｙ）＝０は発生すべき曲線，　Ｓはｘｙ平面と曲面ｚ＝ｆ（ｘ，ｙ）の交点，　A_iとB_iは等変位線を表している。変位は等変位線A_1，A_2，A_3あるいはB_1，B_2，B_3と移るに従って大きくなるので，　領域H_iとH_i'に含まれる画素には第ｉ番目の輝度を与えることにより輝度を変調する。

ところで，画素を選択して図形を発生する場合には，表示図形の精度を高めるため真の図形からの距離が単位長以内にある画素を選択する必要があるので，この範囲での距離と変位の比例関係が問題となる。距離と変位は線分の場合には比例するが，円弧の場合には比例しない。

いま，円弧の基本しきい値をＫとしたとき，真円からｄだけ離れた点Ｐにおける正規化変位は，

$$
\begin{aligned}
|f_N(P)| &= |f(P)| / K \\
&= |(r+d)^2 - r^2| / r \\
&= |2d + d^2 / r| \qquad (5.10)
\end{aligned}
$$

で与えられる。図５．１７は式（５．１０）の関係を図示したものであり，画素間

隔を単位長とし，dの符号は円の外側を正，内側を負としている。これから分かるように，|d|≦1の範囲では正規化変位と距離dは，ほぼ$f_N=2d$の関係を満足している。このことから，円弧の場合においても距離で輝度を変調したのと等価な効果が得られることが分かる。

以上のことから，n階調で画素の輝度を変調するには，次のように輝度を制御すればよい。

いま，画素Pに与えるn階調の輝度を$b_1>b_2>b_3>\cdots>b_n$，正規化変位に対する$(n-1)$個のしきい値を$0<t_1<t_2<t_3<\cdots<t_{n-1}$としたとき，画素Pの輝度変調関数を，

$$B(P)=b_i \quad if \quad t_{i-1}<|f_N(P)|\leq t_i \qquad (5.11)$$

とする。但し，$1\leq i\leq n$，$t_0=0$，$t_n=\infty$（十分大）とする。

式（5.11）は正規化変位を用いているので，しきい値t_iは線分の傾きや円弧の半径に依存しない値となる。

5.3.3 線分発生法

正規化変位の性質5.1から，$t_n<2$とすれば線分の幅はたかだか2画素分であるので，線分の近傍に位置する画素についてのみ輝度を計算すればよい。すなわち，図5.18に示すように，［アルゴリズムI］で求めた画素と，それにy軸方向またはx軸方向に隣接する2個の画素について，式（5.11）を適用すればよい。

ところで，式（5.11）において，変位を正規化するための除算演算が画素対応に必要になり性能劣化の要因になる。このため，実際にアルゴリズムを構成する場合には，しきい値を$T_i=K\cdot t_i$（但し，Kは基本しきい値）とし，式（5.11）を式（5.12）のように変形して用いる。

$$B(P)=b_i \quad if \quad T_{i-1}<|f(P)|\leq T_i \qquad (5.12)$$

但し，$1\leq i\leq n$。

以上のことから，滑らかな線分を発生する以下のアルゴリズムを得る。但し，始点を$P_0(0,\ 0)$とする第1オクタントの線分$f(x,y)=0$とし，基本し

きい値をKとする。

［アルゴリズムⅢ］

Step1: 現在点を$P_i=(x_i, y_i)$とする。

Step2: アルゴリズムⅠから,

$|f(x_i+1, y_i)| \leq K$ ならば,

$P_{i+1}=(x_{i+1}, y_{i+1})=(x_i+1, y_i)$,

$|f(x_i+1, y_i)| > K$ ならば,

$P_{i+1}=(x_{i+1}, y_{i+1})=(x_i+1, y_i+1)$,

を選択する。

Step3: 画素P_{i+1}に対してy軸方向に隣接する画素,

$P'_{i+1}=(x_{i+1}, y_{i+1}+1)$

$P^*_{i+1}=(x_{i+1}, y_{i+1}-1)$

を求める。

Step4: 式（５．１２）を適用し，各画素の輝度$B(P_{i+1})$，$B(P'_{i+1})$および$B(P^*_{i+1})$を求める。

Step5: P_{i+1}が終点ならば終了。そうでなければ，$i \leftarrow i+1$としStep2へもどる。 ■

５．３．４　円弧発生法

線分の場合と同様に考えると，式（５．１１）が適用される画素は図５．１９に示すように絞ることができる。すなわち，［アルゴリズムⅡ］で求めた画素とそれにx軸方向またはy軸方向に隣接する画素に適用すればよい。

以上のことから，滑らかな円弧を発生する以下のアルゴリズムを得る。但し，原点（０，０）を中心とする第１オクタントの円弧とし，基本しきい値をKとする。

［アルゴリズムⅣ］

Step1: 現在点を$P_i=(x_i, y_i)$とする。

Step2: アルゴリズムⅡから,

$|f(x_i, y_i+1)| \leq K$ ならば,

$P_{i+1}=(x_{i+1}, y_{i+1})=(x_i, y_i+1)$,

$|f(x_i, y_i+1)| > K$ ならば,

$$P_{i+1} = (x_{i+1}, y_{i+1}) = (x_i - 1, y_i + 1),$$

を選択する。

Step3: 画素P_{i+1}に対してx軸方向に隣接する画素,

$$P'_{i+1} = (x_{i+1} + 1, y_{i+1})$$

$$P^{\bullet}_{i+1} = (x_{i+1} - 1, y_{i+1})$$

を求める。

Step4: 式（５.１２）を適用し，各画素の輝度$B(P_{i+1})$，$B(P'_{i+1})$および$B(P^{\bullet}_{i+1})$を求める。

Step5: P_{i+1}が終点ならば終了。そうでなければ，$i \leftarrow i + 1$としStep2へもどる。 ■

５.３.５ 計算量の評価

アルゴリズムⅢとアルゴリズムⅣの計算量を，改良型変位比較法（アルゴリズムⅠとアルゴリズムⅡ）と比較した場合の動的ステップ数の相対比として評価する。

計算量を支配する主な要因は，変位計算，変位としきい値の比較演算，座標値の更新演算，アルゴリズム終了判定演算であり，それぞれの１回当たりの動的ステップ数を，h，c，g，eとする。また，輝度変調の階調数をnとする。

ここで，改良型変位比較法で選択される画素数がm個の線分または円弧を発生する場合の，改良型変位比較法と提案アルゴリズムにおける動的ステップ数を求める。

改良型変位比較法の動的ステップ数S_{mh}は,

$$S_{mh} = m \cdot (h + c + g + e) \qquad (5.13)$$

で近似的に求められる。

一方，提案アルゴリズムでは，改良型変位比較法と比べて，変位計算は３倍になり，変位としきい値の比較演算は，変位を（n－１）個のしきい値と比較するため３（n－１）倍になる。

このことから，提案アルゴリズムの動的ステップ数S_{md}は,

$$S_{md} = m \cdot \{3h + 3(n-1)c + g + e\}$$
$$= m \cdot \{3(h + c + g + e) + 3(n-2)c - 2(g + e)\} \quad (5.14)$$

で近似的に求められる。

したがって，動的ステップ数の相対比ρは，次式で与えられる。

$$\rho = \frac{S_{md}}{S_{mh}} = 3 + \frac{3(n-2)c - 2(g+e)}{h + c + g + e} \quad (5.15)$$

アセンブラで作成したプログラムでは，線分の場合，$h \fallingdotseq 5$，$c = 1$，$g \fallingdotseq 1$，$e \fallingdotseq 2$，円弧の場合，$h \fallingdotseq 7$，$c = 1$，$g \fallingdotseq 1$，$e \fallingdotseq 2$であった。したがって，線分と円弧の場合の動的ステップ数の相対比ρ_Lとρ_Aは，それぞれ，

$$\rho_L = 3 + (n-4)/9$$
$$\rho_A = 3 + (3n-12)/11$$

となる。

図5.20に動的ステップ数の相対比ρ（ρ_Lとρ_A）と階調数nの関係を示す。このことから，2値表示を行う改良型変位比較法[44]（アルゴリズムⅠとⅡ）に比べて，提案アルゴリズムの計算量は，4値表示の場合，約3倍，8値表示の場合，約4倍になることが分かる。

5.3.6 実験結果

図5.21に，アルゴリズムⅠとⅡにより2値表示した直線と円弧の表示例を示す。図5.22は，$t_1 = 1.0$，$t_2 \approx 1.5$，$t_3 = 1.99$に設定し，同様の図形をアルゴリズムⅢとⅣを用いて4階調で表示した例である。これから明らかなように，従来の手法では得られなかった滑らかな線分と円弧の表示が可能となっている。

図5.23は輝度の階調数を変化させたときの表示例であり，線分の傾きは3／100，円弧の半径は50画素である。上から順に2，4，8，16レベルの輝度を画素に与えて表示している。

なお，表示装置には多値表示が可能なラスタ走査型表示装置（５１２×５１２画素，階調：最大８ビット／画素，分解能：約２画素／mm）を用いた。

本アルゴリズムで図形を発生したときの図形の滑らかさを決めるパラメータは，式（５．１１）から輝度｛b_1，b_2，・・・，b_n｝，正規化変位に対するしきい値｛t_1，t_2，・・・，t_{n-1}｝および輝度のレベル数ｎ（ｎ＞２）である。これらは互いに相関しており，表示装置の分解能にも依存する。そこで，

$$t_i = t_1 + (t_{n-1} - t_1)(i - 1)/(n - 2) \qquad (5.16)$$

のようにパラメータを決めて実験を行ったところ，以下の点が分かった。

（１）正規化変位が０に近い部分では高輝度の画素が集中するので，ほかの部分に比べて太く見える。この現象は特に線分において傾きが小さいほど顕著になる。このため，$t_1 = 0.8 \sim 0.9$程度に設定し高輝度画素の割合を低めにした方が，比較的図形の滑らかさが向上する。

（２）４階調の表示でも図形がぎざぎざした感じはかなり改善される。しかし，$t_1 < 0.8$に設定すると，低輝度の画素の割合が高くなり発生した図形が途切れて見える。この現象は４階調表示のとき特に著しい。

（３）線分の場合，４階調表示より８階調表示，１６階調表示の方が図形が滑らかに見えるが，８階調表示と１６階調表示の間には大きな差が認められない。一方，円弧の場合，４階調表示，８階調表示，１６階調表示の間には大きな差は認められない。

（４）図形発生時間に関しては，アルゴリズムの性格上，真の線分あるいは円弧の近傍の３画素を対象とするので，アルゴリズムⅠあるいはⅡと比べて，４値表示の場合，３倍程度であり，８値表示の場合，４倍程度である。

５．４　むすび

彩色図形の出力処理の基本となる処理として，比例法を基本原理とする任意サイズの漢字パターン発生法と，アンチエリアシングを可能とし滑らかな線分と円弧を発生する線図形発生法について考察した。漢字パターン発生法では，比例法における品質劣化要因を抑制する前処理を加えた方法として，縮小変換法では任

意倍率細線化縮小法（ＲＡＬＴＨ法），拡大変換法では任意倍率拡大スムージング法（ＭＡＣＳ法）を提案した。また，線図形発生法では，線分あるいは円弧と画素との距離が変位にほぼ比例するという変位比較法の特徴を用いて，変位の大きさで画素の輝度を変調する多値表示法を提案した。以下に得られた結果を要約する。

（１）ＲＡＬＴＨ法では，品質劣化要因となる明朝体飾りの除去処理法と，線分間隙の消滅に伴う品質劣化を抑制するため，線分間隙の内側を細める細線化処理法を考察し前処理として加えた。また，ＭＡＣＳ法では，段差の発生に伴う品質劣化要因を抑制するため，比例拡大処理と同時に平滑化を行う輪郭線補間処理法を考察し前処理として加えた。これにより，比例法の１．５倍～２．２倍の処理時間で任意サイズの漢字パターンが発生できることを示し，比例法の高速性を保ちその品質劣化要因を抑制できることを示した。

（２）線図形発生法では，変位の大きさによる画素の輝度の変調原理について考察し，変位比較に基づいた輝度変調式を導き，同じ原理で線分と円弧のアンチエリアシングが可能となることを明らかにした。これにより，２値表示を対象とした変位比較法の３～４倍程度の時間で，２値表示の方法では得られない滑らかな線分と円弧が８階調程度で発生できることを示した。

第６章　結　　論

６．１　本研究で得られた成果

ＬＳＩやローカルエリアネットワークなどのハードウェア技術や通信技術の発展に支えられて，計算機システムは大きな変貌を遂げつつある。その大きな特徴は処理の分散化と扱う情報メディアの多様化にあるといえる。

処理の分散化についていえば，ソフトウェア技術の発展にも支えられて，既に，計算機システムの利用者は物理的に離れた計算機資源をネットワークを介して有効に活用することが可能になっている。情報メディアの多様化については，計算機システムで扱う情報メディアがデータ，画像，音声といったように多様化し大容量化の一途を辿っている。

このため，これらを統合化して通信したり相互間での通信を可能とするマルチメディア通信の実現が期待されている。その実現のためには解決すべき多くの課題が存在する。その中でも，多様化する情報メディアのうち利用者に最も適した情報メディアに変換して情報の伝達を可能にすることが大きな課題である。これを解決するためには，特に，膨大な情報量を持つ画像を認識して伝送に必要な情報を抽出し編集や加工が容易なデータに変換する入力処理技術と，データを逆変換して画像を再構成する出力処理技術の確立が重要な課題であると考える。

本論文は，以上のような背景の下で，画像ファイルの入出力処理の高度化に資するために彩色図形の入出力処理法について検討したもので，以下に本研究で得られた成果を要約する。

第２章では，線幅の変更・均一化，不要な線の除去，水平垂直線の修正など線図形の制御機能を実現し線図形入力過程における図形品質を向上させることを目的に，彩色図形の輪郭線を構成する線図形を対象とした線図形入力処理法として細線化処理法について検討した。技術的課題として処理の経済化と高速化を取り上げ，処理アルゴリズムの構成法について考察した。

処理の経済化については，画素情報を保持するためのメモリとして，図形を読み取る際の連続する３走査線分の画素情報をのみを保持するメモリを用い，画素情報の入力に同期させて冗長な画素を削除する線順次型処理方式を提案した。

処理の高速化については，画素単位の処理となる３×３の画素マスクを用いた論理演算によらない冗長画素の判定法として，縦線分と横線分のランの連結関係を用いた操作に基づいた方法を提案した。この中で，連続する３走査線間のランの連結状態をランの長さ（縦線分または横線分）で分類し，それぞれの分類毎にランの削除範囲を決定する処理を実現することにより処理アルゴリズムが構成できることを示した。

アルゴリズムの定量的な検討により，線幅がＷの線図形を対象とした場合，３×３の画素マスクを用いた方法に比べて処理計算量は近似的に２／Ｗとなることを明らかにした。また，提案した手法によると，細線化図形の品質は縦線分と横線分を識別するパラメータに依存するが，線幅は一様と仮定したときパラメータの最適値が存在することを定量的に示し，線幅の１．２～２．５倍に設定することにより図形の縮退がない良好な処理結果が得られることを示した。

以上の成果により，彩色図形の入力処理に必要な線図形入力処理に対する要求条件を満足する細線化処理の実現法を示すことができた。

第３章では，彩色図形をコード情報に変換することを目的として，彩色図形の輪郭形状を線分・円弧などの図形要素で表現する図形コマンド符号化法について検討し，テンプレートマッチング法に基づいた手法を提案した。技術的課題として，画素列で表された輪郭情報から線分と円弧の特徴を有する最大長の点列を抽出するため，テンプレートの表現法，テンプレートと画素列との一致度の判定法，最大長点列の探索法について考察した。

テンプレートの表現法については，線分または円弧を中心線とした一様な幅を持つ可変長の帯状図形をテンプレートとして定義した。テンプレートとの一致度判定法は，画素がテンプレートの領域に含まれるかどうかを判定する方法とし，変位比較に基づいた判別式として定式化でき，線分と円弧に対して同じ原理で判定できることを示した。最大長点列の探索法については，画素列に対してテンプレートを２分探索の方法に基づいて適用することにより，最も効率的な探索が可能なことを明らかにした。

更に，定量的な考察を加え，画素総数ｎ個の画素列から１つの線分または円弧を認識するために要する時間はほぼ$n \log_2 n$に比例することを示すと共に，２００個程度の線分または円弧成分からなる図形に対して１／２６ないし１／１９と

いう高いデータ圧縮率が期待できることを示した。

以上の成果により，彩色図形の伝送や蓄積に真に必要となる情報を抽出するための彩色図形の入力処理として，図形コマンド符号化処理の有効性と実現の可能性を明らかにすることができた。

第４章では，２章と３章で提案した線順次型細線化処理と図形コマンド符号化処理の応用として，彩色図形を作成し符号化して画像ファイルに入力するための画像ファイル作成装置の構成法について検討した。具体的には，線図形をファクシミリから入力し彩色図形を会話的に作成する方式を提案し，機能面からみた装置の実現法と性能面からみた機能分担法について考察した。

装置の実現法については，人手作業による彩色図形の作成工程を分析し作業性が中心となる部分を計算機と人間が対話しながら処理を進める会話処理方式で実現し，図形画面作成処理，文字画面作成処理，画面単位の編集処理を可能にすることにより，装置が備えるべき機能条件を満足させることができることを示した。また，図形画面作成処理のうち線画入力処理の基本処理として線順次型細線化処理を適用することにより，輪郭線に対する制御が可能な彩色図形の作成処理が構成できることを明らかにした。

機能分担法については，性能向上を可能とし操作性の向上を図ることを目的に，処理は簡単であるが画素単位の処理のため処理量が非常に多くなる，線画入力処理における密度変換処理と線幅識別処理，および，編集処理における合成処理と平行移動処理はハードウェア化する分担法を採用した。

更に，彩色図形の入力実験により，品質が良い彩色図形を手作業による方法に比べて１／１０以下の時間で作成でき，画素間相関を利用した符号化に比べて約１／１０のデータ量で符号化できることを明らかにした。

以上の成果により，線順次型細線化処理と図形コマンド符号化処理の有効性を確認できた。また，本装置の実現方式はキャプテンシステムや画像応答システムなどの画像情報の検索システムにおける画面情報入力装置に適用されており，彩色図形の入力処理の実現可能性と装置化に向けた方針を示すことができた。

第５章では，画像ファイルにコード情報に変換されて蓄積された彩色図形の出力処理の基本となる漢字パターンと図形パターンの発生法について検討した。

漢字パターン発生法では彩色図形の多様化への対応を目的に画素型漢字パター

ンのサイズ変換処理を取り上げ，技術的課題として高速化を達成するため比例変換法を基本原理にする条件でその品質劣化要因を抑制する処理法について考察した。具体的には，縮小変換時の品質劣化要因を抑制するための明朝体飾り除去処理と細線化処理，および，拡大変換時の品質劣化要因を抑制するための輪郭線補間処理などのパターン処理の漢字パターン設計規則に基づいた構成法を明らかにし，これらパターン処理を比例変換の前処理として付加した任意サイズ漢字パターン発生法の実現法を示した。また，実験的考察により，比例変換法の１．５倍～２．２倍の処理時間で任意サイズの漢字パターンが発生できることを示し，比例変換法の高速性を保ってその品質劣化要因を抑制できる方法であることが確認できた。

図形パターンの発生法では２値表示の線分と円弧の表示品質上の問題を解決することを目的に，線分と円弧のアンチエリアシング法について考察した。具体的には，技術的課題として高速化と円弧への拡張性を取り上げ，画素の面積計算や距離計算を必要としない変位によって画素の輝度を変調する原理について考察し，線分と円弧に対する輝度変調法を変位比較に基づいて定式化した。これにより同じ原理で線分と円弧のアンチエリアシングが可能となることを示した。また，処理計算量の定量的評価と表示実験により，２値表示を対象とした変位比較法の３～４倍程度の処理時間で，２値表示では得られない滑らかな線分と円弧が８階調程度で発生できることが確認できた。

以上の成果により，画像ファイルの出力処理の高速化や表示品質向上に向けての一つの指針を示すことができた。

６．２　今後の研究課題

本論文では，彩色図形を対象とした画像ファイルの入出力処理法について述べたが，今後，情報メディアの多様化が進み，より精細な画像の取り扱いの必要性が増すことが想定される。このため，通信や蓄積に真に必要となる情報を画像から抽出し，その情報から画像を再構成する入出力処理技術の研究が引き続き精力的に進められると考えられ，本研究で残された今後の研究課題について若干の私見を以下に述べる。

（１）入力処理技術の課題

第２章で述べた線図形の細線化処理は，走査線方向の画素の集合であるランの連結関係に基づいて細線化する方法である。このため，分岐歪や輪郭雑音に伴う短線分の発生などの細線化処理が有する一般的な問題の解決は，今後の研究課題として残された。これは，画素を単位とした細線化処理の原理的な問題と考えられ，異なる視点からの解決策が必要と考えられる。その一つの方法として，図形の構造上の知識を処理に反映させる方法が有効と思われる。これに対しては，比較的安定に特徴を得やすい図形の輪郭線のベクトル化結果を，図形の構造上の知識として利用する坂内氏らによる方法[96]が良い解決策を与えるかも知れない。

また，第３章では図形コマンド符号化処理について考察したが，彩色図形を一義的に線分と円弧で表現する方法であることから，図形の表現形式としては粗い形式にとどまっている。すなわち，この表現形式の中から通信や蓄積に真に必要な情報を抽出するためには，線分や円弧を単位とした情報を構造化する過程が不可欠になると考えられ，この点の解決は今後の研究課題として残された。図面認識の分野において，図面の記述規則などの知識を用いて認識の処理過程を制御し，認識結果を構造化するという新しい処理の枠組みの提案と試行[97]がなされており，一つの解決方向を示していると思われる。しかし，この枠組みを彩色図形の認識に発展させるためには，ある意図を持って作られる人工画像の知識とは何かという基本的な問題が潜在しており，これを解明して行くことも見逃すことができない重要な研究課題と考える。

（２）出力処理技術の課題

第５章で検討した画像表示法のうち，漢字パターンのサイズ変換法は，ＪＩＳ規格の（２４×２４）画素漢字パターンの設計規則を処理に反映させて高速化を図った方法である。しかし，設計規則やサイズが異なる漢字パターンへの適用性の検討が必要であり，この点の検証は今後の研究課題として残された。今後，出力装置の解像度の向上が更に進展すると，高密度漢字パターンの取り扱いの必要性が増してくるものと思われる。この場合には，漢字パターンの設計規則の自由度が増し，明朝体飾り除去マスクや拡大スムージングマスクのサイズの拡大や種類の追加が必要になり，性能の劣化要因になることが想定される。このため，漢字パターンの高密度化が進むと，漢字パターンの設計規則に依存しない処理の枠

組みを作ることが課題になると考えられる。これに対しては，漢字パターンの輪郭形状を曲線で表現するアウトライン法[65]が良い解決策になるのではないかと考える。

一方，線図形パターンの発生法の検討では，滑らかな線分と円弧を発生するための輝度の制御法を明らかにすることができたが，任意曲線への拡張性の解明は今後の研究課題として残された。また，表示品質と各種パラメータとの相関性に関しては，本研究では定性的な評価にとどまっている。すなわち，表示品質は走査線密度，輝度の量子化レベル，および輝度制御のためのしきい値に密接に関連しており，この関連性を解明することは今後の研究課題として残された。現時点では，表示品質を定量化する決定的な方法が見当たらず，当面，各種パラメータを変化させて視覚心理実験により表示品質を評価することが必要であると考える。

謝　　辞

卒業研究においてシステム工学の基礎を御教示頂いて以来，常に本研究に対して御激励頂き，本論文をまとめるに当たり終始，御指導，御鞭撻を賜りました広島大学工学部 吉田典可教授に心から感謝の意を表します。 また，論文について詳細に御検討頂き， 適切な御教示を賜りました広島大学工学部 市川忠男教授，佐々木博司教授，中前栄八郎教授， ならびに広島大学総合科学部 磯道義典教授，前田渡教授に厚く感謝いたします。

本論文は，筆者が日本電信電話公社武蔵野電気通信研究所(昭和47年5月～昭和53年3月まで)，同公社横須賀電気通信研究所(昭和60年3月まで)，日本電信電話株式会社横須賀電気通信研究所(昭和60年9月まで)，および，同社複合通信研究所(昭和62年9月まで)において行った研究成果をまとめたものであります。

この間，本研究の直接の上司として御指導頂いた日本電信電話株式会社ヒューマンインタフェース研究所 釜江尚彦所長(元事業所通信研究部長，元画像応用研究室長，元表示機器研究室調査役)，東京電機大学 村上伸一教授(元画像応用研究室調査役)，明電舎株式会社 若菜忠技師長(元画像応用研究室調査役)，日本電信電話株式会社企業通信システム事業本部 川田圭一主席技師(元分散処理プログラム研究室調査役)，ＮＴＴデータ通信株式会社産業システム事業本部 側見稔担当部長(元分散処理プログラム研究室調査役)，また，装置設計，画像処理の面で絶大な御支援と御助言を頂いた 日本電信電話株式会社関連企業本部 星野肇夫担当部長(元画像応用研究室長補佐)，同社ヒューマンインタフェース研究所 末永康仁主幹研究員の各位に深く感謝いたします。

また，本論文を執筆する機会を与えて頂くと共に，暖かい御支援を頂いたＮＴＴデータ通信株式会社 藤田史郎代表取締役社長，三角岑生常務取締役 ，同社開発本部 田中義昭取締役本部長，山下徹企画部長，安部孝二第一技術部長 ，渕沢博孝主幹技師，ならびに本論文をまとめることを強く勧めて頂いた荒川弘熙第二技術部長の各位に心から御礼申し上げます。

更に，研究の過程で画像ファイル作成装置の開発を共に行い，貴重な御討論を頂いた 日本電信電話株式会社ヒューマンインタフェース研究所 小倉健司担当課長，河久保秀二研究主任，装置試験に御協力頂いた同社情報通信処理研究所 名倉正計主任研究員，図形入力装置の設計に御協力頂いた同社ＬＳＩ研究所 土屋敏雄主任研究員，ならびに漢字パターン処理のプログラミングと実験に御協力頂いた同社ヒューマンインタフェース研究所 田中一男主任研究員 ，ＮＴＴデータ通信株式会社開発本部 乙田清次主任技師の各位に心から感謝申し上げます。

参考文献

(1) 山本：”文書通信サービスの標準化動向”，電子情報通信学会誌，Vol.72, No.5,pp.554-558(1989).

(2) 田中：”オーディオビジュアルサービスの標準化動向”，電子情報通信学会誌，Vol.72,No.5,pp.548-553(1989).

(3) 稲田：”ビデオテックス”，電子通信学会誌，Vol.67,No.7,pp.769-773 (1984).

(4) 稲葉：”ＣＡＤとＣＡＭとの結合”，情報処理，Vo.25,No.4,pp.341-348 (1984).

(5) 畠山，小原：”音声蓄積サービス方式”，ＮＴＴ施設，Vol.38,No.6,pp.9 1-96(1986).

(6) 小野，浦野：”マルチメディア通信”，情報処理，Vol.24,No.10,pp.1227 -1232(1983).

(7) J.S.Turner:"FDDI-a tutourial",IEEE Commun.Magazine,Vol.24,No.5,pp. 10-17(1986).

(8) 川原崎，岡田，笹川：”ＡＴＭ通信技術の動向－高速広帯域への展開に向けて－”，電子情報通信学会誌，Vol/71,No.8,pp.806-814(1988).

(9) J.O.Limb and C.Flores : " Description of fasnet - a unidirectional local-area communications network ", Bell Syst.Tech.J., Vol.61, No.7,pp.1413-1440(1982).

(10) 田口，坂下：”ＯＡシステムと文書データベース”，情報処理，Vol.28, No.6,pp.721-729(1987).

(11) 川越，真名垣：”ＣＡＤ/ＣＡＭへのマルチメディアデータベースの応用”，情報処理，Vol.28,No.6,pp.730-739(1987).

(12) 嶋田：”地図・図面情報処理におけるマルチメディアデータベース”，情報処理，Vol.28,No.6,pp.740-755(1987).

(13) 木戸出，垣川：”画像情報処理におけるマルチメディアデータベース”，情報処理，Vol.28,No.6,pp.756-764(1987).

(14) K.A.Lantz and R.F.Rashid: "Virtual Terminal Management in a Multiple Process Enviroment",Proc.7th Symp.Operating Systems Principles (Pacific Grove,Calif.,Dec.10-12)ACM,New Yoek,pp.35-79(1986).

(15) 南：”最近の画像通新技術 ”，画像電子学会誌，Vol.15，No.4，pp.218-223 (1986).

(16) 南：”画像符号化を展望する”，電子情報通信学会誌，Vol.71,No.7，pp.658-662(1988).

(17) 原島，相沢，斉藤：”次世代画像符号化の構想－分析符号化から知的符号化へ－”，電子通信学会画像工学研究会，IE87-1(1987).

(18) 磯崎：”ビデオテックス”，画像電子学会誌，Vol.13,No.3,pp.217-224 (1984).

(19) 中島：”テレビ受像機を端末とした新しい画像情報メディア ”，電子通信学会誌，Vol.63，No.2，pp.175-179 (1980).

(20) 原島：”知的画像符号化と知的通信”，テレビジョン学会誌，Vol.42,No.6,pp.519-525(1988).

(21) 鳥脇：”画像理解のためのディジタル画像処理[Ⅱ]”，pp.66-79,昭晃堂 (1987).

(22) 田村：”コンピュータ画像処理入門”，pp.77-85,総研出版(1985).

(23) M.E.Mortenson:"GEOMETRIC MODELING",pp.30-233,John Wiley & Sons,Inc. (1985).

(24) J.D.Foley，A.Van Dam著（今宮淳美 訳）：”コンピュータ・グラフィックス ”，pp.118-141，日本コンピュータ協会 (1984).

(25) 白井：”パターン理解”，pp.43-96,オーム社(1987).

(26) A.Rosenfeld and J.L.Pfaltz : " Sequential operations in digital picture processing",J.ACM,Vol.13,No.4,pp.471-494(1966).

(27) C.J.Hilditch : " Linear skelton from square cupboards ", in Machine Intelligence, No.4, B.Meltzer and D.Michie, eds.Edinburgh Univ. Press, pp.403-420(1967).

(28) A.Rosenfeld:"Connectivity in digital pictures", J.ACM, Vol.17, No.1, pp.146-160(1970).

(29) R.Steffanelli and A.Rosenfeld:"Some parallel thinning algorithms for digital pictures", J.ACM, Vol.18, No.2, pp.255-264(1971).

(30) 横井, 鳥脇, 福村：" 標本化された二値図形のトポロジカルな性質について "，電子通信学会論文誌(D), Vol.56-D, No.11, pp.662-669 (1973).

(31) 田村：" 細線化法についての諸考察 "，電子通信学会パターン認識・理解研究会, PRL75-66 (1975).

(32) G.Woetzel:"A fast and economic scan-to-line conversion algorithm", SIGGRAPH-ACM Computer Graphics, Vol.12, No.13, pp.125-129 (1978).

(33) T.Pavlidis : " A thinning algorithm for discrete binary images ", Computer Graphics and Image Processing, Vol.73, pp.142-157 (1980).

(34) C.Arcelli:"Pattern thinning by contour tracing ", Computer Graphics and Image Processing, Vol.17, No.2, pp.130-144(1981).

(35) C.Arcelli and G.S.di Baja : " A width-independent fast thinning algorithm", IEEE Tr.PAMI, Vol.7, pp.463-474(1985).

(36) 中山, 木村, 吉田, 福村：" 大規模画像に対する細線化アルゴリズムのパイプライン方式による効率化 "，電子通信学会論文誌(D), Vol.J67-D, No.7, pp.761-767 (1984).

(37) 岡田, 小倉：" ２値図形の細線化に関する一考察 "，昭53年電子通信学会部門別全国大会, No.514 (1978).

(38) 小倉, 岡田：" 線図形の細線化に関する一考察 "，昭54年度電子通信学会総合全国大会, No.1085 (1979).

(39) 小倉, 岡田, 村上：" 線図形の実時間細線化入力法"，電子通信学会画像工学研究会, IE79-16, pp.93-100 (1979).

(40) 岡田，小倉：”ランレングスによる線分分離法とその図形処理への応用”，テレビジョン学会画像表示研究会，IPD49-1，pp.19-24 (1980).

(41) 岡田，小倉，村上 ：”水平・垂直線要素の分類を用いた線順次形細線化法”，電子通信学会論文誌(D)，Vol.J64-D，No.5，pp.403-410 (1981).

(42) 秦野：”スプライン関数”，情報処理，Vol.22,No.1,pp.19-27(1981).

(43) 穂坂，黒田 ：”ＣＡＤにおける曲線曲面の創成について”，情報処理，Vol.17,No.12,pp.1120-1127(1976).

(44) U.Montanari:"A note on minimal length polygonal approximation to a digitized contour",C.ACM,Vol.13,No.1,pp.41-47(1970).

(45) C.T.Zahn and Z.Roskies : " Fourier Descriptors for Plane Closed Curves", IEEE-C, Vol.21, No.3, pp.269-281 (1972).

(46) T.Pavlidis and S.L.Holowitz : "Segmentation of plane curves",IEEE Trans.Comput.,Vol.C-23,pp.860-870(1974).

(47) C.M.Williams : " An efficient algorithm for the piecewise linear approximation of planar curves ", Computer Graphics and Image Processing,Vol.8,pp.286-293(1978).

(48) C.M.Williams :" Bounded straight-line aproximation of digitized planar curves and lins",Computer Graphics and Image Processing,Vol.16,pp.370-381(1981).

(49) 名倉：”手書き線図形の直線と円弧による近似”，電子通信学会論文誌(D)，Vol.J64-D，No.9，pp.839-845 (1981).

(50) 佐藤幸男：”平面曲線の最適折線近似”，電子通信学会論文誌(D)，Vol.J65-D，No.9，pp.1145-1150 (1982).

(51) 安居院，飯塚，中嶋：”原図形に忠実な直線近似法”，電子通信学会論文誌(D)，Vol.J68-D，No.8，pp.1539-1540 (1985).

(52) K.Ramachandran : " Coding method for vector representation of engineering drawings",Proc.IEEE,Vol.68,No.7,pp.813-817(1980).

(53) 星野，小倉，河久保：”ラン端点の線近似によるカラー図形の符号化 ”，電子通信学会論文誌(D)，Vol.J85-D，No.4，pp.615-622 (1985).

(54) 秦，富田，大西，中田：”多色画像の図形コマンド符号化方式 ”，電子情報通信学会論文誌(D)，Vol.J71-D，No.2，pp.306-314 (1988).

(55) 村上，岡田，土屋：”図形の直線および円弧成分の一抽出法 ”，電子通信学会画像工学研究会，IE73-26 (1973).

(56) 村上，岡田，土屋：”線図形の入力法に関する一考察 ”，電子通信学会論文誌(D)，Vol.63-D，No.2，pp.117-124 (1976).

(57) 星野，岡田，村上：”簡易図形のコマンド表現法について ”，昭54年度電子通信学会総合全国大会，No.1093 (1979).

(58) 小倉，岡田：”面図形のコマンド符号化法の検討 ”，昭56年度電子通信学会総合全国大会，No.1034 (1981).

(59) 黒崎：”高速漢字プリンタシステムについて ”，情報処理，Vol.16，No.9，pp.802-807 (1975).

(60) 森：”ドット漢字パターンマトリクスの次数変換法 ”，電子通信学会論文誌(D)，Vol.60-D，No.10，pp.801-808 (1977).

(61) 渡部，筒井，加藤，及川，石山 ：”漢字パターンの拡大・縮小法の一考察 ”，電子通信学会画像工学研究会，Vol.79，No.16，IE79-2 (1979).

(62) 井上，木村，武川：”漢字パターンの拡大・縮小法 ”，電子通信学会画像工学研究会，Vol.79，No.16，IE79-1 (1979).

(63) 斉藤，松葉，杉山 ：”インクジェット式漢字プリンタ用画素形文字構成法 ”，電子通信学会論文誌(D)，Vol.J62-D，No.11，pp.734-741 (1979).

(64) 小川，中根，池沢：”漢字パターン変換処理の一検討 ”，電子通信学会論文誌(D)，Vol.J65-D，No.2，pp.234-241 (1982).

(65) 西川，長田：”輪郭表現による文字パターンの拡大・縮小方式 ”，昭61年度電子通信学会総合全国大会，No.1558 (1986).

(66) 田中，乙田，岡田：”文字フォントの性質に着目した任意倍率文字サイズ変換法”，電子通信学会論文誌(D)，Vol.J69-D,No.3,pp.460-466 (1986).

(67) J.E.Bresenham : " Algorithm for computer control of a digital plotter",IBM System Journal,Vol.4,pp.25-30(1965).

(68) J.E.Bresenham:"A linear algorithm for incremental digital display of circular arcs",C.ACM,Vol.20,pp.100-106(1977).

(69) 釜江，小杉，星野 ：”図形のドット表示”，電子通信学会論文誌(A)，Vol.56-A，No.7，pp.401-408 (1973).

(70) 末永，釜江，小林：”新しい直線・円弧発生アルゴリズム（改良型変位比較法）とその能力評価”，電子通信学会画像工学研究会，IE77-88 (1978).

(71) Y.Suenaga,T.Kamae and T.Kobayashi:"A high-speed algorithm for the generation of straight lines and circular arcs",IEEE Trans.Comput., Vol.C-28,pp.728-736(1979).

(72) F.C.Crow : " The aliasing problem in computer-generated shaded images",C.ACM,Vol.20,No.11,pp.799-805(1977).

(73) F.C.Crow : " A comparism of antialiasing techniques ",IEEE CG & A, Vol.1,pp.40-47(1981).

(74) F.Crow : " The Use of Grayscale for Improved Raster Display of Vectors and Characters", SIGGRAPH-ACM Computer Graphics, Vol.12, No.3,pp.1-5 (1978).

(75) H.Fuchs and J.Barros : " Efficient Generation of Smooth Line Drawings on Video Displays", SIGGRAPH-ACM Computer Graphics, Vol. 13, No.2, pp.260-269 (1979).

(76) S.Gupta : " Filtering Edges for Gray-Scale Display ", SIGGRAPH-ACM Computer Graphics, Vol.15, No.3, pp.1-5 (1981).

(77) 西田,中前:"カラーディスプレイにおけるスムーズな線分の発生法",情報処理学会論文誌,Vol.22,No.6,pp.505-511(1981).

(78) M.L.V.Pitteway and D.J.Watkinson:"Bresenham's algorithm with gray scale",C.ACM,Vol.23,pp.625-626(1980).

(79) 岡田，釜江，森：”ドットによる直線の発生”，１９７７年テレビジョン学会全国大会，No.11-9，pp.233-234（1977).

(80) 岡田，釜江：”中間調ドットを用いた円弧の発生法”，昭53年度電子通信学会総合全国大会，No.1114（1978).

(81) 岡田，釜江：”直線と円弧の多値ドット表示法”，電子通信学会論文誌(D)，Vol.61-D，No.7，pp.489-495（1980).

(82) 小倉，星野，河久保，岡田：”線画像入力処理法”，研究実用化報告，Vol.30，No.11，pp.2675-2688（1981).

(83) 田村：”コンピュータ画像処理入門”，pp.69-73，総研出版（1985).

(84) A.Rosenfeld著(石田,島村,佐藤 共訳)：”電子計算機による画像処理”，pp.166-167，共立出版（1971).

(85) 安田：”ファクシミリの基礎と応用”，電子通信学会編（1977).

(86) 岸野，星野，河久保，米沢：”ランレングス符号化方式を用いたカラー図形ファイル装置の検討”，電子通信学会画像工学研究会，IE77-75（1978).

(87) 星野，岡田，名倉，村上：”コマンド形図形ファイル方式の検討”，電子通信学会画像工学研究会，IE78-75，pp.25-32（1978).

(88) 星野，小倉，河久保，岡田："ランレングス符号を導入したカラー図形作成装置",電子通信学会論文誌（D），Vol.J65-D,No.2,pp.210-217（1982).

(89) 西川：”磁気記録メモリ”，電子通信学会誌，Vol.67,No.11，pp.1178-1183(1984).

(90) 吉田：”光ディスクメモリ”，電子通信学会誌，Vol.67,No.11,pp.1215-1222(1984).

(91) 小倉，星野，河久保，岡田，村上：”情報提供用画面作成装置”，テレビジョン学会画像システム研究会，IT45-3，pp.13-18（1980).

(92) 小倉，岡田：”ファクシミリを用いた線図形入力処理の一検討”，１９７９年テレビジョン学会全国大会，No.11-4，pp.289-290 (1979).

(93) 星野，河久保，村上：”ランレングス符号による簡易画像の彩色法”，電子通信学会論文誌(D)，Vol.63-D，No.10，pp.899-906 (1980).

(94) 岡田，小倉，川名：”簡易画像作成装置の文字入力法”，１９８０年テレビジョン学会全国大会，No.15-14，pp.341-342 (1980).

(95) ”ＪＩＳハンドブック情報処理”，pp.1591-1669，日本規格協会 (1987).

(96) 大沢，坂内：”多次元データ構造を用いた図面処理 － 図形のベクトル化－”，電子通信学会論文誌(D)，Vol.J68-D,No.4,pp.845-852 (1985).

(97) 吉田 :”線図形の認識と理解”，電子情報通信学会誌，Vol.71,No.11,pp.1192-1197 (1988).

付録１　補題３．２の証明

画素数ｎの画素列の分割アルゴリズムから，t_kとt_k'の間に

$$t_k' - t_k \leqq 1$$

が成立するとき，１つの極大図形成分が抽出されたことが分かる。ここで，画素列を一定の内分比（ｐ：ｑ）で繰り返し分割を行ったとし，ｉとｊをそれぞれ部分画素列が１つの図形成分として認識できた回数，および認識できなかった回数，Ｎを総分割回数（＝ｉ＋ｊ）とする。このとき，

$$t_k' - t_k = n\{q/(p+q)\}^i \cdot \{p/(p+q)\}^j \qquad (A.1)$$

が成り立つ。このことから，

（１）ｑ／ｐ＜１の場合

$$t_k' - t_k < n\{p/(p+q)\}^N \qquad (A.2)$$

となるので，

$$N \leqq \log_{(1+q/p)} n \qquad (A.3)$$

が成り立つ。

（２）ｑ／ｐ≧１の場合

同様に考えて，

$$N \leqq \log_{(1+p/q)} n \qquad (A.4)$$

が成り立つ。

式（Ａ．３）と（Ａ．４）の右辺はｑ／ｐに関し，それぞれ単調減少，単調増加となる。したがって，ｑ／ｐ＝１の内分比で分割するのが最も効率がよく，そのときＮはたかだか$\log_2 n$となる。

（証明終り）

付録２　性質５．１の証明

直線ｆ（ｘ，ｙ）＝ａｘ＋ｂｙ＋ｃ＝０に対し，座標値の集合Ａ，Ｂ，Ｃを以下のように定義する。但し，Ｆ（Ｐ）は正規化変位を表す。

$$A = \{P \mid |F(P)| < t\}$$

$$B = A \cap \{P \mid F(P) \geqq 0\}$$

$$C = A \cap \{P \mid F(P) < 0\}$$

いま，任意のx_0に対して，

$$F(x_0, y_1) = \max_{P \in B} \{F(P)\} < t$$

$$F(x_0, y_2) = \min_{P \in C} \{F(P)\} > -t$$

とする。これらをy_1，y_2について解くと，正規化変位の定義から，

$$y_1 < t/2 - (ax_0 + c)/b$$

$$y_2 > -t/2 - (ax_0 + c)/b$$

となる。明らかに，$y_1 > y_2$であるから

$$0 < y_1 - y_2 < t$$

である。故に，ｙ軸方向のドットの個数はたかだか［ｔ］＋１である。なお，［ｘ］はｘを越えない最大の整数を表す。

（証明終り）

付録３　性質５．２の証明

円弧 $f(x,y)=x^2+y^2-r^2=0$ に対し，座標値の集合Ａ，Ｂ，Ｃを以下のように定義する。但し，Ｆ(Ｐ)は正規化変位を表す。

$$A=\{P \mid |F(P)|<t\}$$

$$B=A\cap\{P \mid F(P)\geqq 0\}$$

$$C=A\cap\{P \mid F(P)<0\}$$

いま，P_1 と P_2 を，

$$F(P_1)=\max_{P\in B}\{F(P)\}<t$$

$$F(P_2)=\min_{P\in C}\{F(P)\}>-t$$

とする。原点を中心とし点 P_1，P_2 を通る円弧の半径 r_1，r_2 は，それぞれ

$$r_1<(r^2+tr)^{1/2}$$

$$r_2>(r^2-tr)^{1/2}$$

を満足する。これより，$2<r_1-r_2<\sqrt{2}\,t$ となることが分かる。

故に，半径方向のドットの個数は $[\sqrt{2}\,t]+1$ を越えない。なお，［ｘ］はｘを越えない最大の整数を表す。

（証明終り）

図　表

分野＼年代		1984	1985～1989
入力処理技術	線図形入力処理	(1966) 逐次処 (A. Ros (1984) パイプライン処理による実時間細線化(中山, 他) (1985) ラスタ走査回数を削減した 細線化法(C. Arcelli, 他) (1 細化(T. Pavlidis) (C. ら細線化結果の リによ 岡田, 他)	
	図形コマンド符号化処理	着目した 名倉) による 近似(佐藤, 他) リ信号の	(1985) 平均距離最小化法による 曲線の折れ線近似(安居院, 他) (1985) ラン端点線近似による 彩色図形符号化 (星野, 他) (1988) 多角形・円弧近似 による彩色図形の 符号化(秦, 他)
出力処理技術	任意サイズ漢字発生法	ーン排他的 (小川, 他)	(1986) 文字輪郭表現による 拡大縮小法(西川, 他) (1986) 文字パターン前処理機能を 付加した任意サイズ変換法 (岡田, 他)
	図形発生法	(1965) 線分発 (J. E. I ンチエリアシング(H. Fuch, 他) よる輝度変調法(M. L. Pitteway, 他) 片計算による線分アンチエリアシング(S. Gupta) 片計算を改良した線分アンチエリアシング(西田, 他) 凡 づいた線 ジング	

分野＼年代	～1974	1975～1979	1980～1984	1985～1989
入力処理技術 線図形入力処理	(1966) 逐次処理と並列処理の体系づけ (A. Rosenfeld, 他) (1967) 細線化アルゴリズムの原理 (C. J. Hilditch) (1970) ディジタル画像の性質を体系化 (A. Rosenfeld) (1971) 並列処理による細線化法 (R. Steffanelli, 他) (1973) 連結数によるアルゴリズムの定式化 (横井, 他) (1975) 細線化アルゴリズムの比較検討 (田村)	(1978) メモリを削減した実時間細線化 (G. Woetzel, 他)	(1980) 輪郭追跡による処理の高速化 (T. Pavlidis) (1981) 輪郭追跡処理による細線化結果の評価 (C. Arcelli) (1981) 3走査ラインメモリによる実時間細線化 (岡田, 他)	(1984) パイプライン処理による実時間細線化 (中山, 他) (1985) ラスタ走査回数を削減した細線化法 (C. Arcelli, 他)
入力処理技術 図形コマンド符号化処理	(1970) 周長最小化法による多角形近似 (U. Montanari) (1972) フーリエ記述子による線図形表現 (C. T. Zahn, 他) (1974) 線分個数最小化法による多角形近似 (T. Pavlidis, 他) (1976) テンプレートマッチングによる線分・円弧近似 (岡田, 他)	(1978) 最大長線分を探索する多角形近似 (C. M. Williams)	(1981) 曲線の曲率変化に着目した直線・曲線分離法 (名倉) (1982) 周長最大法による最適折れ線近似 (佐藤, 他) (1980) 線分近似によるファクシミリ信号の符号化 (K. Ramachandran)	(1985) 平均距離最小化法による曲線の折れ線近似 (安居院, 他) (1985) ラン端点線近似による彩色図形符号化 (星野, 他) (1988) 多角形・円弧近似による彩色図形の符号化 (秦, 他)
出力処理技術 任意サイズ漢字発生法	(1975) 整数倍補間拡大法 (黒崎)	(1977) 線分比例配置法 (森) (1979) 線分比例配置法の高速化 (渡部, 他) (1979) サブパターン変換テーブル法 (井上, 他) (1979) 画素間論理演算法 (斉藤, 他)	(1982) 差分パターン排他的論理和法 (小川, 他)	(1986) 文字輪郭表現による拡大縮小法 (西川, 他) (1986) 文字パターン前処理機能を付加した任意サイズ変換法 (岡田, 他)
出力処理技術 図形発生法	(1965) 線分発生法の原理 (J. E. Brezenham) (1973) 変位比較による線分・円弧発生法 (釜江, 他)	(1977) 線分発生原理の円弧への拡張 (J. E. Brezenham) (1978) 改良型変位比較法 (末永, 他) (1977) アンチエリアシングの基本手法 (F. C. Crow) (1978) 画素内挿によるｱﾝﾁｴﾘｱｼﾝｸﾞ (F. Crow) (1979) 正方画素素片計算による線分ｱﾝﾁｴﾘｱｼﾝｸﾞ (H. Fuch, 他) (1980) 線分と画素の距離による輝度変調法 (M. L. Pitteway, 他) (1981) 円形画素素片計算による線分ｱﾝﾁｴﾘｱｼﾝｸﾞ (S. Gupta) (1981) 正方画素素片計算を改良した線分ｱﾝﾁｴﾘｱｼﾝｸﾞ (西田, 他) (1980) 変位による輝度変調に基づいた線分と円弧のアンチエリアジング (岡田, 他)	

凡例

本研究内容の位置

図１．１　研究経過

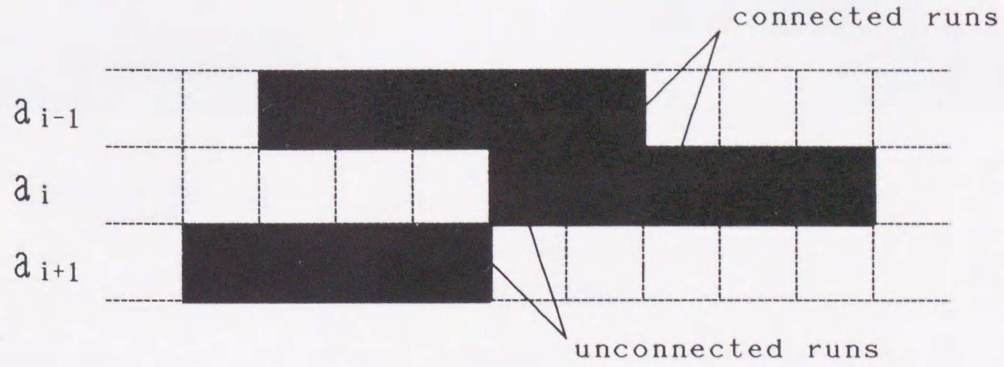
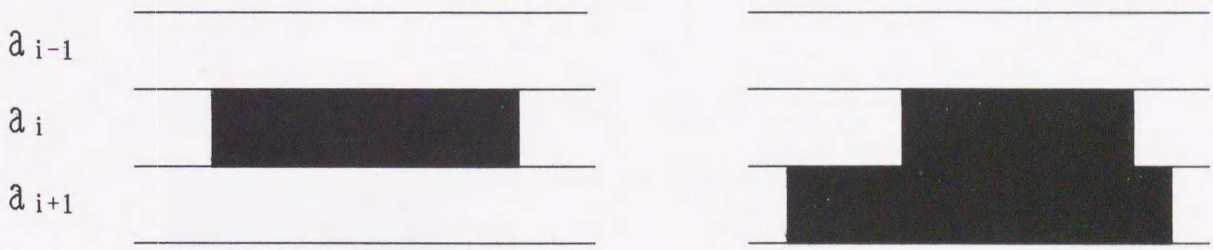
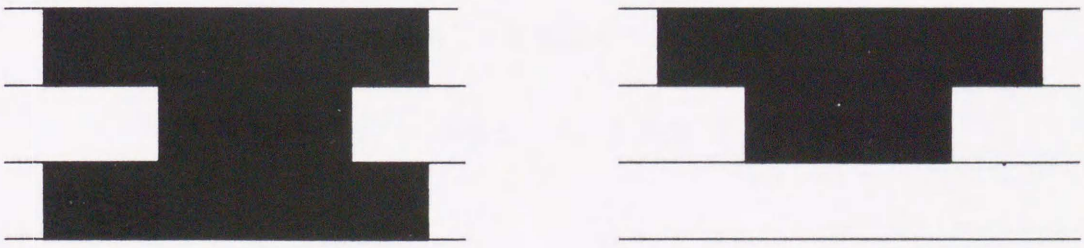


図２．１　ランの連結

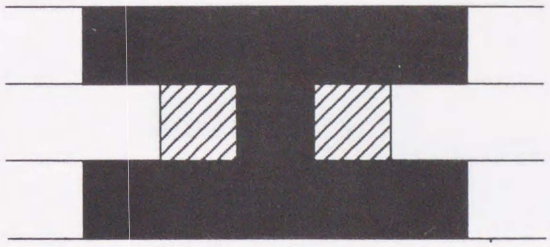


（ａ）連結状態１　　（ｂ）連結状態２

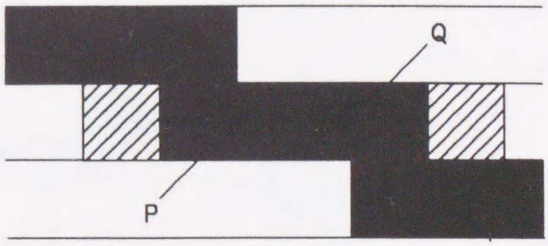


（ｃ）連結状態３　　（ｄ）連結状態４

図２．２　隣接走査線間のランの連結状態

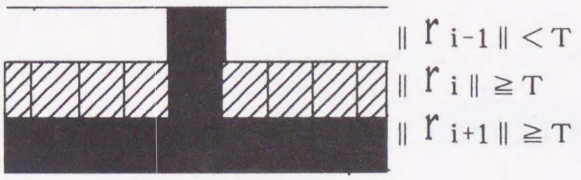


（ａ）共通部有りの場合

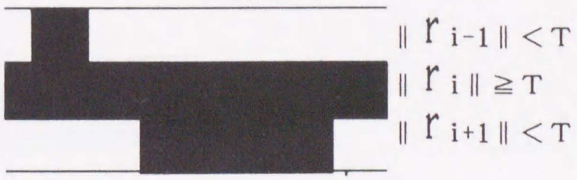


（ｂ）共通部無しの場合

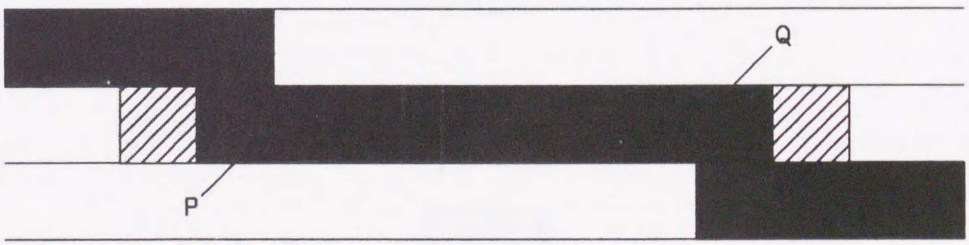
図２．３　　手順３．１の説明図



（ａ）共通部有りの場合



（ｂ）共通部無しの場合（その１）



（ｃ）共通部無しの場合（その２）

図２．４　　手順３．２の説明図

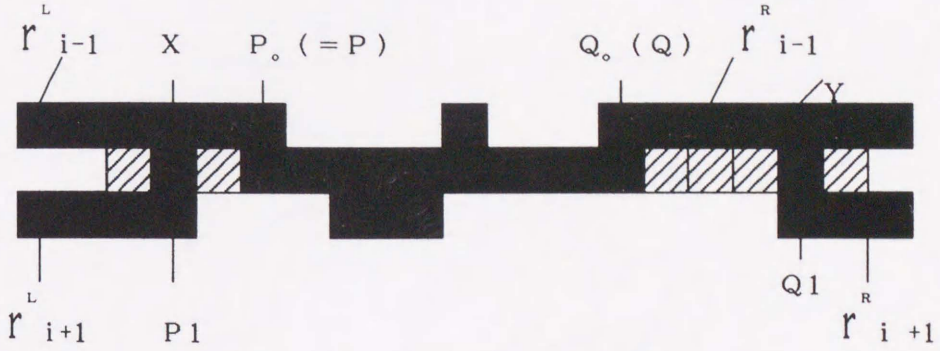
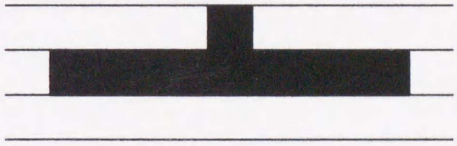
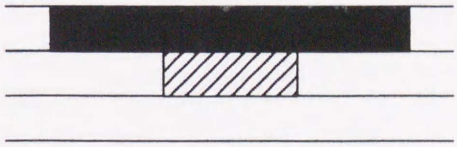


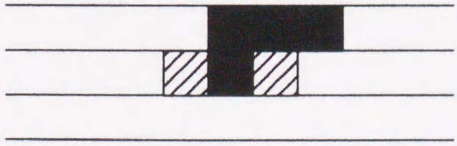
図２．５　　手順３．３の説明図



(a) $\| r_i \| \geq T$

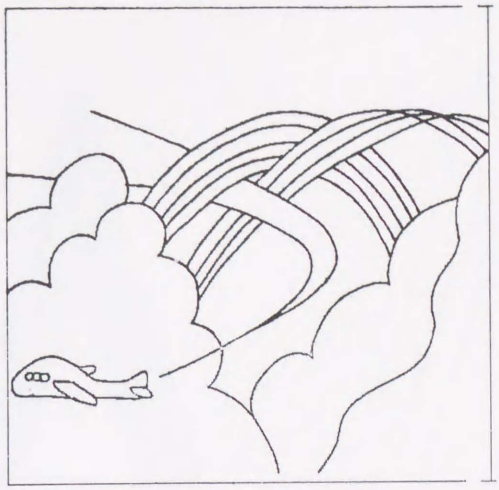


(b) $\| r_{i-1} \| \geq T$, $\| r_i \| < T$

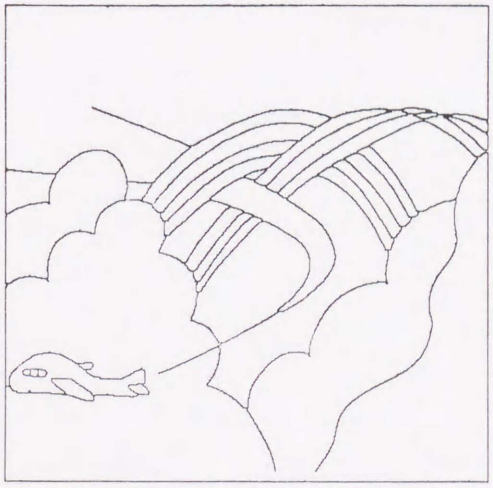


(C) $\| r_{i-1} \| < T$, $\| r_i \| < T$

図２．６　　手順４の説明図

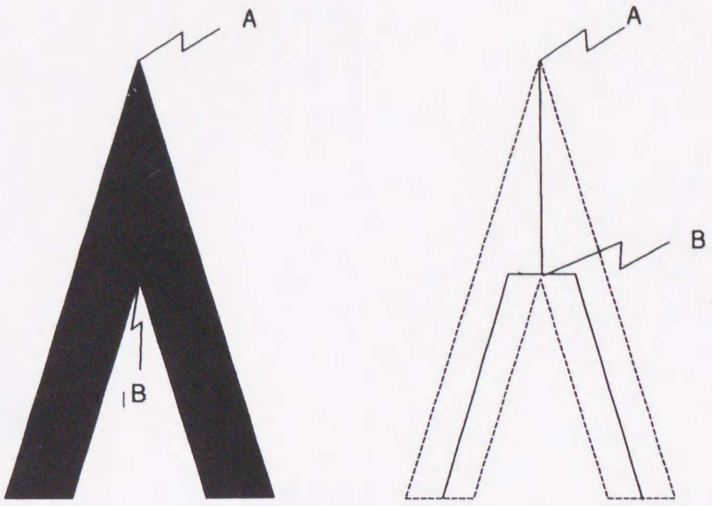


（ａ）　原図形



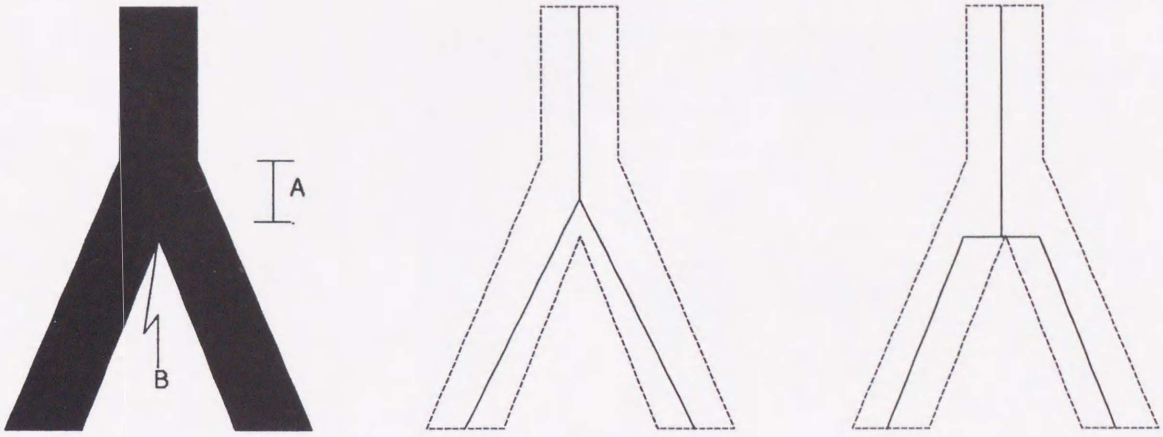
（ｂ）　細線化図形（Ｔ＝２０）

図２．７　　細線化処理例



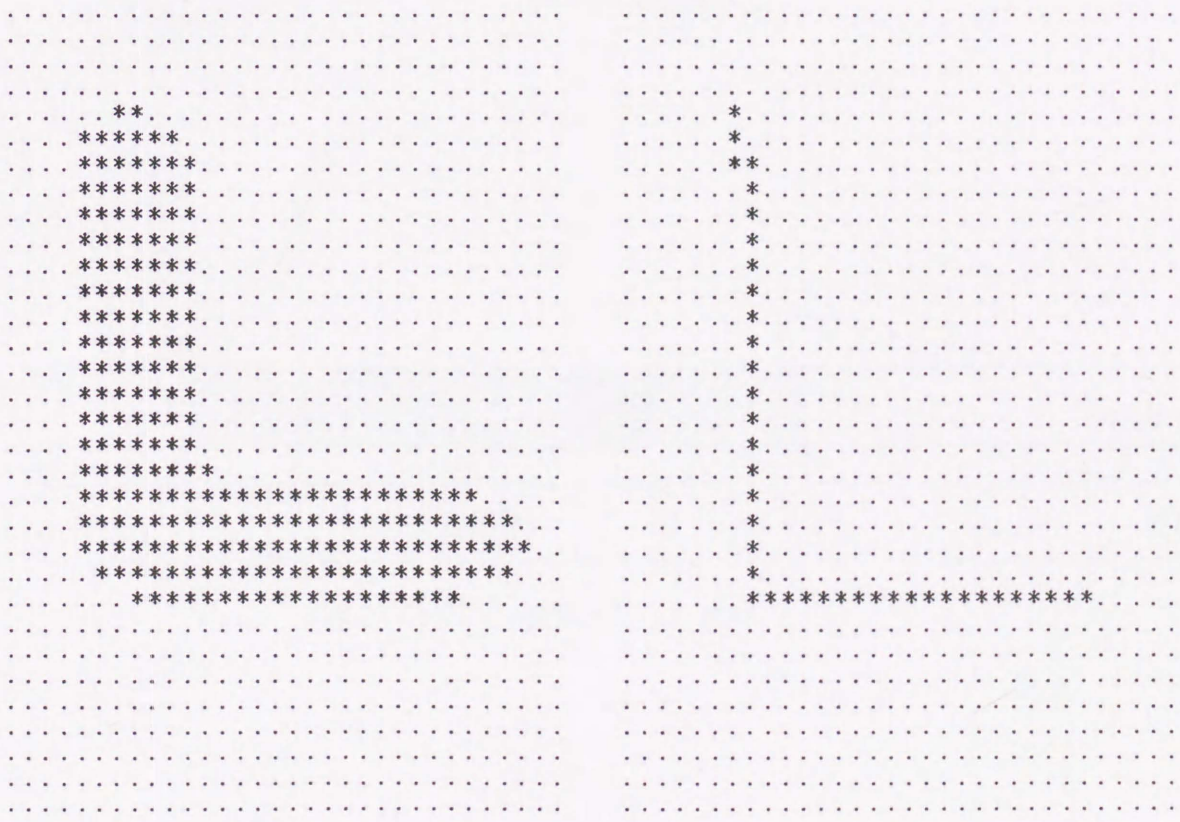
（ａ）原図形　　（ｂ）細線化結果

図２．８　　短線分発生の説明図



（ａ）　原図形　　（ｂ）理想的な抽出例　　（ｃ）細線化結果

図２．９　　歪現象の説明図（その１）



（ａ）　原図形　　（ｂ）細線化結果

図２．１０　　歪現象の説明図（その２）

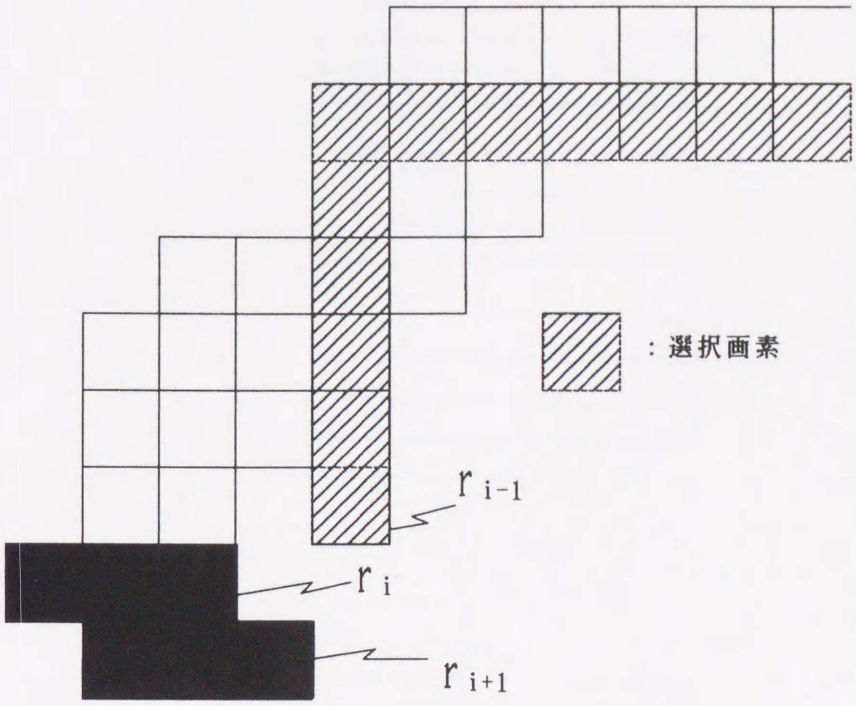


図２．１１　　連結状態の変化

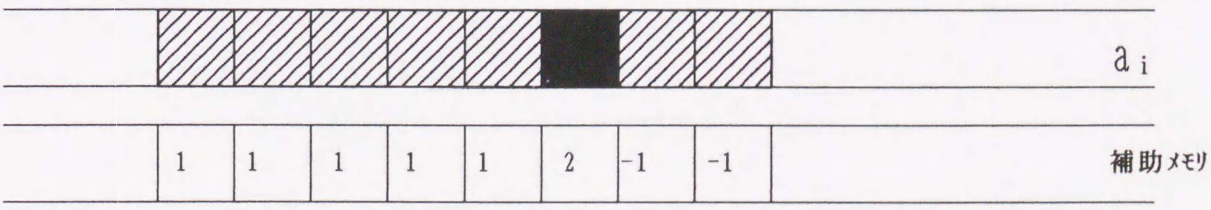
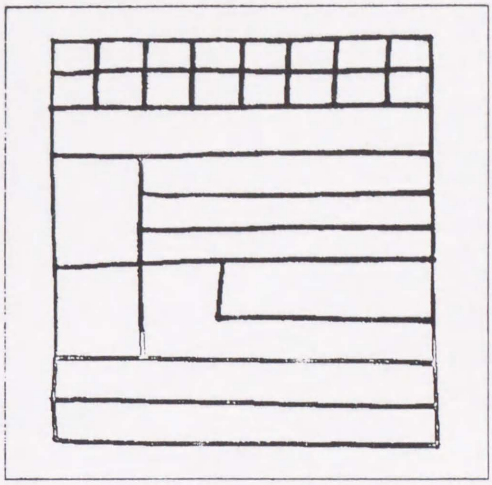
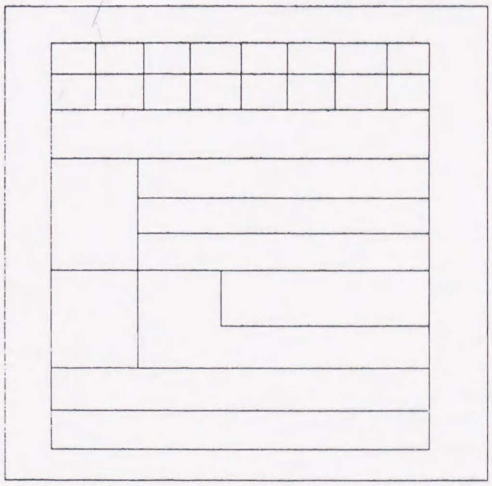


図２．１２　　補助メモリの構成

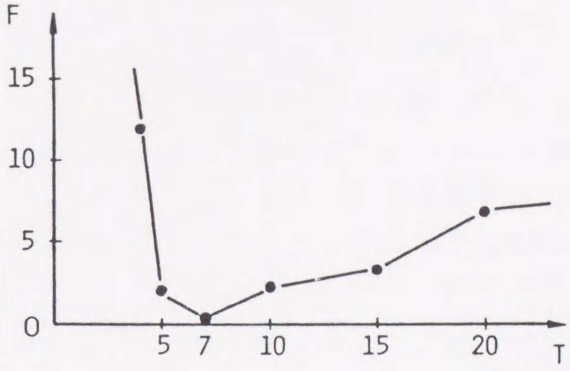


（ａ）　原画



（ｂ）　処理結果

図２．１３　　水平垂直線化処理の例



（ａ）　評価関数Ｆとしきい値Ｔの関係

（ｂ）　分類結果（Ｔ＝７）

図２．１４　　縦横線の分類例

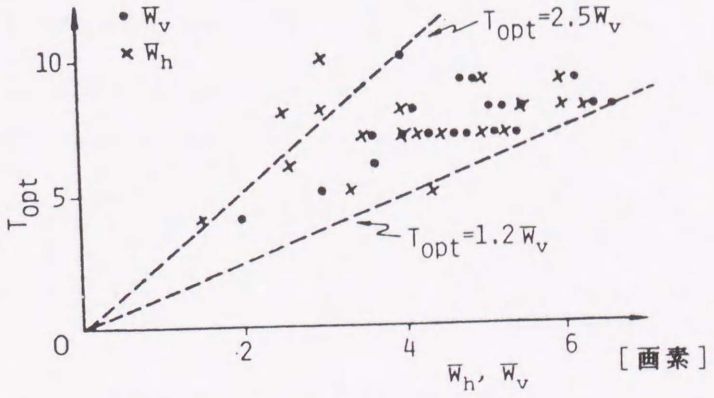


図２．１５　　最適しきい値T_{opt}と平均線幅W_h，W_vの関係

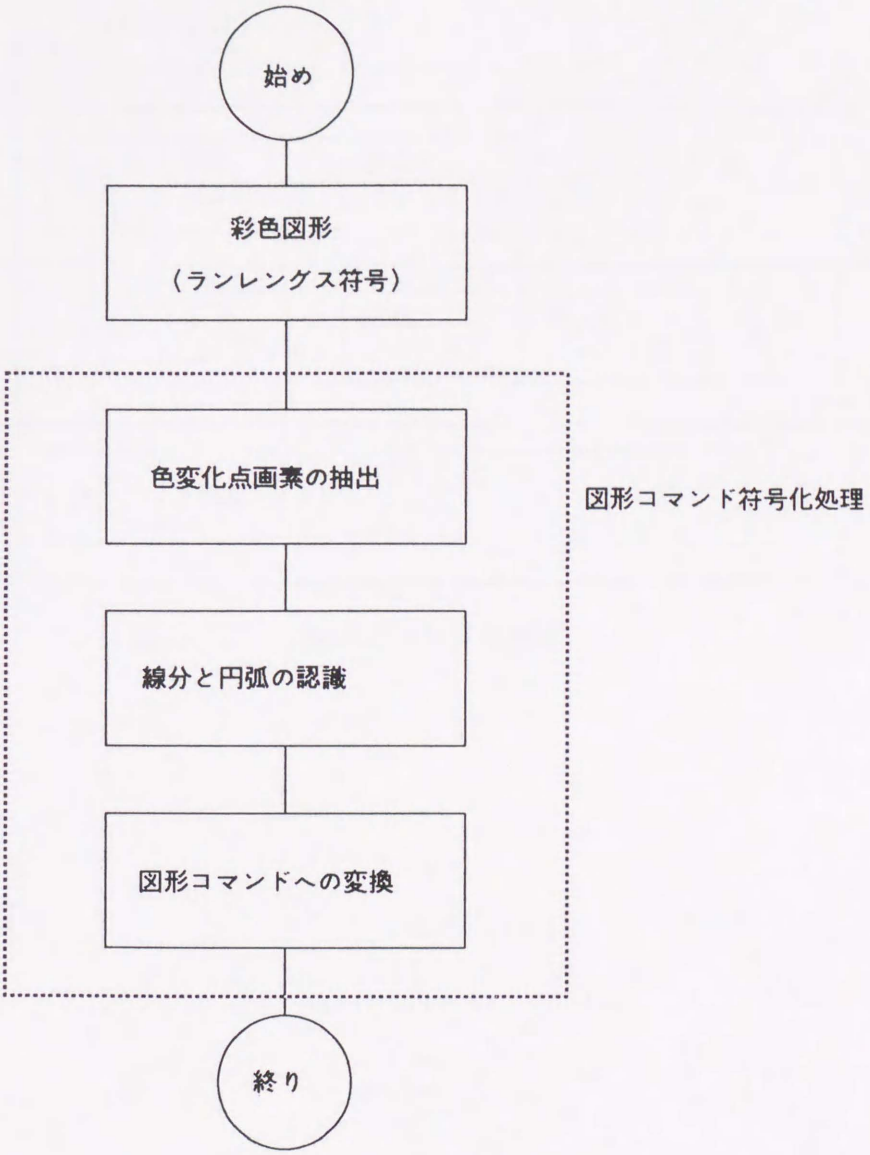


図３．１　　図形コマンド符号化処理の手順

種類	符号形式
点	OP | 点（x，y） | 色
線分	OP | 始点（x0, y0） | 終点（x1, y1） | 色
円弧	OP | 始点（x0, y0） | 終点（x1, y1） | 中心点（x2, y2） | 色
円	OP | 半径（r） | 中心点（x，y） | 色

図３．２　　図形コマンドの構成

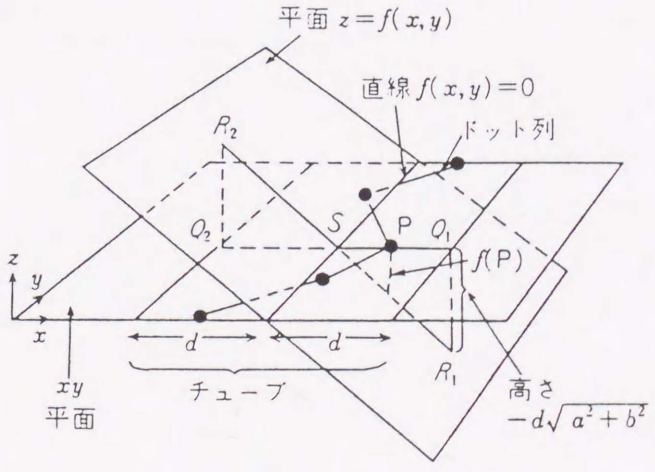


図３．３　　線分判別式の幾何学的意味

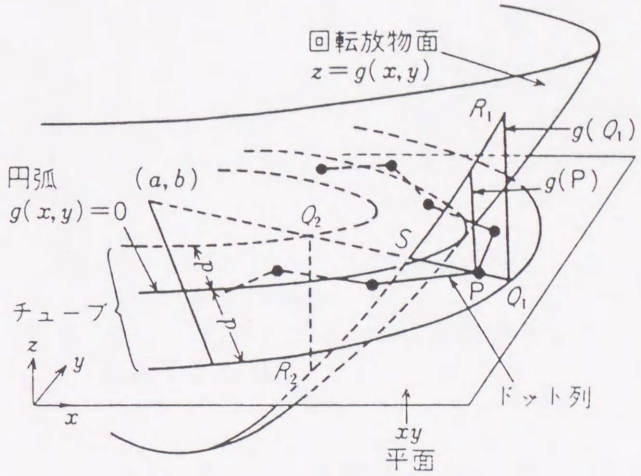
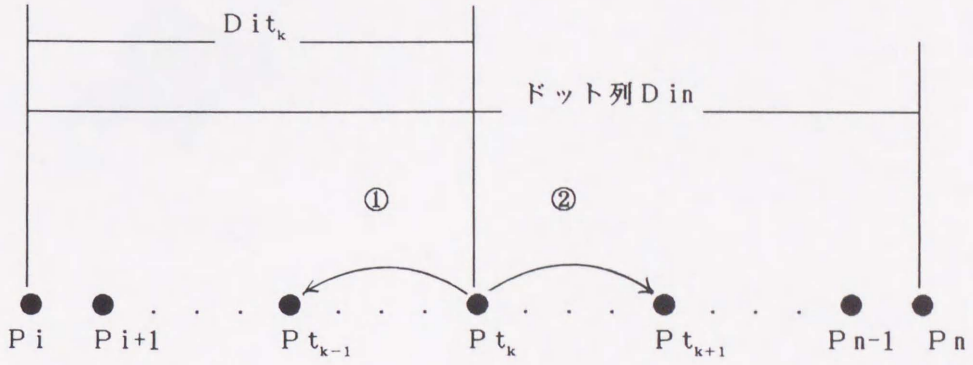
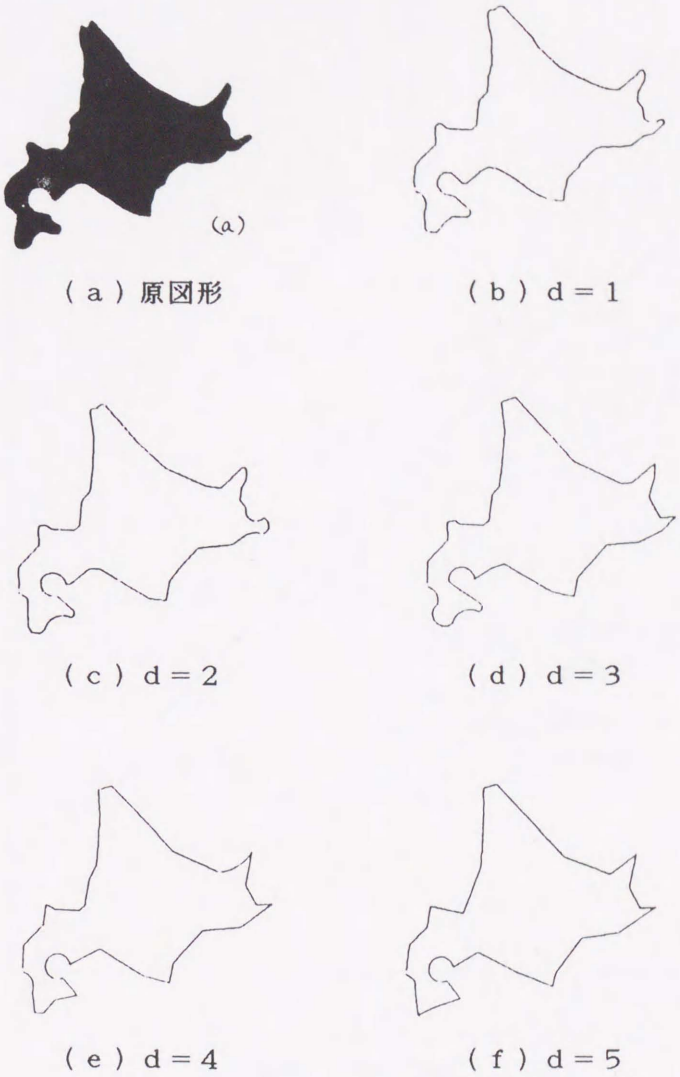


図３．４　　円弧判別式の幾何学的意味



① ドット列Dit$_k$が１つの図形成分として
認識できない場合
② ドット列Dit$_k$が１つの図形成分として
認識できる場合

図３．５　　画素列（ドット列）の分割



（ａ）原図形　　（ｂ）ｄ＝１

（ｃ）ｄ＝２　　（ｄ）ｄ＝３

（ｅ）ｄ＝４　　（ｆ）ｄ＝５

図３．６　線分と円弧の認識例

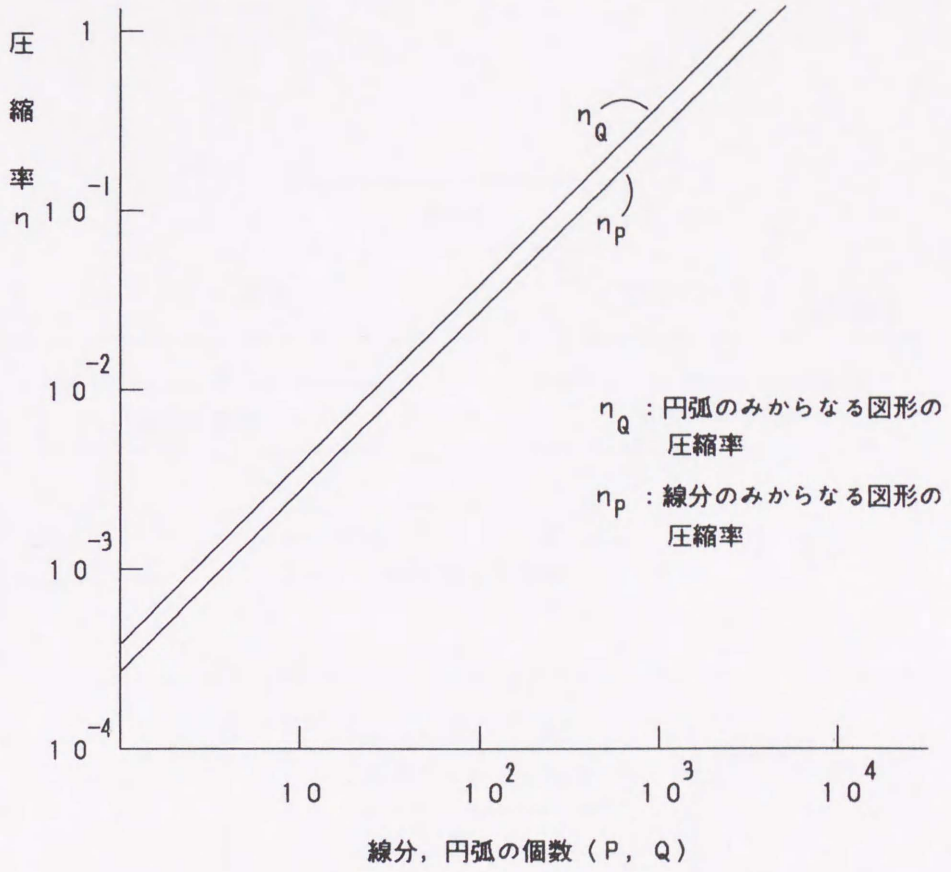

圧縮率 η
1
10^-1
10^-2
10^-3
10^-4
10
10^2
10^3
10^4
η_Q
η_P
η_Q ：円弧のみからなる図形の圧縮率
η_P ：線分のみからなる図形の圧縮率
線分，円弧の個数（P，Q）

図３．７　　データ圧縮率

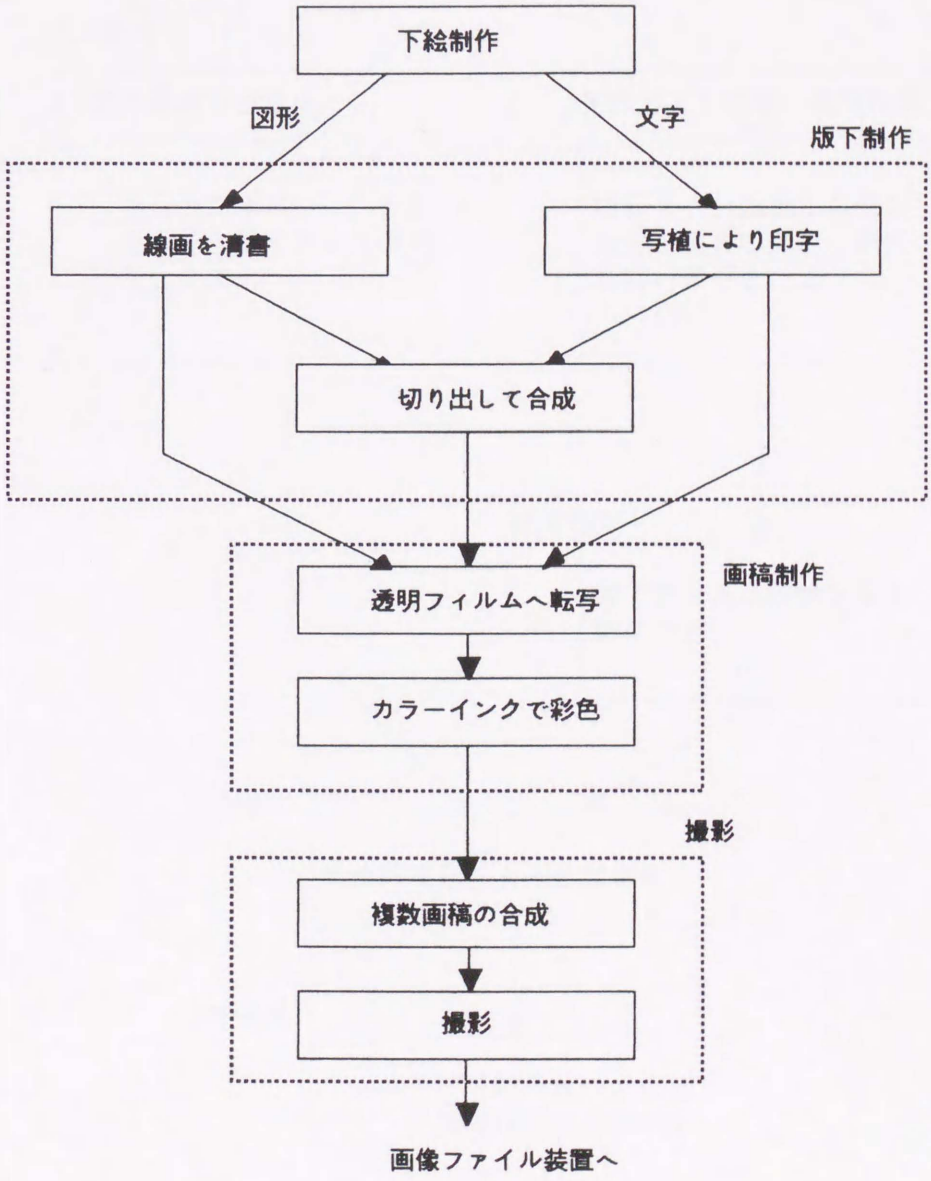


図４．１　　手作業による彩色図形作成法

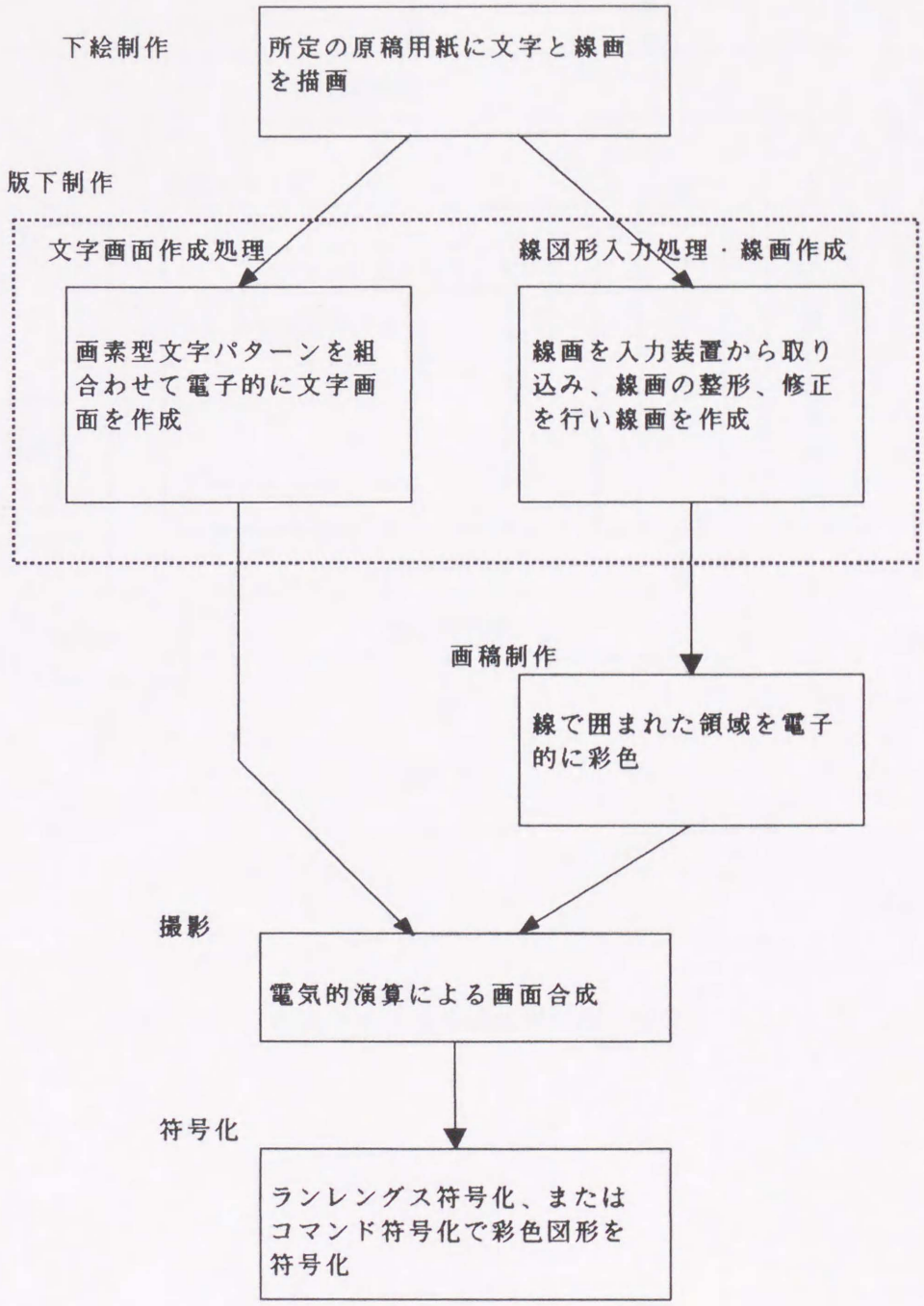


図４．２　　画像ファイル作成方式の基本構想

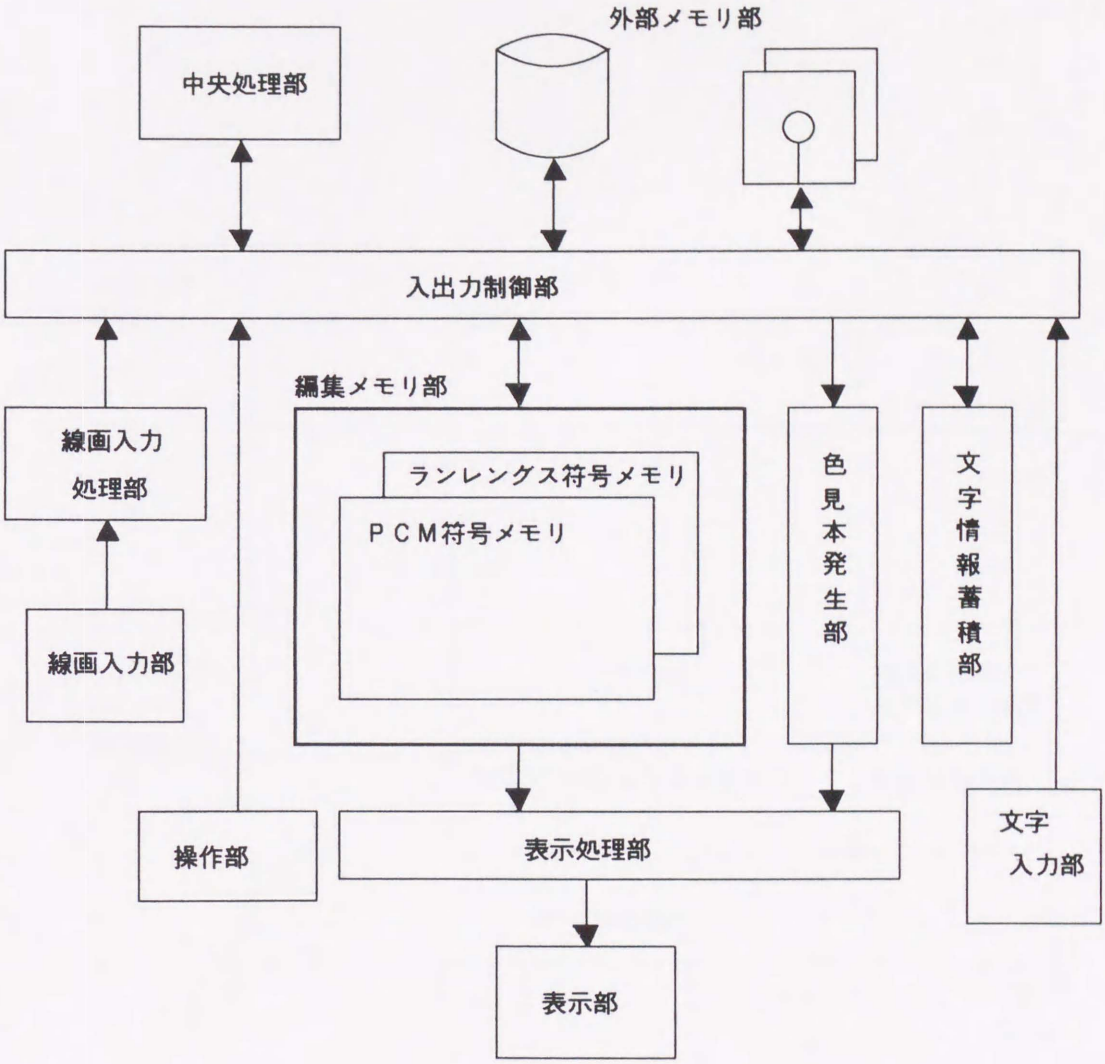


図４．３　　画像ファイル作成装置の基本構成

<table>
<tr><td>彩色図形
作成モード</td><td>汎用モード</td><td>線幅均一化
モード</td><td>水平垂直化
モード</td><td>領域指定
モード</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="3">線図形入力
処理</td><td rowspan="2"></td><td colspan="2">太め処理</td><td>細線化処理/
水平垂直化処理</td></tr>
<tr><td colspan="2">細線化処理/水平垂直化処理</td><td>線幅識別処理</td></tr>
<tr><td colspan="4">密度変換処理</td></tr>
</table>

図４．４　　彩色図形作成モードと線図形入力処理の関係

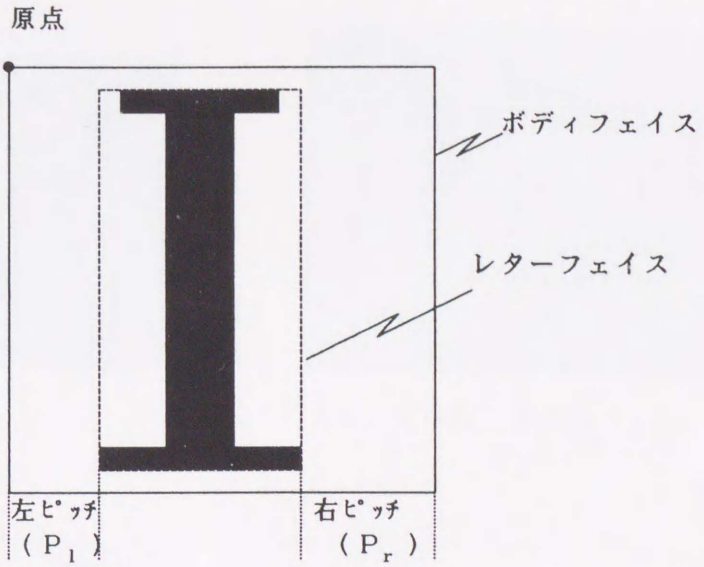


図４．５　　画素型文字パターンとピッチの定義

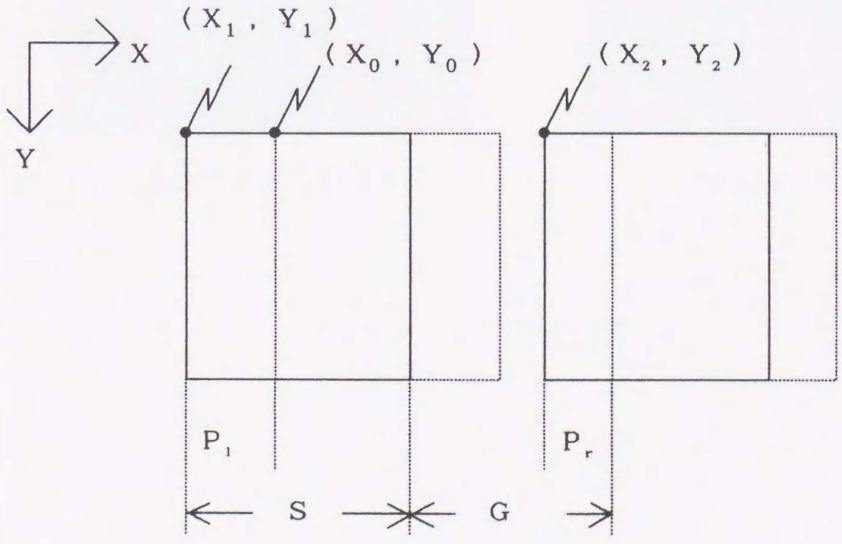
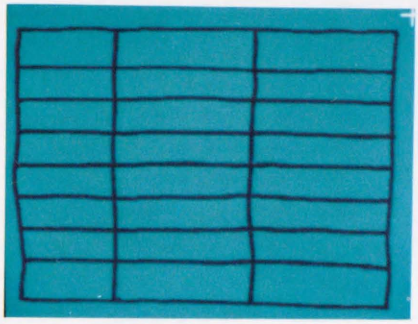


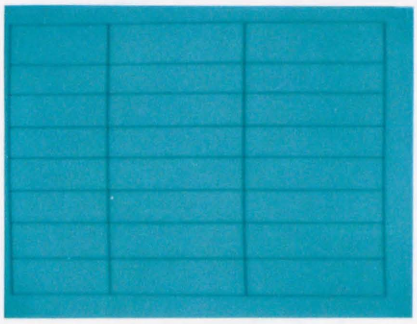
図４．６　　文字表示位置の制御法



（ａ）入力線画

県　名	人口（千人）	男女比
茨　城	2342	98.1
栃　木	1698	96.7
群　馬	1756	95.8
埼　玉	4821	102.2
千　葉	4149	102.0
東　京	11674	102.7
神奈川	6398	105.7

（ｂ）文字画面作成例



（ｃ）水平垂直化処理結果

県　名	人口（千人）	男女比
茨　城	2342	98.1
栃　木	1698	96.7
群　馬	1756	95.8
埼　玉	4821	102.2
千　葉	4149	102.0
東　京	11674	102.7
神奈川	6398	105.7

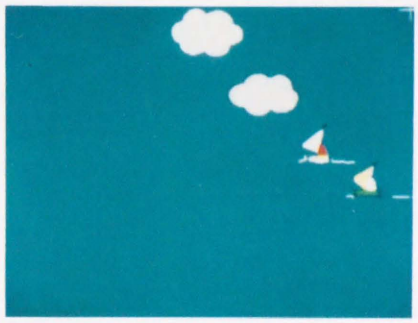
（ｄ）合成画面

図４．７　　表の作成例

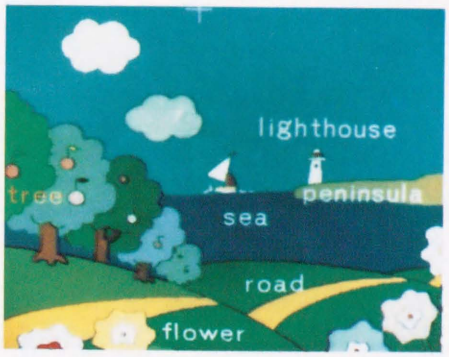
（ａ）入力線画

（ｂ）細線化処理結果

（ｃ）彩色処理（彩色画Ａ）



（ｄ）彩色画Ｂ



（ｅ）合成処理と文字入力の結果

図４．８　　イラスト図形の作成例

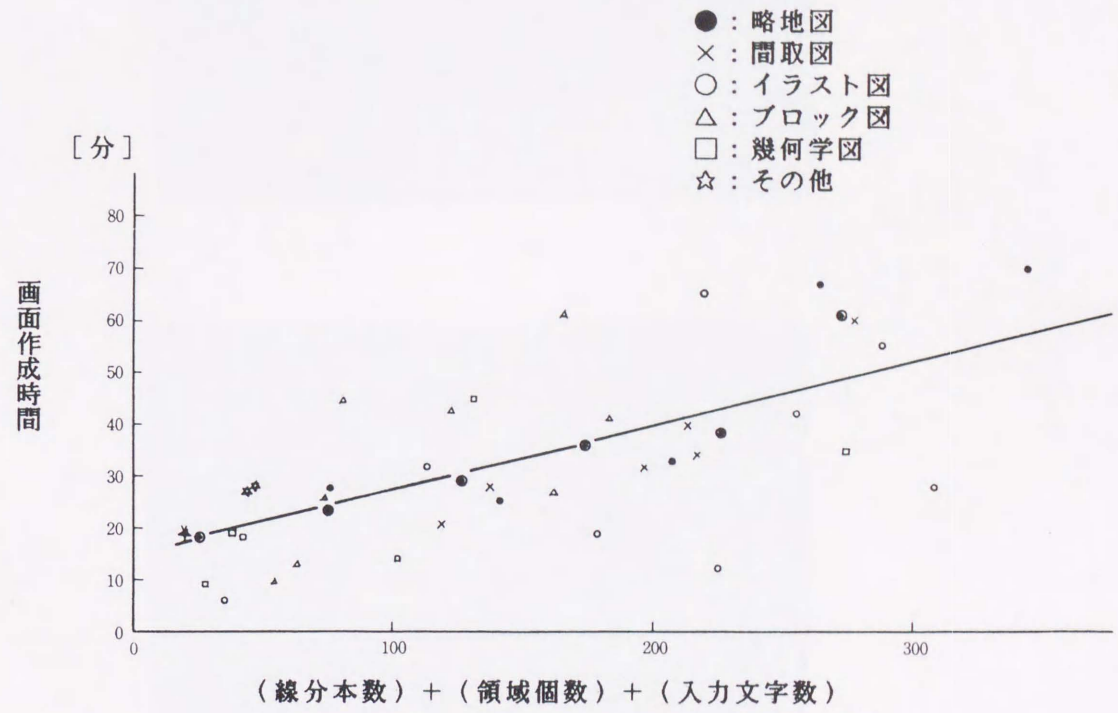
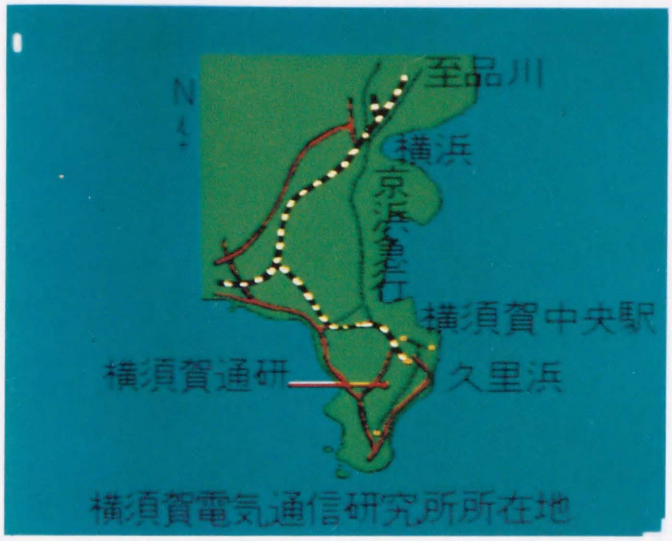


図４．９　彩色図形作成時間の測定結果



（ａ）　Ｅ＝０の場合

（ｂ）　Ｅ＝０．２５の場合

（ｃ）　Ｅ＝１．０の場合

図４．１０　　彩色図形のコマンド符号化例

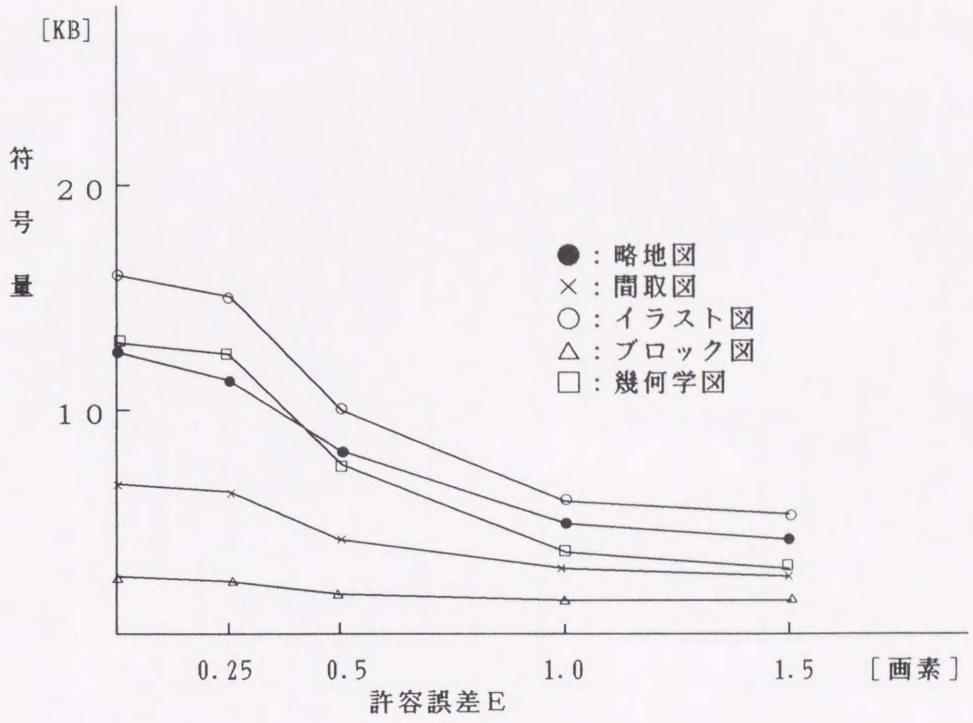


図４．１１　　図形コマンド符号化による画面当たりの符号量

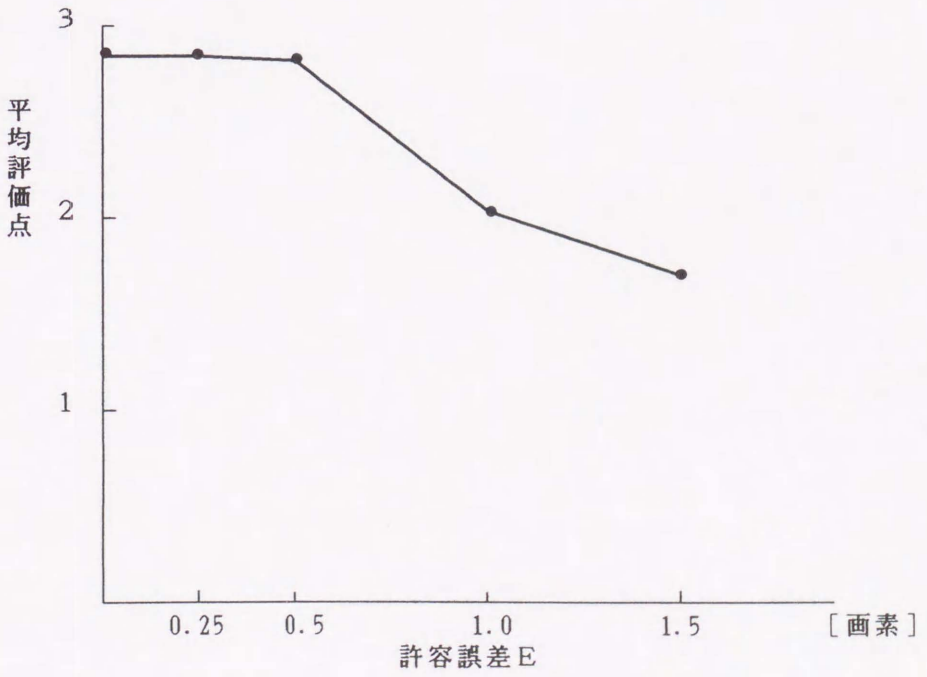


図４．１２　　符号化画面の主観評価結果

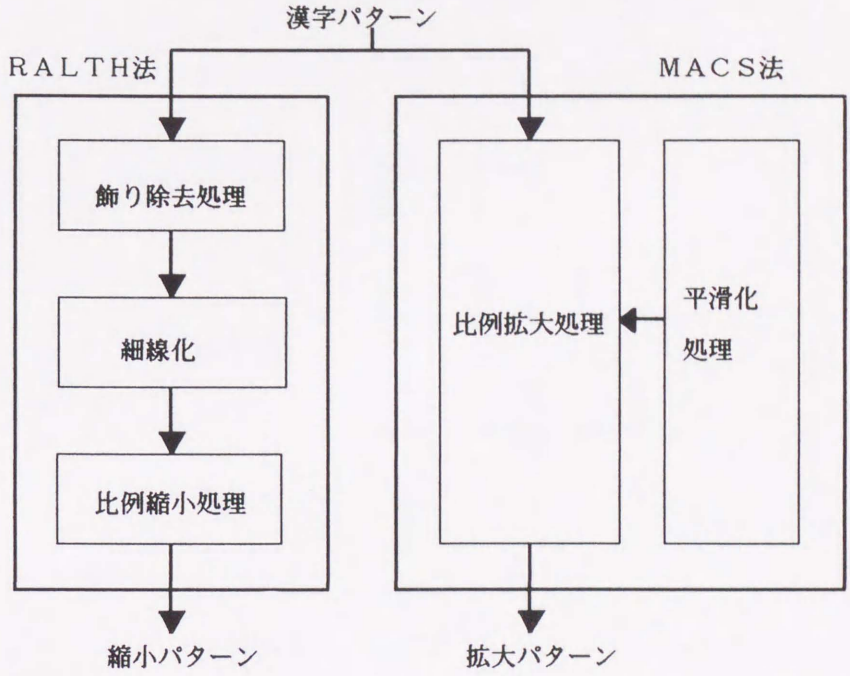


図５．１　　ＲＡＬＴＨ法／ＭＡＣＳ法の処理の流れ

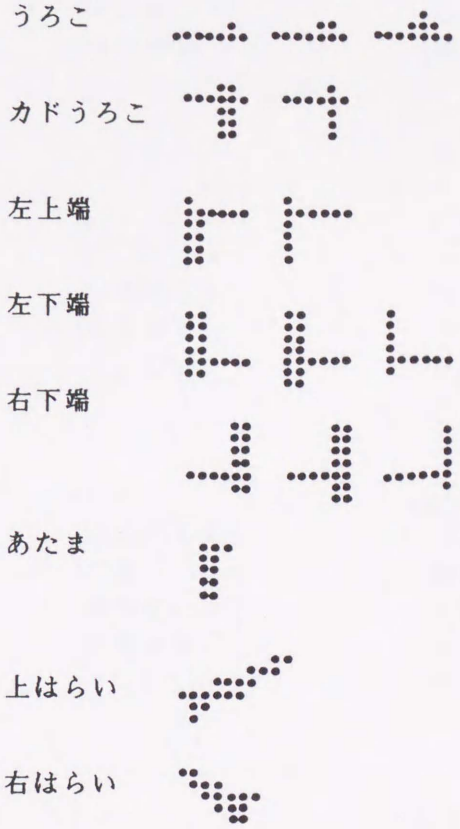


図５．２　明朝体飾りの分類

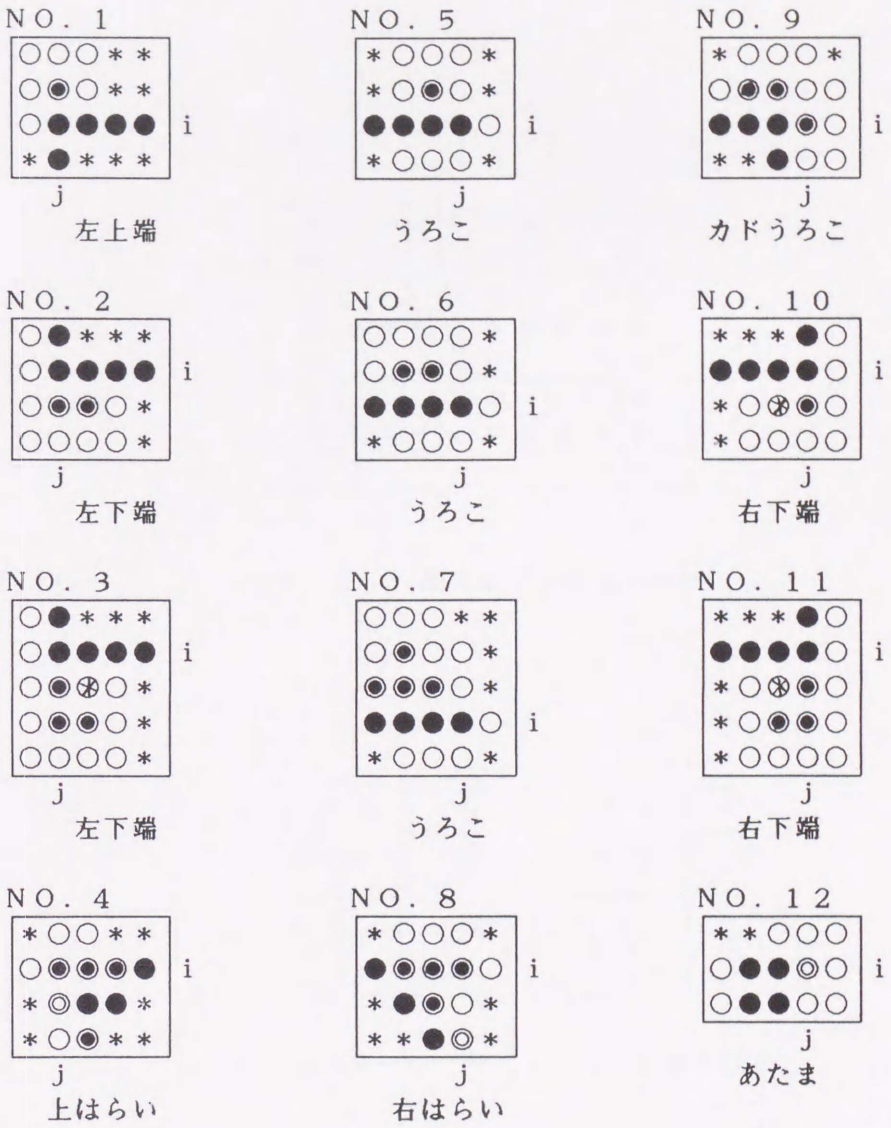


図５．３　飾り除去論理マスク

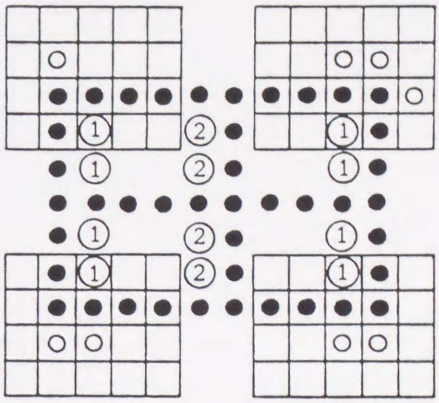
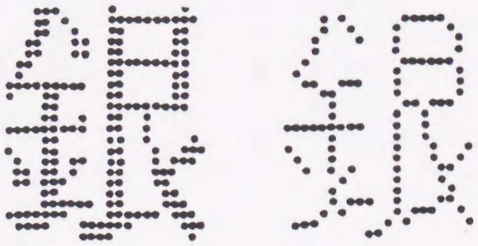
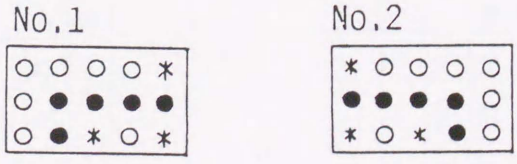


図５．４　　飾り除去と細線化の例



（ａ）原パターン　　（ｂ）細線化結果

図５．５　　細線化歪の例



（ａ）左上端　　（ｂ）右上端

図５．６　　カド部検出マスク

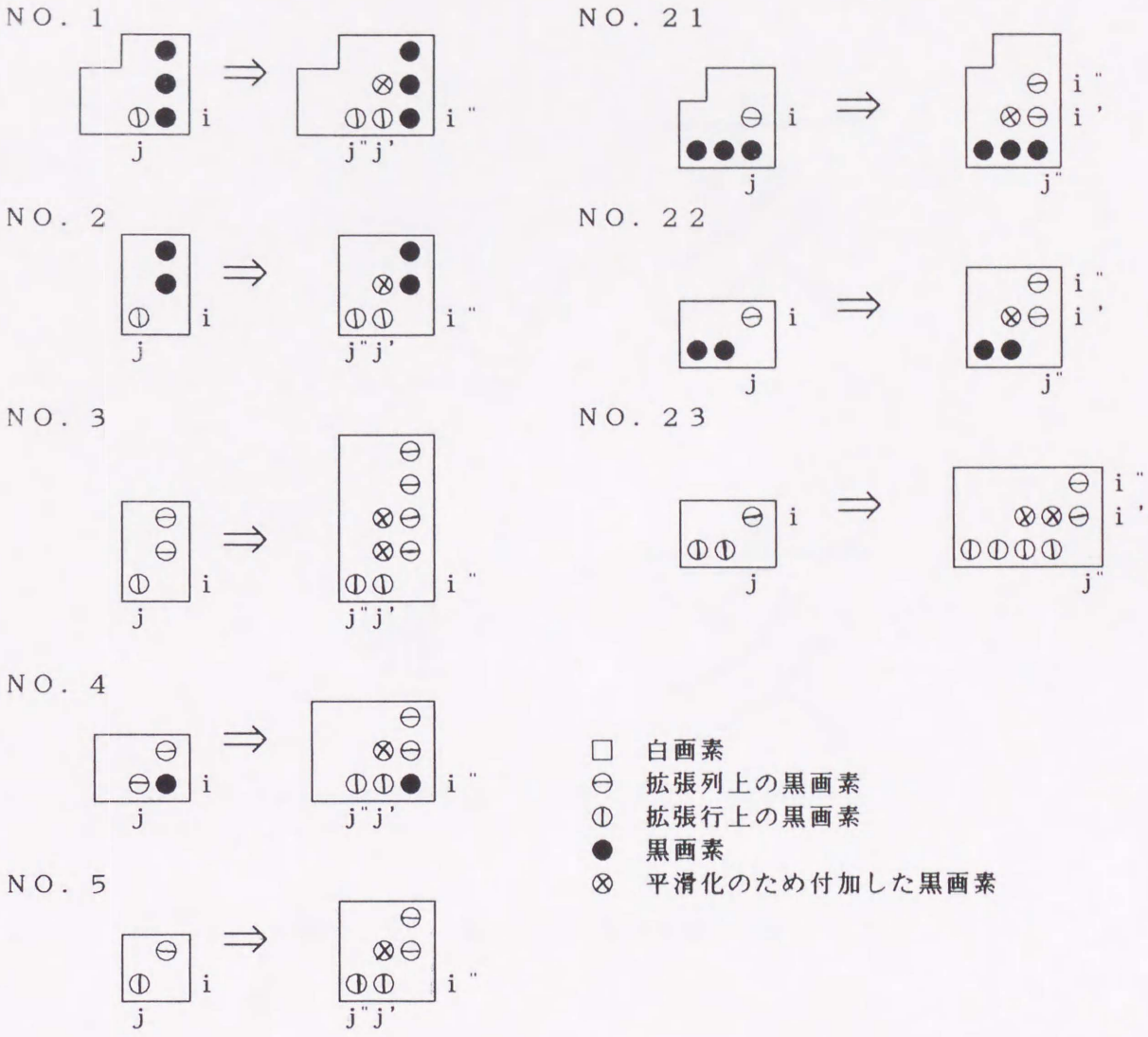


図５．７　　拡大スムージング論理マスク

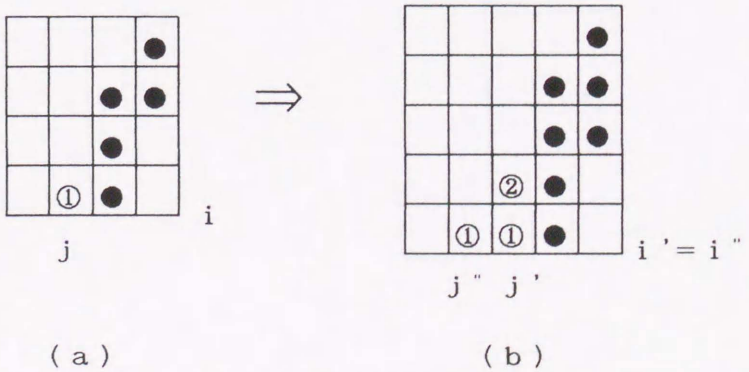


図５．８　　拡大スムージング論理マスクの適用例

（ａ）　ＲＡＬＴＨ法　　（ｂ）　ＭＡＣＳ法

（ｃ）　比例縮小法　　（ｄ）　比例拡大法

図５．９　　縮小変換と拡大変換の比較

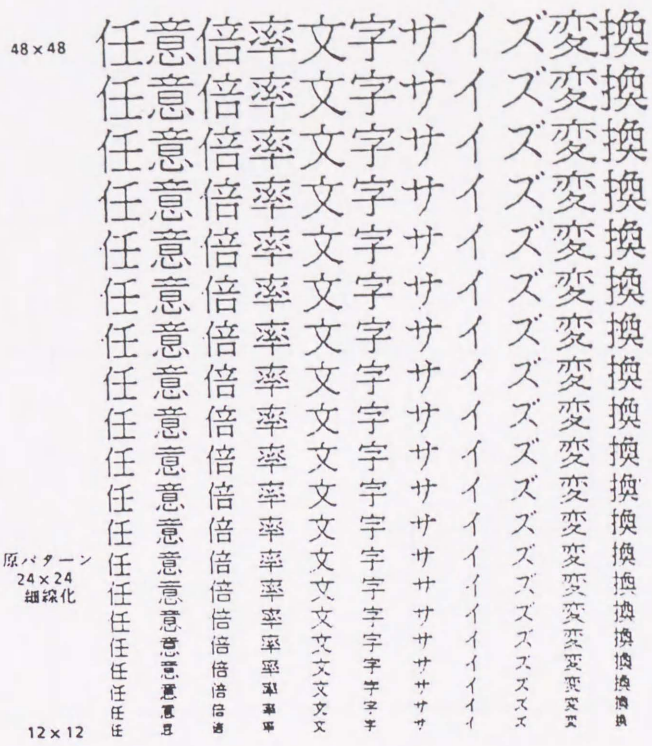


図５．１０　　文字サイズ変換結果

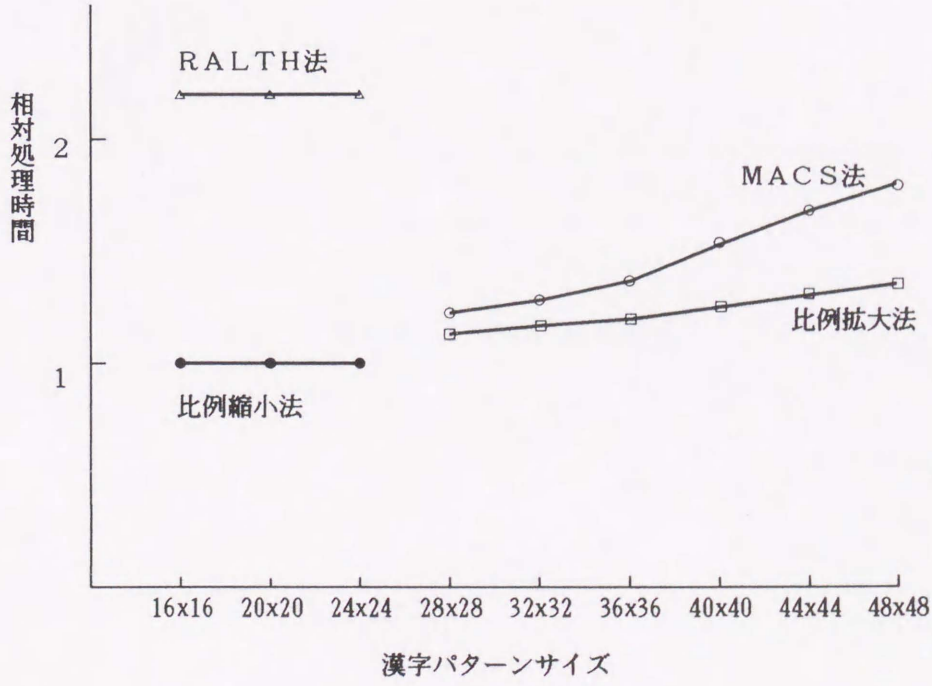


図５．１１　　相対処理時間と漢字パターンサイズの関係

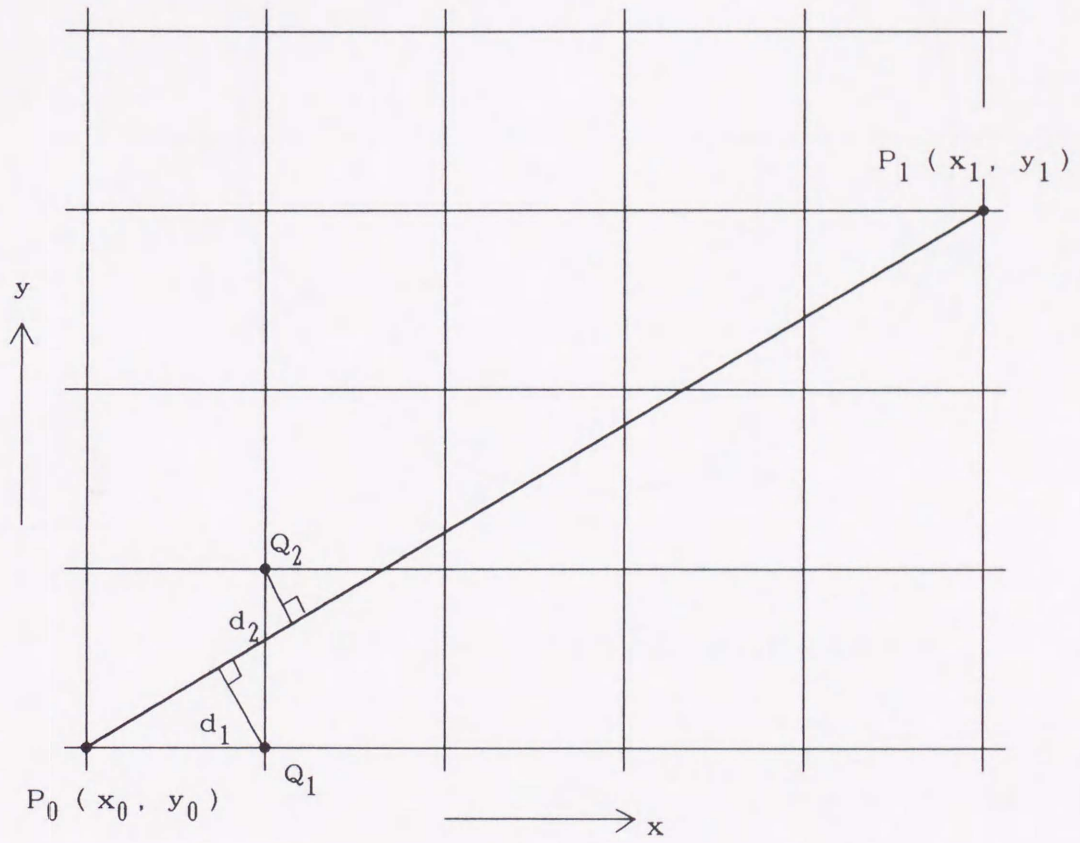


図５．１２　　線分発生のための画素選択

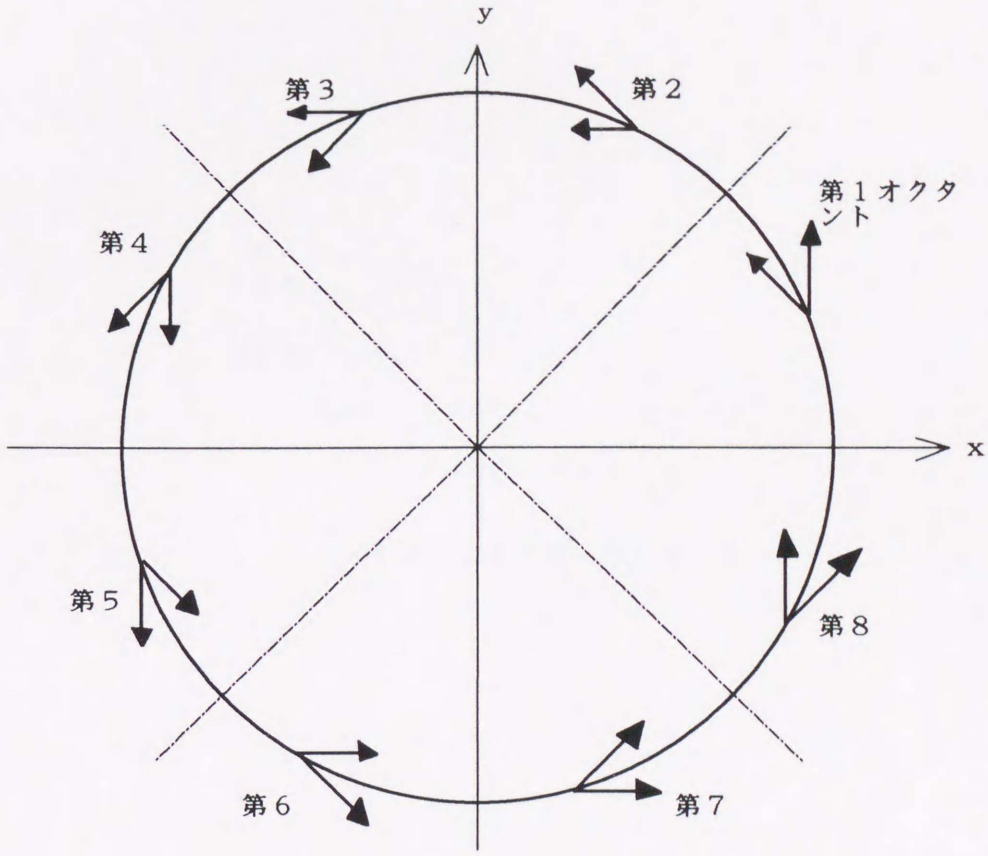


図５．１３　　円弧発生のための画素選択方向

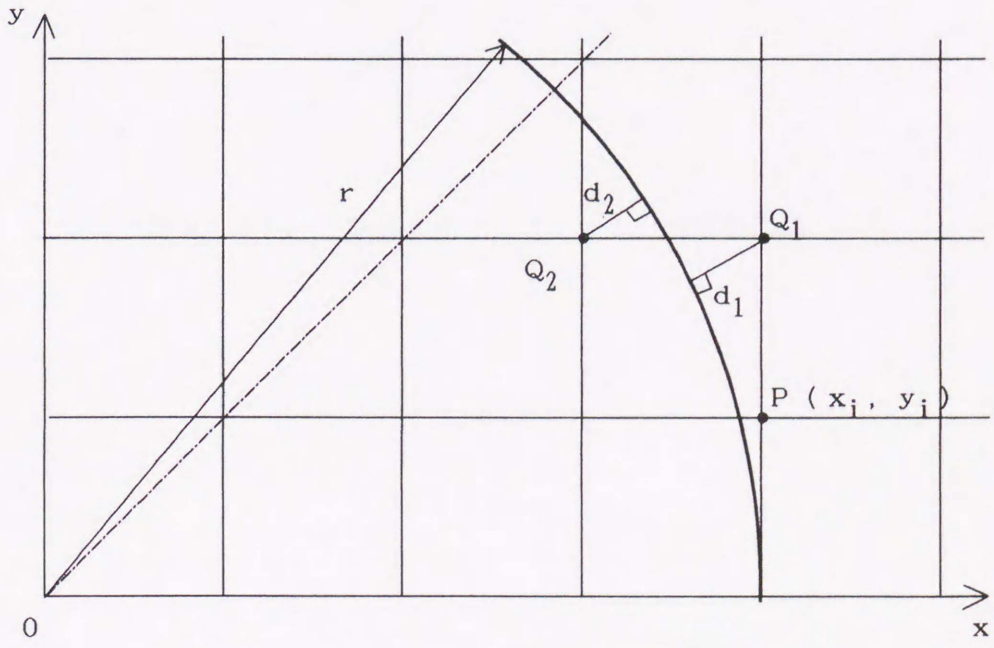


図５．１４　　円弧発生のための画素選択

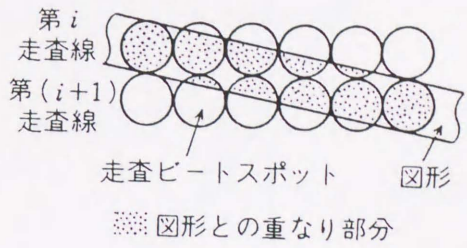


図５．１５　　ＴＶカメラによる細い図形の走査

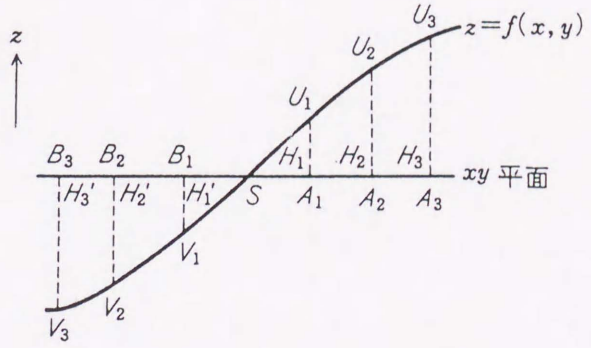


図５．１６　　曲面ｚ＝ｆ（ｘ，ｙ）と変位の関係

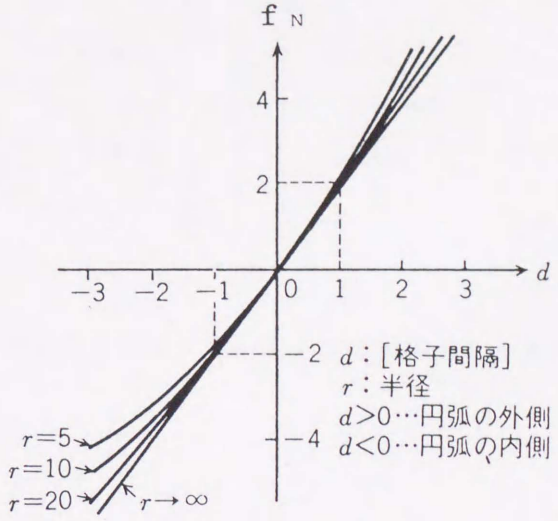


図５．１７　　真円からの距離と正規化変位の関係

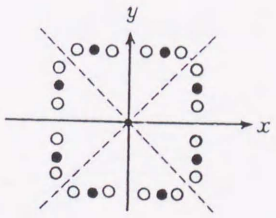


図５．１８　　線分発生で式（５．１１）に適用される画素

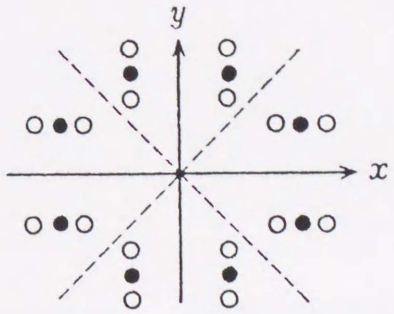


図５．１９　　円弧発生で式（５．１１）に適用される画素

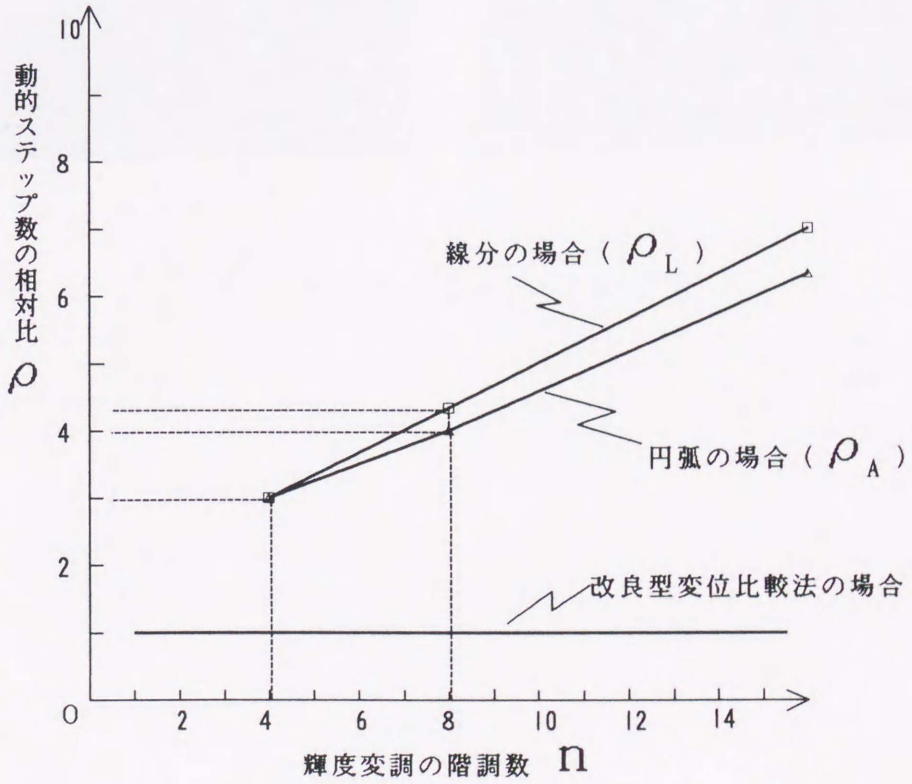


図５．２０　　線分・円弧発生の動的ステップ数の相対比

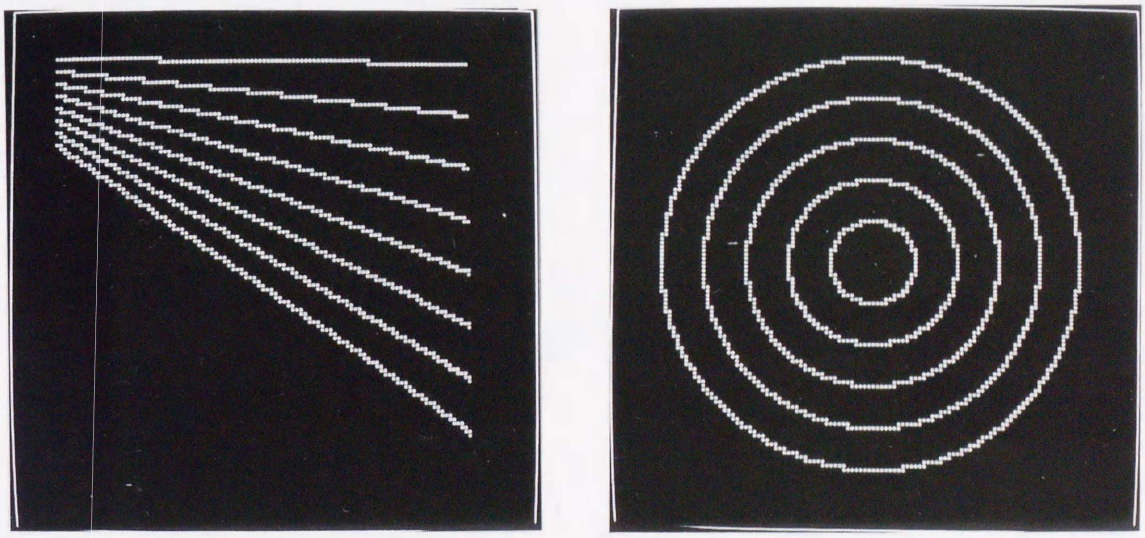


図５．２１　　２値表示による表示例

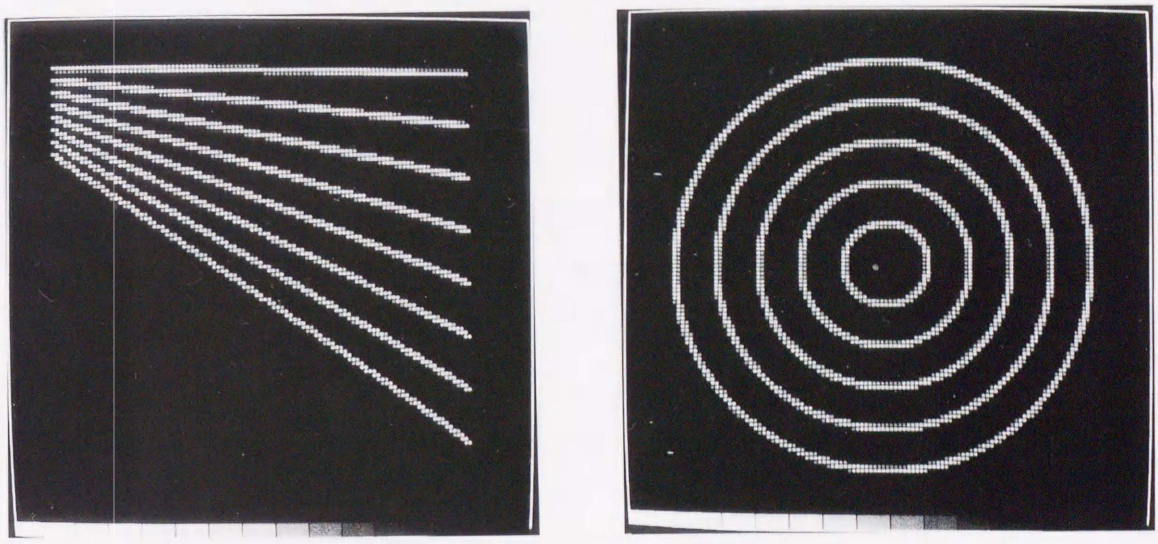


図５．２２　　４値表示による表示例

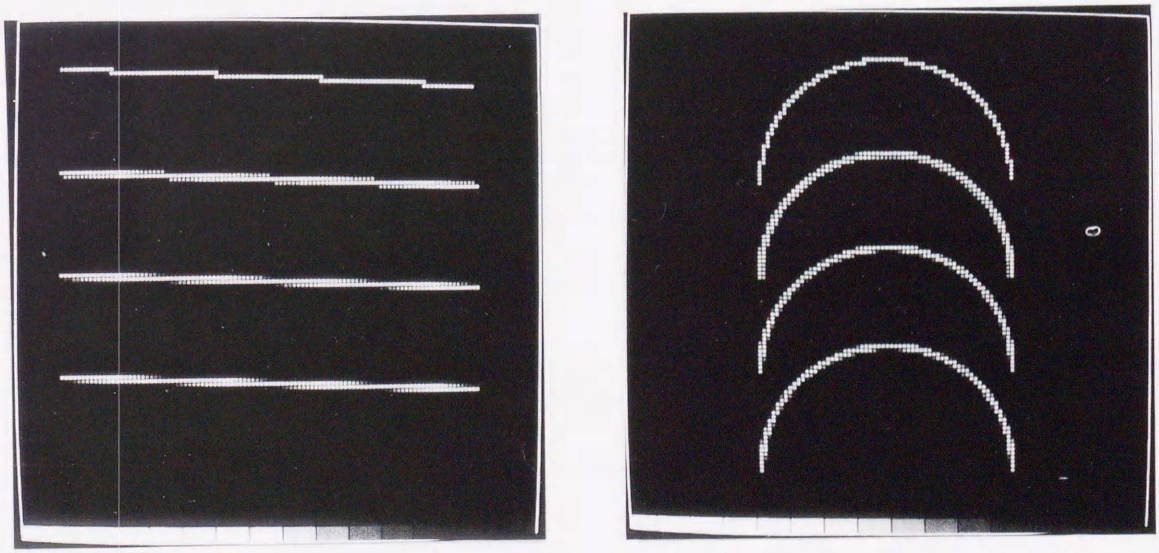


図５．２３　　階調数増加による効果

表４．１　　画像ファイル作成方式に対する要求条件

工程		要求条件	手作業による方法(*)
下絵制作	カット図の使用	可能	○
版下制作	文字の種類	３０００字程度	○
	文字の大きさ	２～３種類	○
	文字単位の修正	可能	×
	図形の形状	任意	○
	図形の一部修正	可能	×
画稿制作	色の種類	３０種程度	○
	色の修正	可能	×
	彩色時間	１０分／枚程度	×
	色むら	無	×
	色のはみだし	無	△
撮影	合成	可能	○
	合成時の色変化	無	△
	平行移動	可能	○
	撮影装置	廉価　取扱い容易	△
その他	遠隔入力	可能	△
	符号量	１０ＫＢ／枚程度	×
	原画の経年変化	無	△
	原画コピー	可能	△

(*)○：手作業による方法でほぼ満足されているもの
　△：改善が望まれるもの
　×：手作業による方法で実現不可のもの

表４．２　画像ファイル作成装置の諸元

項番	名称		諸元
1	中央処理部		・ミニコンピュータ（HITAC−10ⅡL） ・語長：　16ビット ・主記憶容量：　64KB
2	外部メモリ部	磁気ディスク	・固定ヘッド型 ・容量：　3MB
		フレキシブルディスク	・容量：　1MB
3	編集メモリ部	PCM符号メモリ	・容量：　224KB （512画素×512画素×7面）
		ランレングス符号メモリ	・容量：　64KB
4	色見本発生部		・64色（R,G,B各2ビット）
5	線画入力部		・ファクシミリ ・水平解像度：　8本／mm ・垂直解像度：　7.7本／mm
6	線画入力処理部		・入力処理用バッファ：　18.2KB
7	文字入力部		・電磁結合型タブレット ・収容字数：　漢字2775字 非漢字725字
8	文字情報蓄積部		・容量：　21.5KB
9	表示部		・14インチCRT（3台） ・2：1インタレース
10	操作部		・図形／文字指示：　ジョイスティック ・数値入力：　テンキー ・処理機能指示：ファンクションスイッチ

表４．３　彩色処理の概要

処理項目		処理の概要
1	指定領域検出処理	入力線画に対し、操作者が指定した画素を含む連結領域を検出する処理。
2	未彩色領域検出処理	線で囲まれた未彩色の連結領域を自動的に検出する処理。
3	彩色処理	検出した連結領域に対して色符号を与え、領域を塗りつぶす処理。
4	画素彩色処理	入力線画あるいは彩色した線画の細かい部分を、画素単位で彩色する処理。

表４．４　編集処理の概要

処理項目		処理の概要
1	拡大処理	８，１２，１６倍の倍率で図形を拡大する処理。
2	平行移動処理	図形を指定された量だけ並行移動するよう符号上で行う処理。
3	合成処理	２画面分の図形の各画素について演算を行い、一方の編集メモリに演算結果を書き込む処理。

表４．５　文字画面編集処理の概要

処理項目		処理の概要
1	文字パターン変換処理	入力された文字コードを文字パターンに変換する処理。
2	論理演算処理	文字パターンを編集メモリに書き込む際、置換え、合成等の論理演算を施す処理。
3	編集処理	文字の削除、挿入、字づめ、などにより文字画面の編集を行う処理。
4	平行移動処理	文字画面全体を指定された量だけ並行移動するよう位置情報を制御する処理。
5	合成処理	２つの文字画面の文字情報を１つの文字画面となるよう論理演算を施して合成する処理。
6	文字領域処理	文字表示領域を線画として入力し、この領域の中で文字を順次書き込むよう制御する処理。

表4.6　画像ファイル作成装置のシステム規模

機　能		概　要	実現手段 (ﾊｰﾄﾞ/ｿﾌﾄ)	使用言語	ステップ数 [Ｋｓ]	外部バッファ [ＫＢ]
制御機能 （ＯＳ）		システム制御	ソフト	アセンブラ	1.8	－
		入出力ドライバ	ソフト	アセンブラ	1.6	－
		ユーティリティ	ソフト	アセンブラ	2.2	－
図形画面作成処理	線画入力処理	ファクシミリ入力制御	ソフト	アセンブラ	1.2	0.8
		密度変換処理	ハード	－	－	2.5
		線幅識別処理	ハード	－	－	2.3
		細線化処理	ソフト	アセンブラ	2.0	1.0
		水平垂直化処理	ソフト	アセンブラ	3.0	
		太め処理	ソフト	アセンブラ	1.6	－
	彩色処理	領域検出・着色・修正処理	ソフト	アセンブラ	4.1	288 (編集メモリ)
	編集処理	平行移動処理	ﾊｰﾄﾞ/ｿﾌﾄ	アセンブラ	4.1	
		合成処理	ﾊｰﾄﾞ/ｿﾌﾄ	アセンブラ	1.7	
		拡大処理	ハード	－	－	3.1
	符号化処理	図形コマンド符号化	ソフト	フォートラン	1.8	－
文字画面作成処理		文字コードの入力文字列の編集	ソフト	アセンブラ	7.9	21.5
ファイル入出力処理		画面情報の入出力	ソフト	アセンブラ	3.8	－
合　計					36.8	319.2