(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3688938号
(P3688938)
(45)発行日　平成17年8月31日(2005.8.31)　(24)登録日　平成17年6月17日(2005.6.17)

(51)Int.Cl.[7]　F I
H04N 1/387　H04N 1/387 101
H04N 1/407　H04N 1/40 101E

請求項の数 7　(全 21 頁)

(21)出願番号　特願平11-146593
(22)出願日　平成11年5月26日(1999.5.26)
(65)公開番号　特開2000-332995(P2000-332995A)
(43)公開日　平成12年11月30日(2000.11.30)
審査請求日　平成15年3月27日(2003.3.27)
(31)優先権主張番号　特願平11-73321
(32)優先日　平成11年3月18日(1999.3.18)
(33)優先権主張国　日本国(JP)

(73)特許権者　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号
(74)代理人　100073760
弁理士　鈴木　誠
(74)代理人　100097652
弁理士　大浦　一仁
(72)発明者　作山　宏幸
東京都大田区中馬込１丁目３番６号　株式会社　リコー内

審査官　日下　善之

最終頁に続く

(54)【発明の名称】画像データ生成方法、装置および記録媒体

(57)【特許請求の範囲】

【請求項１】

第１の解像度の画像データから、より解像度の高い第２の解像度の画像データを生成する画像データ生成方法であって、前記第１の解像度の画像データの注目画素毎に、閾値の配列を要素とする閾値マトリックスを用いて階調数変換を施し、前記注目画素の位置および該変換された階調数に対応した所定の第１の符号パターンを参照することで前記第２の画像データを生成し、前記注目画素に対応した符号パターン及び所定個数の前記第１の解像度の画像データに対応した符号パターンから構成される所定形状の注目符号パターン群と、特定符号パターン群とを比較し、当該比較結果が一致の場合に、前記注目符号パターン群の所定の第２の符号パターンを、前記所定の第１の符号パターンの何れとも異なる第３の符号パターンに変更することを特徴とする画像データ生成方法。

【請求項２】

第１の解像度の画像データから、より解像度の高い第２の解像度の画像データを生成する画像データ生成装置であって、前記第１の解像度の画像データの注目画素毎に、閾値の配列を要素とする閾値マトリックスを用いて階調数変換を施す階調数変換手段と、前記注目画素の位置および該変換された階調数に対応した所定の第１の符号パターンを参照することで前記第２の画像データを生成する生成手段とを有し、前記注目画素に対応した符号パターン及び所定個数の前記第１の解像度の画像データに対応した符号パターンから構成される所定形状の注目符号パターン群と、特定符号パターン群とを比較し、当該比較結果が一致の場合に、前記注目符号パターン群の所定の第２の符号パターンを、前記所定の第

１の符号パターンの何れとも異なる第３の符号パターンに変更することを特徴とする画像データ生成装置。

【請求項３】

前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較することを特徴とする請求項２記載の画像データ生成装置。

【請求項４】

前記第１の画像データが複数の色成分を有し、該色成分毎に、前記所定形状の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも１つの色成分については、前記特定符号パターン群の数を、他の色成分の特定符号パターン群の数よりも少なくしたことを特徴とする請求項２記載の画像データ生成装置。

【請求項５】

前記第１の画像データが複数の色成分を有し、該色成分毎に、前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも１つの色成分については、前記注目符号パターン群の数を、他の色成分の注目符号パターン群の数よりも少なくしたことを特徴とする請求項２記載の画像データ生成装置。

【請求項６】

前記少なくとも１つの色成分は黄色であることを特徴とする請求項４または５記載の画像データ生成装置。

【請求項７】

請求項１記載の画像データ生成方法をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、画像情報を、出力装置用のデータとして中間調処理および解像度変換する画像データ生成方法、装置および画像データ生成処理プログラムを記録した記録媒体に関し、アプリケーションプログラム、ディスプレイドライバ、プリンタドライバ等のデバイスドライバに適用される技術である。

【０００２】

【従来の技術】

コンピュータの普及により、テキスト、グラフィックス等のベクトルデータや、自然画等のイメージデータを含んだ複雑な文書が容易に作成できるようになった。このような文書を高品位でプリントできるように、プリンタの解像度も向上しているが、その結果、プリンタへ送信すべきデータ量が増大し、処理時間や転送時間の増大を招いている。

【０００３】

上記したような文書を所定の解像度および階調表現能力を有するプリンタを用いて出力する場合の代表的な方法（以下、第１の方法）として、例えば特開平５－９６７９２号公報に記載されているものがある。

【０００４】

この第１の方法では、プリンタドライバ内で、ベクトルデータを所定解像度のビットマップデータとしてラスター展開するとともに、自然画等のイメージデータに対して所定解像度への解像度変換を行った後、両者に対して所定階調数への階調数変換（中間調処理）を施し、階調変換後（中間調処理後）のデータをプリンタに対して送信するものである。従って、プリントスピードを向上させるためには、ラスター展開、解像度変換、および中間調処理の高速化が必要であり、またこれと同時に、プリンタへのデータ転送の高速化も必要である。

【０００５】

【発明が解決しようとする課題】

そこで、ラスター展開および中間調処理を高速化する方法が、「ＰＰＡ　Ｐｒｉｎｔｅ

ｒ Ｓｏｆｔｗａｒｅ Ｄｒｉｖｅｒ Ｄｅｓｉｇｎ」（ＨＰ Ｊｏｕｒｎａｌ，Ｊｕｎｅ １９９７）に記載されている。

【０００６】

この方法（以下、第２の方法）では、ラスター展開および中間調処理を高速化するため、ラスター展開およびイメージデータの解像度変換をプリンタの解像度よりも低い解像度で行い、中間調処理を施した後プリンタに送信し、プリンタのファームウエアを使って全体の解像度変換（イメージデータ単体の解像度変換と区別するため、これを以下“全体の”と呼ぶ）を行って、上記低解像度をプリンタの解像度に変換して出力する（なお、前掲した文献には、テキストをプリンタの１／２の解像度でラスター展開し、ファームウエアを使ってテキストの解像度を倍に変換することが記載されている）。

【０００７】

上記したように、ラスター展開、ビットマップの解像度変換および中間調処理を低解像度で行えば、処理すべきデータ量が削減される結果、高速化が図られる。しかし、中間調処理後のデータに解像度変換を施した場合、解像度変換を行ってから中間調処理を施す場合に比べて、明らかに画質が低下しやすい。

【０００８】

以上で説明した第１の方法、第２の方法を簡単に比較すると、以下のような傾向となり、概略「第１の方法は画質優先で、第２の方法は速度優先」と言える。

【０００９】

ラスター展開　；第１の方法＝遅い
　　第２の方法＝速い
イメージの解像度変換　；第１の方法＝遅い
　　第２の方法＝速い
中間調処理　；第１の方法＝遅い
　　第２の方法＝速い
転送データ量　；第１の方法＝多い
　　第２の方法＝少ない
画質　；第１の方法＝良い
　　第２の方法＝悪い。

【００１０】

本発明は上記した事情を考慮してなされたもので、

本発明の目的は、ラスター展開とイメージの解像度変換が従来の第１、第２の方法よりも速く、中間調処理、転送データ量、画質については従来の第１、第２の方法の中間となる、両者の長所を併せ持ち、中間調処理と全体の解像度変換を同時に行う画像データ生成方法、装置および記録媒体を提供することにある。

【００１１】

すなわち、

ラスター展開　；第１の方法＝遅い
　　本発明＝速い
　　第２の方法＝速い
イメージの解像度変換　；第１の方法＝遅い
　　本発明＝速い
　　第２の方法＝速い
中間調処理　；第１の方法＝遅い
　　本発明＝第１と第２の中間
　　第２の方法＝速い
転送データ量　；第１の方法＝多い
　　本発明＝第１と第２の中間
　　第２の方法＝少ない
画質　；第１の方法＝良い

本発明＝第１と第２の中間
第２の方法＝悪い。

【００１２】

【課題を解決するための手段】

前記目的を達成するために、請求項１記載の発明では、第１の解像度の画像データから、より解像度の高い第２の解像度の画像データを生成する画像データ生成方法であって、前記第１の解像度の画像データの注目画素毎に、閾値の配列を要素とする閾値マトリックスを用いて階調数変換を施し、前記注目画素の位置および該変換された階調数に対応した所定の第１の符号パターンを参照することで前記第２の画像データを生成し、前記注目画素に対応した符号パターン及び所定個数の前記第１の解像度の画像データに対応した符号パターンから構成される所定形状の注目符号パターン群と、特定符号パターン群とを比較し、当該比較結果が一致の場合に、前記注目符号パターン群の所定の第２の符号パターンを、前記所定の第１の符号パターンの何れとも異なる第３の符号パターンに変更することを特徴としている。

【００１３】

請求項２記載の発明では、第１の解像度の画像データから、より解像度の高い第２の解像度の画像データを生成する画像データ生成装置であって、前記第１の解像度の画像データの注目画素毎に、閾値の配列を要素とする閾値マトリックスを用いて階調数変換を施す階調数変換手段と、前記注目画素の位置および該変換された階調数に対応した所定の第１の符号パターンを参照することで前記第２の画像データを生成する生成手段とを有し、前記注目画素に対応した符号パターン及び所定個数の前記第１の解像度の画像データに対応した符号パターンから構成される所定形状の注目符号パターン群と、特定符号パターン群とを比較し、当該比較結果が一致の場合に、前記注目符号パターン群の所定の第２の符号パターンを、前記所定の第１の符号パターンの何れとも異なる第３の符号パターンに変更することを特徴としている。

【００１４】

請求項３記載の発明では、前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較することを特徴としている。

【００１５】

請求項４記載の発明では、前記第１の画像データが複数の色成分を有し、該色成分毎に、前記所定形状の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも１つの色成分については、前記特定符号パターン群の数を、他の色成分の特定符号パターン群の数よりも少なくしたことを特徴としている。

【００１６】

請求項５記載の発明では、前記第１の画像データが複数の色成分を有し、該色成分毎に、前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも１つの色成分については、前記注目符号パターン群の数を、他の色成分の注目符号パターン群の数よりも少なくしたことを特徴としている。

【００１７】

請求項６記載の発明では、前記少なくとも１つの色成分は黄色であることを特徴としている。

【００１８】

請求項７記載の発明では、請求項１記載の画像データ生成方法をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体であることを特徴としている。

【００１９】

【発明の実施の形態】

以下、本発明の一実施例を図面を用いて具体的に説明する。

（実施例１）

図１は、本発明の実施例１の構成を示す。図において、１はラスター展開または解像度変換した後の画像データ、２は閾値が配列された閾値マトリックス、３は符号パターンが格納された符号パターン格納部、４はプリントデータであり、画像データ１と閾値マトリックスは処理装置５内にあり、符号パターン格納部３とプリントデータ４はプリンタ６内にある。また、本発明の実施例１では、例えば解像度６００ｄｐｉ、階調数２値のプリンタを出力装置として用いる。

【００２０】

以下、本発明を説明する前に前述した従来の第１、第２の方法について図を用いて説明する。図２は、階調数変換（中間調処理）で使用する４×４の閾値マトリクスを示す。マトリックスのデータの各画素は、例えば０から１５までの値（各画素が４ビットの多値）をとる。

【００２１】

図３（ａ）は、前記した第１の方法によって、ベクトルデータを６００ｄｐｉでラスター展開した後のデータ、あるいはイメージデータを６００ｄｐｉに解像度変換した後のデータである。また、図４（ａ）は、同様に前記した第２の方法によって、３００ｄｐｉでラスター展開または解像度変換した後のデータを示す。なお、この例では、図３（ａ）は、図４（ａ）を公知の最近傍法で２倍に解像度変換した場合と同じものになっている。

【００２２】

ラスター展開（あるいは解像度変換）自体の処理量は、概ね解像度の２乗に比例するので、第１の方法のラスター展開（あるいは解像度変換）の処理量は、第２の方法の処理量の４倍となる。

【００２３】

第１の方法においては、図３（ａ）のデータについて各画素値と、各画素位置に対応する図３（ｂ）（図２と同一のマトリックス）の閾値マトリクスの閾値とを比較し、公知のディザ法によって２値化を行う。この例の場合、閾値以上の値を有する画素の値を１と出力するので、階調値として図３（ｃ）を得るが、閾値に対して計１６回の比較演算が必要となる。該データに対して、圧縮等を行わない場合には、図３（ｄ）に示すデータがプリントすべきデータ（図３（ｃ）と同一）となるため、プリンタに対しては、１６画素×２ｂｉｔ＝３２ｂｉｔのデータが転送されることになる。

【００２４】

同様に、第２の方法では、図４（ａ）のデータに対して、図２の閾値マトリックスを作用させる場合は、４×４のマトリックスの１／４のマトリックスが順に使用され、この例では図４（ｂ）のマトリックスによって２値化を行って、図４（ｃ）の階調値を得る（図４（ａ）のデータの右に隣接する２×２のデータ（図は省略）に対しては、図２中で図４（ｂ）のマトリックスの右に隣接する２×２のマトリックス（閾値３、１５、１１、７）が使用される）。

【００２５】

この第２の方法の場合、閾値に対して計４回の比較演算しか行われないので、第１の方法に比べて中間調処理に要する時間は少ない。また、データ圧縮等を行わない場合には、プリンタに対して４画素×２ｂｉｔ＝８ｂｉｔのデータが転送されることになり、転送データ量も少ない。

【００２６】

ただし、第２の方法においては、プリンタ側のファームウエアで、図４（ｃ）のデータ全体の解像度を２倍にするため、プリントすべきデータは図４（ｄ）となる。なお、この例では、解像度変換法として公知の最近傍法を用いる。

【００２７】

図５は、本実施例を説明する図であり、図５（ａ）は本実施例の画像データ１であり、前述した第２の方法と同様に、３００ｄｐｉでラスター展開または解像度変換を行った後のデータ（２×２画素）である。また、本実施例の閾値マトリックス２は、図５（ｂ）のように閾値が配列されている。そして、画像データ１を、閾値マトリックス２によって０

から４までの階調数に変換する。

【００２８】

図５（ｂ）は、要素が４個の配列の２×２のマトリックスであり、配列を構成する４つの閾値は、図２のマトリックスを２×２のサブマトリックス４つに分解したときの各サブマトリックス内の閾値と同じであるが、閾値が値順にソートされている。

【００２９】

本実施例における階調数変換は、図５（ａ）の１画素のデータと、その画素位置に対応する４個の閾値とを比較して（つまり、図５（ｂ）のソートされた閾値を順に比較して）、閾値が画素データを越えるまでの（比較回数－１）を階調値とする。すなわち、例えば、画像データ（１５）と、それの対応する閾値マトリックスの閾値（１、５、９、１３）とを比較し、画像データ（７）と、それに対応する閾値マトリックスの閾値（３、７、１１、１５）とを比較する。

【００３０】

この結果、図５（ａ）のデータに対応する階調数として図５（ｃ）が得られ、その比較回数は計１１回となる。

【００３１】

また、本実施例においては、図５（ｃ）に示した階調値をプリンタ６に対して送信するが、そのデータ量は、４画素×３ｂｉｔ（５階調）＝１２ｂｉｔである。

【００３２】

そして、本実施例においては、プリンタ６側には、２×２の符号パターンである５つの配列を要素とし、図５（ａ）の４つの画素の各位置に対応した２×２の符号パターンマトリックス（図６）が符号パターン格納部３に設けられている。

【００３３】

図６の符号パターンマトリックスの各要素は、図２のマトリックスを２×２のサブマトリックス４つに分解し、各サブマトリックスに２×２のデータ（１画素１６階調）を作用させたときに、とり得る２×２のパターンを、階調数順にソートしたものである。符号パターンは、ベクトル量子化におけるコードブックに相当するものである。

【００３４】

そして、図５（ａ）の値１５の注目画素（＊印）に対しては、図６の中の太線内（＊印）の５つの符号パターンが対応し、ファームウェア中で、注目画素の階調値４に対応した二重線で囲まれた符号パターン（１、１、１、１）が選択される。他の画素も同様に、画素の位置と階調値に対応した符号パターンが選択され、最終的にプリントすべきデータ４として、図５（ｄ）が得られる。

【００３５】

本実施例では、第２の方法で図４（ｄ）の最終データを得るための全体の解像度変換に相当する処理を、符号パターンの参照によって行うため、“全体の解像度変換”にかかる処理は、第２の方法よりも高速ではない。しかし、図４（ｄ）、３（ｄ）、５（ｄ）を比較すれば明らかなように、本実施例の方法による最終データ図５（ｄ）は、第２の方法よりも高画質（例えば、原データの階調性がよりよく保存されている）で、第１の方法に匹敵しうるものである（本実施例では、簡単のため、図３と図４を最近傍法で解像度変換した関係にしたが、両者の関係がそうでない場合には、本実施例の最終データは第２の方法よりは劣ることになる）。

【００３６】

以上の例において、３つの方法の処理量の比較結果を簡単にまとめると、以下のようになる（中間調処理の処理量は、閾値との比較回数で近似している）。

【００３７】

ラスター展開　；第１の方法＝４
　　本発明＝１
　　第２の方法＝１
イメージの解像度変換　；第１の方法＝４

本発明＝１
第２の方法＝１
中間調処理　；第１の方法＝１６
本発明＝１１
第２の方法＝４
転送データ量　；第１の方法＝３２
本発明＝１２
第２の方法＝８
全体の解像度変換　；第１の方法＝不要
本発明＝中速
第２の方法＝高速
画質　；第１の方法＝良い
本発明＝中～良
第２の方法＝悪い。

従って、本実施例は、第１の方法と第２の方法の両方の長所を併せ持った方法であるということができる。

【００３８】

図７は、本発明の実施例１の処理フローチャートである。ｘ行×ｙ列の画像データを、ｋ＾２（ｋの２乗）個のソートされた閾値の配列を要素とするｍ行×ｎ列の閾値マトリックスで階調数変換して階調数を得て（ステップ１０２）、ｋ×ｋの符号パターンｋ＾２＋１個の配列を要素とするｍ行×ｎ列の符号パターンマトリックスを参照して、ｋｘ行×ｋｙ列のプリントデータを得る（ステップ１０３）。マトリックスの右端、下端に達したときは（ステップ１０６、１１０）、それぞれ画像データをｎ＋１画素、ｍ＋１画素ずらした位置から同様に処理し、画像データの下端に至るまで（ステップ１０９）同じ処理を繰り返す。

【００３９】

図中、Ｄ（ｉ，ｊ）は、画像データ中の（行，列）＝（ｉ，ｊ）に位置するデータであり、Ｍ（ｐ、ｑ）は、閾値マトリックス中の（行，列）＝（ｐ、ｑ）に位置する閾値配列であり、ｔ（ｉ，ｉ）は、Ｄ（ｉ，ｉ）にＭ（ｐ、ｑ）を作用させたときに得られる階調数であり，Ｐ（ｐ、ｑ，ｔ（ｉ、ｊ））は、Ｄ（ｉ、ｊ）に対して参照すべき、符号パターンマトリックス中の１つの符号パターンである。

【００４０】

また、ソートされた閾値の配列を要素とするｍ行×ｎ列の閾値マトリックスは、ｋｍ行×ｋｎ列の閾値マトリックスから図８の処理フローによって作成でき、ｍ行×ｎ列の符号パターンマトリックスは、ｋｍ行×ｋｎ列の閾値マトリックスから、図９の処理フローによって作成できる。

【００４１】

なお、実施例１においては、原データの縦横比を維持するため、符号パターンの大きさをｋ×ｋとしているが、維持の必要がない場合には、縦横比が１でない符号パターンを作成すればよい。

【００４２】

（実施例２）

実施例１では、ｋ＾２個のソートされた閾値の配列を用いてｋ＾２＋１階調への変換を行った。通常、プリンタへのデータ転送は２進法で行われるため、ここで階調数が２のべき乗になるように構成すると、（転送データ／階調数）が最小となり転送効率がよい。

【００４３】

図１０は、実施例２を構成する閾値マトリックスであり、３個のソートされた閾値の配列を用いて４階調（０から３）への変換を行えば、４画素分の階調データは８ｂｉｔ（＝４画素×２ｂｉｔ（４階調））で済むため、（転送データ／階調数）を実施例１よりも小さくすることができる。

なお、図１０の各サブマトリックスは、図５（ｂ）の各サブマトリックスの閾値配列の中央２つの値を、該２つの値の平均値１つで置き換えたものである。

【００４４】

（実施例３）

上記した実施例は、ディザ法による中間調処理を第２の方法よりも高速に行うものであるが、公知の誤差拡散法にも適用可能である。図１１は、本発明を誤差拡散法に適用した場合の実施例３を説明する図である。図１１（ａ）は、実施例３で使用する周知の誤差マトリックスであり、＊印の注目画素（図５（ａ）の画素値１５）の階調数変換で生じた誤差（１５－１３＝２）を、図１１（ｂ）のように注目画素に隣接する右および下の画素に半分ずつ加算し、図１１（ｃ）に示すデータの注目画素（＊印）に対して、さらに階調数変換を行う。本実施例でも、図６と同様な符号パターンマトリックスが使用できる。図１２は、実施例３の処理フローを示し、図７の処理にさらに誤差を配分するステップ２０４が追加されている。

【００４５】

（実施例４）

実施例１は、図２のマトリックスを基に符号パターンマトリックスを生成したが、図１３に示すような典型的なドット分散型の閾値マトリックスを基に符号パターンマトリックスを生成すると、図１４のように、同一階調値における符号パターンの配列自体もドット分散型となる。すなわち、図１４において、例えば階調値１に対応した４つの符号パターンを、原画像データの画素位置に対応させて配置すると、４×４のマトリックスができるが、これはドット分散型である。なお、図２のマトリックスもドット分散型的な性質を有する。

【００４６】

図１４から明らかなように、符号パターンマトリックスを構成する符号パターンの配列は全て同じとなるから、図１４の例の場合、プリンタ側で５種類の符号パターンのみを保持することにより、符号パターンマトリックスの容量を小さくできると同時に、マトリックスの参照も高速となる。

【００４７】

（実施例５）

また、本発明では、図示しない公知のブルーノイズ型の閾値マトリックス（閾値の配列がランダム）を基に符号パターンマトリックスを生成することも可能である。この場合には、同一階調値における符号パターンの配列自体もブルーノイズ型となる。そして、本実施例では、ディザ法と同様の処理速度で、誤差拡散に匹敵し得る画質を得ることができる。

【００４８】

なお、ブルーノイズ型の閾値マトリックスに関しては、「Ｄｉｇｉｔａｌ Ｈａｌｆｔｏｎｉｎｇ Ｕｓｉｎｇ ａ Ｂｌｕｅ Ｎｏｉｓｅ Ｍａｓｋ」, Ｔ. Ｍｉｔｓａ ａｎｄ Ｋ. Ｐａｒｋｅｒ, １９９１ Ｉｎｔｅｒｎａｔｉｏｎａｌ Ｃｏｎｆｅｒｅｎｃｅ ｏｎ Ａｃｏｕｓｔｉｃｓ, Ｓｐｅｅｃｈ, ａｎｄ Ｓｉｇｎａｌ Ｐｒｏｃｅｓｓｉｎｇ, Ｆｅｂｒｕａｒｙ ａｎｄ Ｍａｙ １９９１（ブルーノイズマスク法の原理など）および「Ｍｏｄｉｆｉｅｄ Ａｐｐｒｏａｃｈ ｔｏ ｔｈｅ Ｃｏｎｓｔｒｕｃｔｉｏｎ ｏｆ ａ Ｂｌｕｅ Ｎｏｉｓｅ Ｍａｓｋ」ｂｙ Ｍ. Ｙａｏ ａｎｄ Ｋ. Ｐａｒｋｅｒ, Ｊｏｕｒｎａｌ ｏｆ Ｅｌｅｃｔｒｏｎｉｃ Ｉｍａｇｉｎｇ, Ｖｏｌ. ３, Ｎｏ. １, ９２‐９７, Ｊａｎｕａｒｙ １９９４（ブルーノイズマスクの作成法等）を参照されたい。

【００４９】

（実施例６）

上記した実施例では、プリンタ６側が符号パターンマトリックス３を保持していたが、処理装置５（ホスト）側から符号パターンマトリックス３をダウンロードするように構成してもよい。また、階調数変換に使用する閾値マトリックスを切り替える場合には、符号

パターンの変更を伴う。これらの場合ホストが、階調データの送信に先立ち、符号パターンマトリックスをプリンタに送信すればよい。

【００５０】

図１５は、実施例６の処理フローチャートである。プリントを開始する前に、ホスト５は符号パターンが送信済みでなく（ステップ３０１）、符号パターンに変更があるとき（ステップ３０２）、符号パターンをプリンタ６に送信し（ステップ３０３）、その後、ホストがプリンタに階調値を送信し（ステップ３０４）、プリンタ６は送信された階調値に対応した符号パターンを参照してプリントデータを出力する（ステップ３０５）。

【００５１】

このように、本実施例では、階調数をプリンタなどの画像出力装置に対して送信するので、転送データ量が削減でき、また、符号パターンをダウンロード可能にしたので、符号パターンの変更に対応することができる。

【００５２】

（実施例７）

ところで、前述した第２の方法は、例えば特開平２－１１２９６６号公報に示されるようなビットパターンのテンプレートマッチングを行うことによって、ジャギーのスムージングを行うことが可能である。しかし、ビットパターン同士でテンプレートマッチングを行う場合には、処理量が多くなり、速度低下を招きやすいという問題がある。

【００５３】

そこで、本実施例では、後述する符号パターンのテンプレートマッチングを行い、簡易な処理を行うことによって、高速にジャギーのスムージングを行う方法を提案する。

【００５４】

以下、本実施例について説明するが、本実施例においても、実施例１と同様に出力装置として解像度６００ｄｐｉ、階調数２値のプリンタを用いる。また、想定する閾値マトリックスは図２と同様であり、図１４と同じ符号パターンマトリックスを使用する。ただし、本実施例では、図１６に示すように、異なる符号パターンにパターン番号（０、１、２、３ａ、３ｂ、４）を付与したものを用いる。また、原画像データとしては、黒色のみのモノクロ画像を想定する。

【００５５】

さて、図１７、１８は、「文」という黒文字のベクトルフォントを３００ｄｐｉでラスター展開し、実施例１の処理手順によって得られたビットパターン（符号パターン）を示す（なお、本来は１枚の図であるが、これを２つに分割し、図１７は「文」の左半分であり、図１８は「文」の右半分である）。グレーの部分には符号パターン中の値１が、白い部分には符号パターン中の値０が対応していて、この例では濃い黒色の文字の処理結果であることから、階調値としては０または４のいずれかとなっている。

【００５６】

図１７の左上（ａ）には、３００ｄｐｉ、６００ｄｐｉの各画素のサイズが示されていて、（ｂ）には、６００ｄｐｉの画素サイズに対応した２×２の符号パターンの並び、およびパターン番号の並びが示されている。また、（ｂ）中には、本実施例における、符号パターンのサイズおよび符号パターン群のサイズが示されている。

【００５７】

この例においては、注目符号パターンを含む４つの符号パターンから成る正方形の注目符号パターン群を、６種類の正方形の特定符号パターン群（図１９）と照合し、一致した場合には、注目符号パターン群をそれぞれ図２０に示す符号パターン群に変更する。図２０において黒色で示した部分は、符号パターン中の変更された位置を示すが、図１６から明らかなように、図２０の変更後の符号パターン（黒色を含む符号パターン）の内の３つは、図１６には存在しない。つまり、図２０（ｅ）の符号パターンは、図１６のパターン番号１として存在するが、図２０（ｃ）、（ｄ）、（ｆ）の符号パターンは、図１６のパターン中にはない。

【００５８】

従って、本実施例では、図２１に示す３つの符号パターン（パターン番号１ｂ、１ｃ、１ｄ）を追加して使用する。また、図１６から明らかなように、符号パターンには重複が生じるので、予め重複を排除したパターン番号を図１６、図２１に示すように定義する。なお、符号パターン数の増加が好ましくない場合には、符号パターンを追加しないという選択も可能である。

【００５９】

ここで、図１９（ａ）の注目符号パターン群を例にとると、図１９と図２０の関係は以下の通りである。図１９（ａ）の特定符号パターン群は、第１象限から左回りに０、４、０、４というパターン番号の順列を形成する。そこで、注目符号パターン群が、第１象限から左回りに０、４、０、４というパターン番号列を形成した場合には、これを図２０（ａ）に示す１ｄ、４、１ｂ、４というパターン番号列（符号パターン群）に変更する。

【００６０】

また、この例においては、１つの注目符号パターン（図１７（ｃ）の＊印）に対する注目符号パターン群（比較のためのウインドウ）のとり方として、図１７（ｃ）に示すＡ，Ｂ，Ｃ，Ｄの４種類（太線Ａ，二重線Ｂ，点線Ｃ，三重線Ｄ）を採用し、注目符号パターン自体のとり方としては、図１７（ｄ）の＊印に示す通り、縦方向・横方向ともに１つ置きとしている（１つ置きで十分だからである）。図１７、１８全体に配置された格子は、図１７（ｄ）の＊印に対する、注目符号パターン群Ａのとりかたと一致している。

【００６１】

本実施例においては、１つの注目符号パターンに対し、注目符号パターン群（比較のためのウィンドウ）Ａをとり、特定符号パターン群（図１９）と一致した場合には注目符号パターン群を図２０に変更して、さらに注目符号パターン群Ｂと比較し、といった処理を繰り返す。

【００６２】

なお、本実施例においては、注目符号パターン群Ａにおける符号パターンの変更が、注目符号パターン群Ｂにおける比較・変更に影響を与えないため、１つの注目符号パターンに対してＡ，Ｂ，Ｃ，Ｄととって比較・変更した後、次の注目符号パターンに移るという処理方法でも、全ての注目符号パターンに対してＡをとって比較し、全ての注目符号パターンに対してＢをとって比較し、という処理方法でも結果は同じである。

【００６３】

そこで、図２２は、全ての注目符号パターンについて、注目符号パターン群（比較のためのウィンドウ）Ａをとった場合の処理結果を示したものであり、黒い部分が変更された符号パターンを示す。これによって、ジャギーのスムージングがなされ、比較のためのウィンドウが１つだけの場合でもある程度の効果があることが分かる。

【００６４】

同様に、図２３は、全ての注目符号パターンについて、注目符号パターン群（比較のためのウィンドウ）Ｂをとった場合の処理結果を示したものであり、図１７、１８全体に配置された格子は、図１７（ｄ）の＊印に対する、注目符号パターン群Ｂのとりかたと一致している。図中の１１から２５部分が新たに変更された符号パターンを示す。

【００６５】

同様に、注目符号パターン群（比較のためのウインドウ）Ｃをとった処理結果が図２４、注目符号パターン群（比較のためのウインドウ）Ｄをとった処理結果が図２５図であり、それぞれ３１から４８の部分、５１から６２の部分が新たに変更された符号パターンを示す。比較のためのウィンドウを複数使用する分、スムージングの効果も大きくなっているのが明らかである。

【００６６】

このように本実施例によれば、簡単かつわずか６つの特定パターンの比較・検出という簡易な方法によって、かなりのスムージング効果が発揮されることがわかる（もちろん、スムージングによって弊害が生じる部分も存在するが、全体の効果からみればわずかなものである）。そして、簡易な方法であるので、高速にスムージングを行うことが可能にな

る。

【００６７】

なお、原画像データの端部においては、注目符号パターン群（比較のためのウィンドウ）を４つ全てとることができない場合があるが、その場合はとり得る注目パターン群のみ比較すればよい。

【００６８】

図２６は、以上で説明した実施例７の処理フローチャートである。まず、実施例１で説明した方法によって、注目画素に対応する階調数を得る（ステップ４０１）。上記した階調数変換をホスト側で行い、階調数をホストからプリンタに送信する（ステップ４０２）。

【００６９】

以下の処理はプリンタによって行われる。すなわち、階調数に対応した符号パターンマトリックス（図１６）を参照して、注目画素のパターン番号を得る（ステップ４０３）。上記した処理を全ての注目画素について行う（ステップ４０４）。

【００７０】

注目符号パターンについて、Ａ～Ｄの全ての注目符号パターン群と全ての特定符号パターン群（図１９）を比較し（ステップ４０５）、一致したとき（ステップ４０６）、注目符号パターン群中の所定符号パターン（パターン番号）を変更する（ステップ４０７）。

【００７１】

例えば、注目画素のパターン番号が０（白）で、注目符号パターンが図１９の（ｆ）に一致したとき、注目画素のパターン番号０を、図２０（ｆ）に変更し、つまりパターン番号１ｂ（図２１）に変更する。また、例えば、注目画素のパターン番号が４（黒）で、注目符号パターンが図１９の（ｄ）に一致したとき、注目符号パターン中のパターン番号０を、パターン番号１ｄ（図２１）に変更する。上記した処理を全ての注目符号パターンについて行う（ステップ４０８）。

【００７２】

注目画素のパターン番号（図１６）と変更されたパターン番号（図２１）から、注目画素に対応する６００ｄｐｉの最終画像データを得る（ステップ４１０）。

【００７３】

（実施例８）

上記した実施例では、黒色のみのモノクロ画像を想定したが、カラー画像の場合には公知のように、以上の処理を各画像成分毎に行えばよい。すなわち、３００ｄｐｉでラスター展開または解像度変換を行った後のデータを、公知の方法で各色成分、例えばＣ（シアン），Ｍ（マゼンタ），Ｙ（黄色），Ｋ（黒）毎のプレーンに分離し、各プレーン毎に図２６の処理を行えばよい。

【００７４】

ここで、例えば上記４プレーンの全てに関して、６つの特定符号パターン群全て（図１９）を用いた比較を行っても良いが、スムージング効果よりも高速性を求める場合、視認性の低い黄色については、特定符号パターン群の数を減らしてもよい。すなわち、特定符号パターン群のサブセットを使用する。例えば、黄色に関しては、図１９（ａ）、（ｂ）に示した２つの特定符号パターン群との比較のみに止める。この場合の処理フローは、黄色のみに関して、図２６における比較ステップ４０５を「注目符号パターンについて全ての注目符号パターン群と、特定符号パターン群のサブセットを比較（パターン番号列同士を比較）」と置き換えればよい。

【００７５】

あるいは、黄色に関しては、上記４つの注目符号パターン群Ａ，Ｂ，Ｃ，Ｄ（図１７（ｃ））を用いた比較を行わずに、Ａのみを用いた比較に止めても良い。前述したように、比較のためのウィンドウが１つだけの場合でもスムージング効果があるからである。この場合の処理フローは、黄色のみに関して、図２６における比較ステップ４０５を「注目符号パターンについて、注目符号パターン群Ａと、全ての特定符号パターン群を比較（パタ

ーン番号列同士を比較）」と置き換えればよい。

【００７６】

上記した例では、視認性の低い黄色について注目符号パターン群の数を少なくしたので、ユーザによって観察されるスムージング効果の低下を抑えながら処理速度を向上させることができる。そして、処理速度を最優先させるならば、黄色については特定符号パターン群との比較を行わないという選択も可能である。

【００７７】

なお、上記した実施例では文字への適用について示したが、本発明はこれに限らず、グラフィックス画像や自然画等にも適用可能である。

【００７８】

（実施例９）

本発明は上記した実施例に限定されず、ソフトウェアによっても実現することができる。本発明をソフトウェアによって実現する場合には、図２７に示すように、ＣＰＵ、メモリ、表示装置、ハードディスク、キーボード、ＣＤ－ＲＯＭドライブ、スキャナなどからなるコンピュータシステムを用意し、ＣＤ－ＲＯＭなどのコンピュータ読み取り可能な記録媒体には、本発明の画像データ生成機能を実現するプログラムなどが記録されている。そして、本発明の画像データ生成処理を実行させるときは、上記した記録媒体に記録されたプログラムを読み出し、これをプリンタドライバとしてＯＳの一部に組み込むことによって本発明の処理機能が実現される。また、処理対象となる原画像データは例えばハードディスクに格納されていて、これに上記した処理が施され、原画像データよりも解像度の高い画像データがプリンタなどに出力される。

【００７９】

【発明の効果】

以上、説明したように、本発明によれば、原画像データの注目画素に対して、ソートされた閾値を用いて階調数変換を行い、符号パターンを参照するので、原画像データよりも解像度の高い画像データを高速に生成することができ、従来の第２の方法よりも高画質な画像を得ることができる。

【００８０】

本発明によれば、階調数変換時に誤差拡散処理を適用するので、高画質な画像データを生成することができる。

【００８１】

本発明によれば、同一階調値における符号パターンのドット配置がドット分散型の配置をとるようにしたので、符号パターンの数を少なくすることができる。

【００８２】

本発明によれば、同一階調値における符号パターンのドット配置がブルーノイズ型の配置をとるようにしたので、高速かつ誤差拡散処理に匹敵する高画質な画像データを生成することができる。

【００８３】

本発明によれば、階調数の送信がｎ進法でなされる場合に、階調数をｎのべき乗としたので、階調数当たりの転送データ量を最小にすることができる。

【００８４】

本発明によれば、テンプレートマッチングによって、注目符号パターン群の所定の第２の符号パターンを、第３の符号パターンに変更するので、原画像データよりも解像度が高くかつスムージングのかかった画像データを高速に生成することができる。

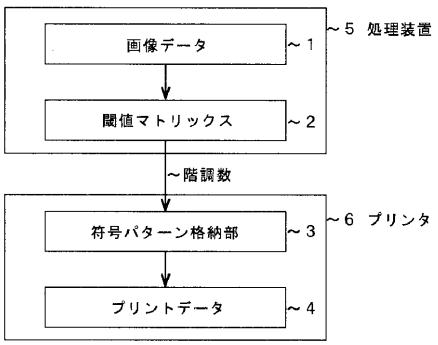
【００８５】

本発明によれば、階調数変換をホストコンピュータ側で行い、階調数を画像出力装置に対して送信し、画像出力装置が符号パターンを参照して原画像データよりも解像度の高い画像データを生成すると共に、第３の符号パターンへの変更を行うので、ホストからプリンタへの転送データ量が削減でき、かつスムージングのかかった最終画像データを高速に生成することができる。

【００８６】
本発明によれば、変更の対象となる第３の符号パターンが、画素に対応した符号パターンのいずれとも異なるので、符号パターンが拡張される結果、高いスムージング効果を得ることができる。
【００８７】
本発明によれば、注目符号パターン群を複数用いるので、高いスムージング効果を得ることができる。
【００８８】
本発明によれば、原画像データの色成分の少なくとも１つについて、特定符号パターン群の数を少なくしたので、スムージング効果への影響を抑えながら処理速度を高めることができる。
【００８９】
本発明によれば、原画像データの色成分の少なくとも１つについて、注目符号パターン群の数を少なくしたので、スムージング効果への影響を抑えながら、処理速度を高めることができる。
【００９０】
本発明によれば、１つの色成分を視認性の低い黄色としたので、ユーザによって観察されるスムージング効果の低下を抑えながら、処理速度を向上させることができる。
【図面の簡単な説明】
【図１】　本発明の実施例１の構成を示す。
【図２】　階調数変換（中間調処理）で使用する４×４の閾値マトリクスを示す。
【図３】　（ａ）～（ｄ）は、従来の第１の方法を説明する図である。
【図４】　（ａ）～（ｄ）は、従来の第２の方法を説明する図である。
【図５】　（ａ）～（ｄ）は、本実施例を説明する図である。
【図６】　符号パターンマトリックスを示す。
【図７】　本発明の実施例１の処理フローチャートである。
【図８】　本発明の実施例１の閾値マトリックス作成の処理フローチャートである。
【図９】　本発明の実施例１の符号パターンマトリックス作成の処理フローチャートである。
【図１０】　本発明の実施例２を構成する閾値マトリックスである。
【図１１】　本発明を誤差拡散法に適用した場合の実施例３を説明する図である。
【図１２】　本発明の実施例３の処理フローチャートである。
【図１３】　ドット分散型の閾値マトリックスを示す。
【図１４】　本発明の実施例４の符号パターンマトリックスを示す。
【図１５】　本発明の実施例６の処理フローチャートである。
【図１６】　本発明の実施例７で使用する符号パターンマトリックスを示す。
【図１７】　「文」という黒文字のベクトルフォントをラスター展開し、実施例１の処理手順によって得られた符号パターンの左半分を示す。
【図１８】　「文」の右半分を示す。
【図１９】　（ａ）～（ｆ）は、特定符号パターン群を示す。
【図２０】　（ａ）～（ｆ）は、変更後の符号パターン群を示す。
【図２１】　追加された符号パターンとパターン番号を示す。
【図２２】　図１７、１８の全ての注目符号パターンについて、比較のためのウィンドウＡをとった場合の処理結果を示す。
【図２３】　図１７、１８の全ての注目符号パターンについて、比較のためのウィンドウＢをとった場合の処理結果を示す。
【図２４】　図１７、１８の全ての注目符号パターンについて、比較のためのウィンドウＣをとった場合の処理結果を示す。
【図２５】　図１７、１８の全ての注目符号パターンについて、比較のためのウィンドウＤをとった場合の処理結果を示す。

【図２６】　本発明の実施例７の処理フローチャートである。
【図２７】　本発明の実施例９の構成を示す。
【符号の説明】
１　画像データ
２　閾値マトリックス
３　符号パターン格納部
４　プリントデータ
５　処理装置
６　プリンタ

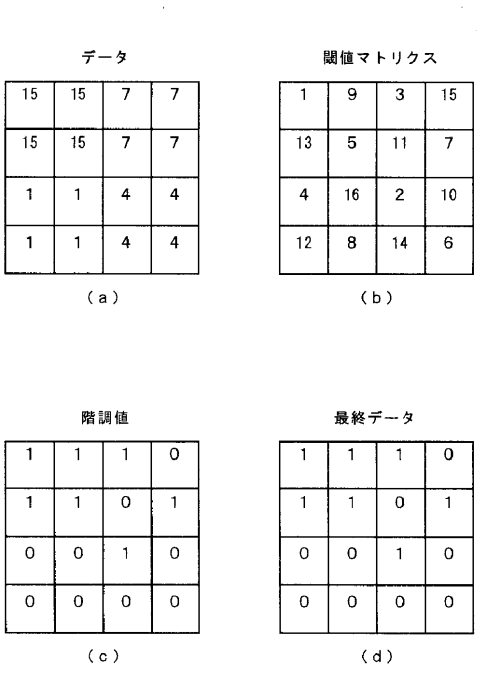
【図１】



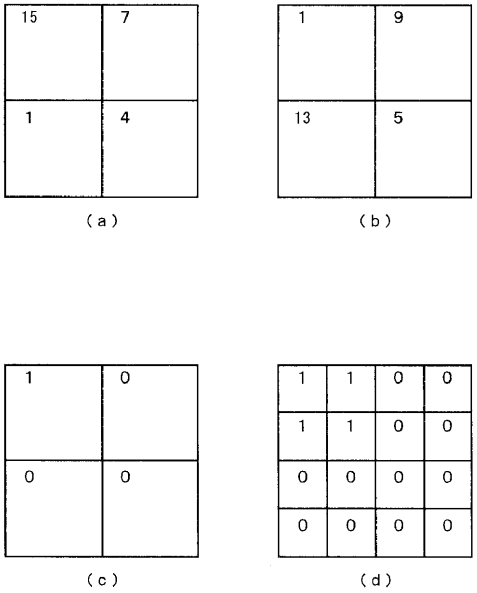
【図２】

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

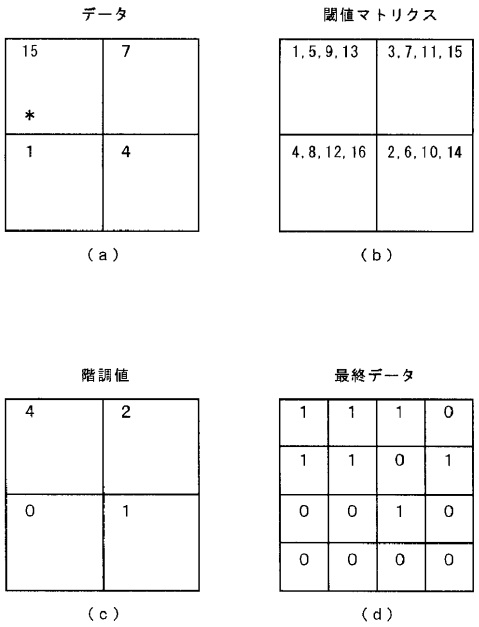
【図３】



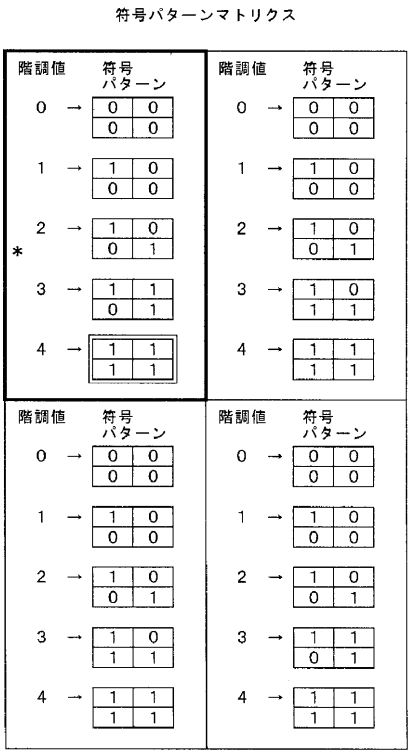
【図４】


15
7
1
4
(a)
1
9
13
5
(b)
1
0
0
0
(c)
1 1 0 0
1 1 0 0
0 0 0 0
0 0 0 0
(d)

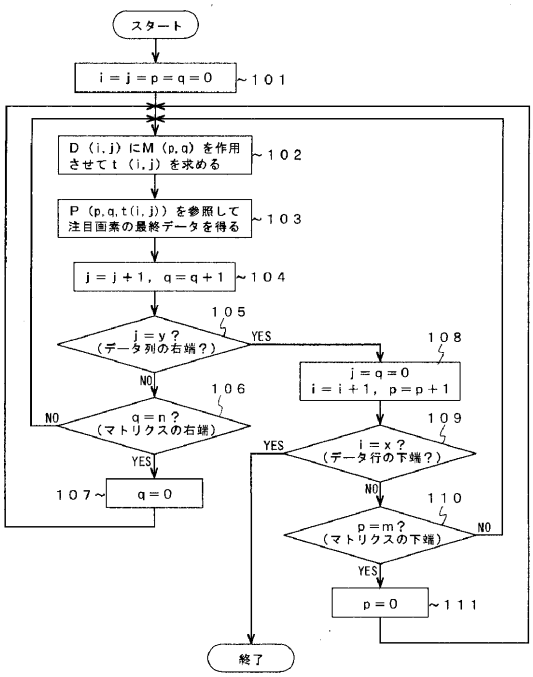
【図５】


データ
15
7
*
1
4
(a)
閾値マトリクス
1, 5, 9, 13
3, 7, 11, 15
4, 8, 12, 16
2, 6, 10, 14
(b)
階調値
4
2
0
1
(c)
最終データ
1 1 1 0
1 1 0 1
0 0 1 0
0 0 0 0
(d)

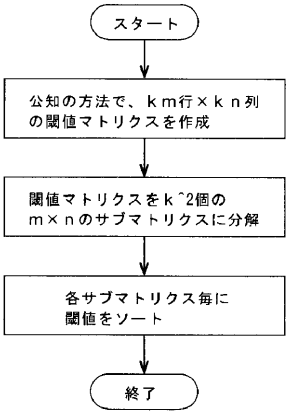
【図６】


符号パターンマトリクス
階調値
符号パターン
0 → 0 0 / 0 0
1 → 1 0 / 0 0
2 → 1 0 / 0 1
*
3 → 1 1 / 0 1
4 → 1 1 / 1 1
階調値
符号パターン
0 → 0 0 / 0 0
1 → 1 0 / 0 0
2 → 1 0 / 0 1
3 → 1 0 / 1 0
4 → 1 1 / 1 1
階調値
符号パターン
0 → 0 0 / 0 0
1 → 1 0 / 0 0
2 → 1 0 / 0 1
3 → 1 0 / 1 1
4 → 1 1 / 1 1
階調値
符号パターン
0 → 0 0 / 0 0
1 → 1 0 / 0 0
2 → 1 0 / 0 1
3 → 1 1 / 0 1
4 → 1 1 / 1 1

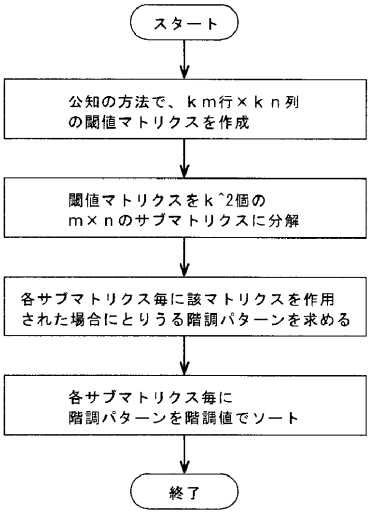
【図７】


スタート
i＝j＝p＝q＝0
～101
D（i, j）にM（p, q）を作用させてt（i, j）を求める
～102
P（p, q, t(i, j)）を参照して注目画素の最終データを得る
～103
j＝j＋1, q＝q＋1
～104
105
j＝y？（データ列の右端？）
YES
NO
108
j＝q＝0
i＝i＋1, p＝p＋1
106
q＝n？（マトリクスの右端）
NO
YES
109
i＝x？（データ行の下端？）
YES
NO
107～
q＝0
110
p＝m？（マトリクスの下端）
NO
YES
p＝0
～111
終了

【図８】



【図９】



【図１０】

1，7，13	3，9，15
4，10，16	2，8，14

【図１１】

*	1/2
1/2	0

（ａ）

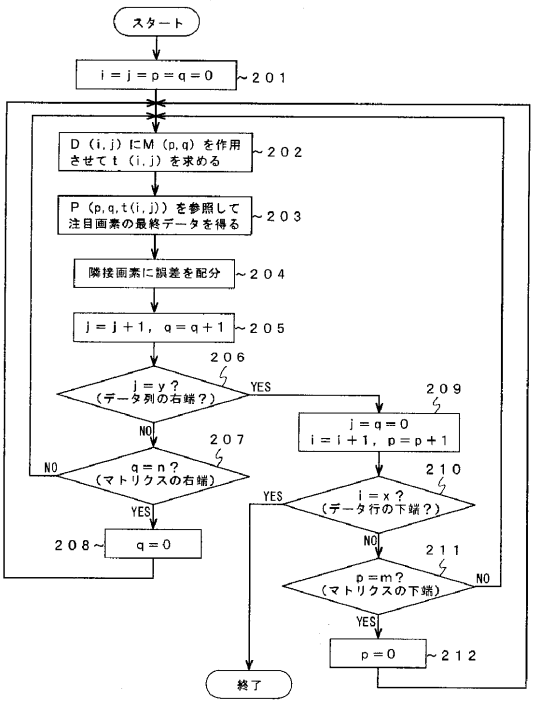
*	＋1
＋1	0

（ｂ）

4	8 *
2	4

（ｃ）

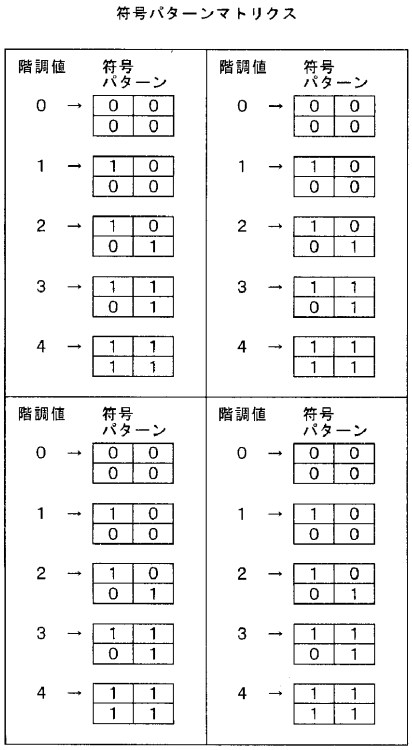
【図１２】



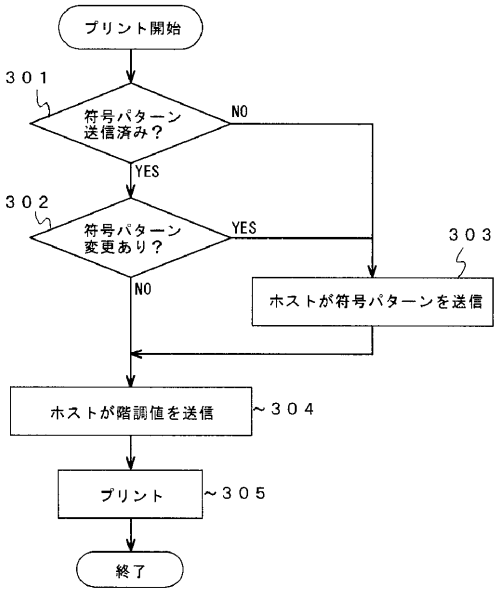
【図１３】

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

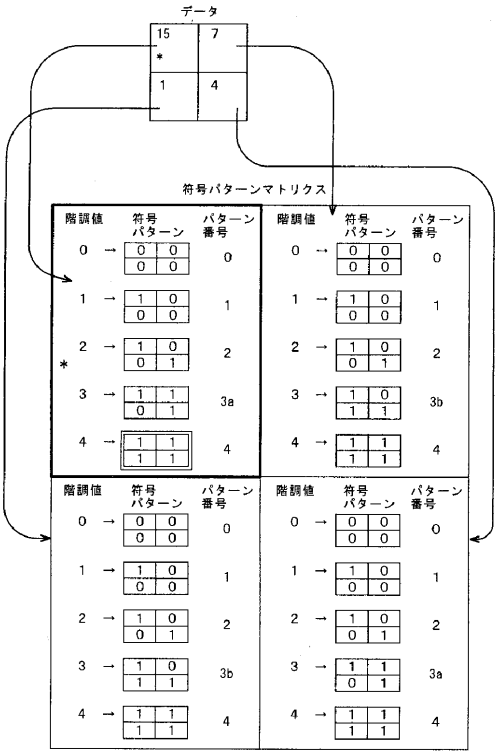
【図１４】



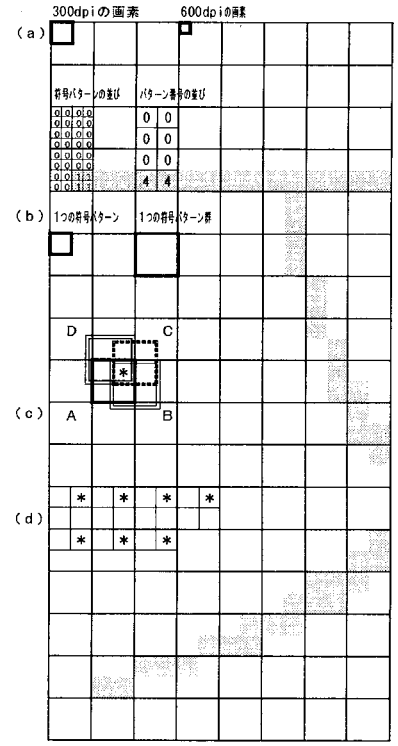
【図１５】



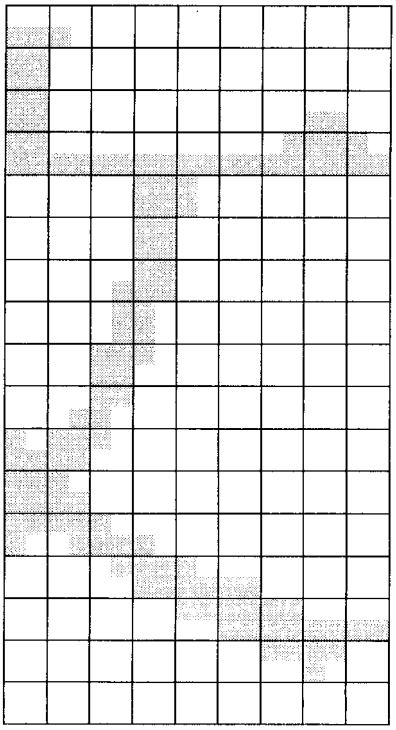
【図１６】



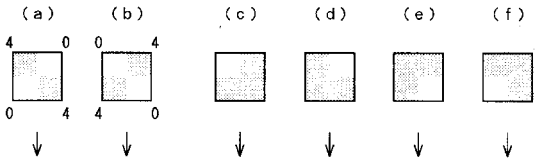
【図１７】



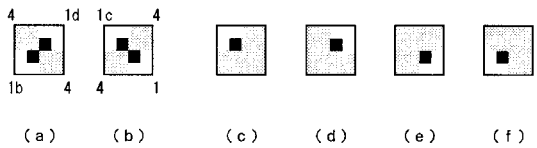
【図１８】



【図１９】



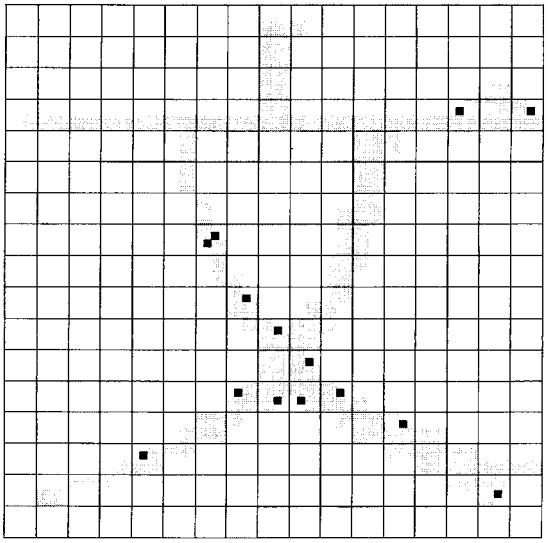
【図２０】



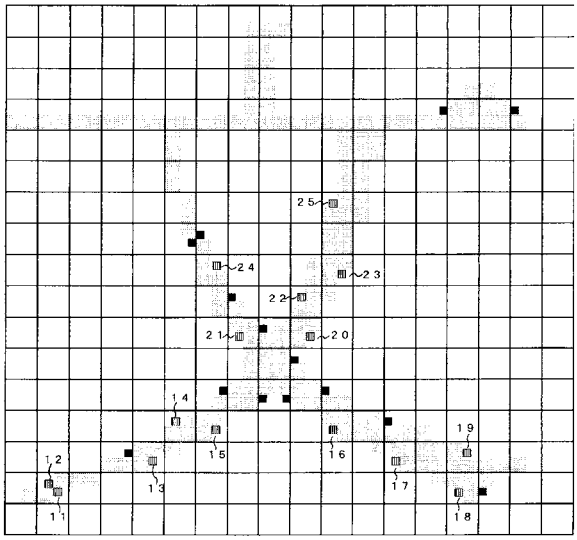
【図２１】

追加符号パターン		パターン番号
0	1	1b
0	0	
0	0	1c
0	1	
0	0	1d
1	0	

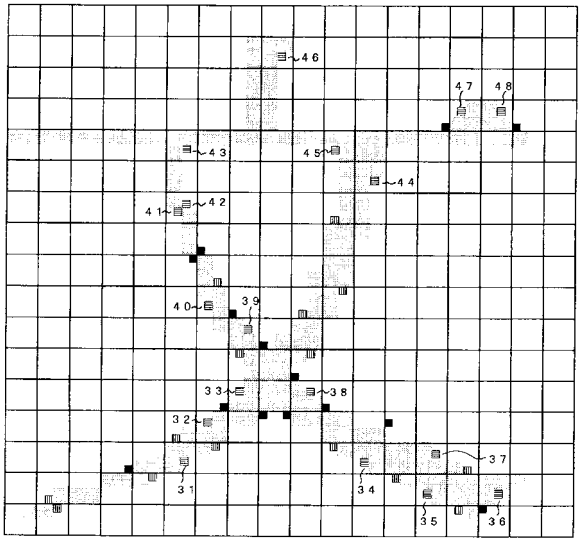
【図２２】



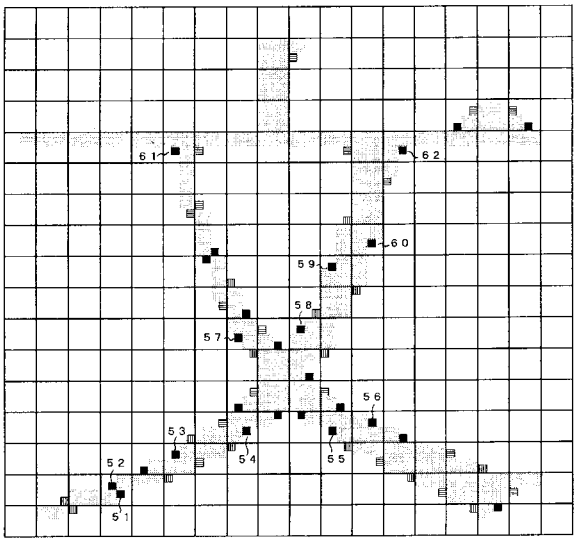
【図２３】



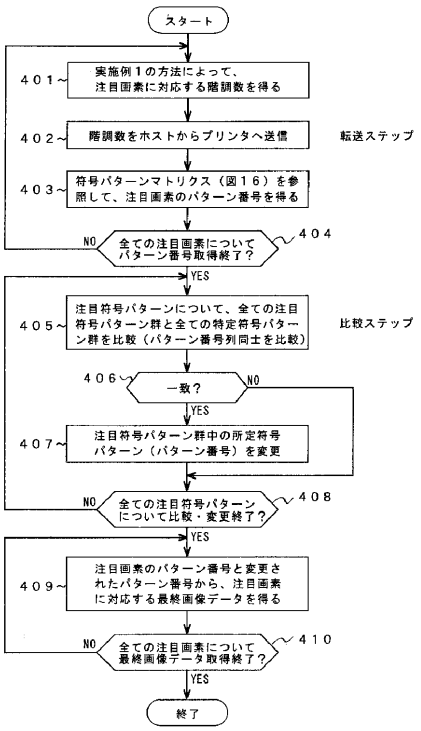
【図２４】



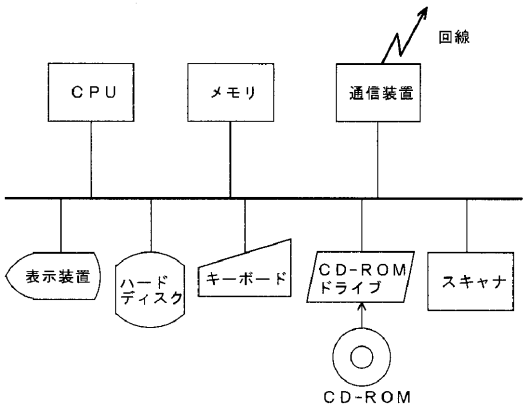
【図２５】



【図２６】



【図２７】



フロントページの続き

(56)参考文献 特開平１０－１９１０３８（ＪＰ，Ａ）
特開平１０－１９１０５４（ＪＰ，Ａ）
特開平０８－０５１５４６（ＪＰ，Ａ）
特開平０９－２７５４９９（ＪＰ，Ａ）
特開平０６－１２５４６２（ＪＰ，Ａ）
特開平０６－２２５１４０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
H04N 1/387 101
H04N 1/407