

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4217706号  
(P4217706)

(45) 発行日 平成21年2月4日(2009.2.4)

(24) 登録日 平成20年11月14日(2008.11.14)

(51) Int.Cl.	F 1
HO4N 1/409	(2006.01)
GO6T 5/00	(2006.01)
GO6T 3/00	(2006.01)
GO6T 5/20	(2006.01)
HO4N 1/40	(2006.01)
	HO4N 1/40
	GO6T 5/00
	GO6T 3/00
	GO6T 5/20
	HO4N 1/40

請求項の数 10 (全 26 頁) 最終頁に続く

(21) 出願番号	特願2005-296962 (P2005-296962)	(73) 特許権者	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22) 出願日	平成17年10月11日 (2005.10.11)	(74) 代理人	100076428 弁理士 大塚 康徳
(65) 公開番号	特開2006-295877 (P2006-295877A)	(74) 代理人	100112508 弁理士 高柳 司郎
(43) 公開日	平成18年10月26日 (2006.10.26)	(74) 代理人	100115071 弁理士 大塚 康弘
審査請求日	平成17年12月7日 (2005.12.7)	(74) 代理人	100116894 弁理士 木村 秀二
(31) 優先権主張番号	特願2004-318239 (P2004-318239)	(72) 発明者	無木 浩 東京都大田区下丸子3丁目30番2号 キ ヤノン株式会社内
(32) 優先日	平成16年11月1日 (2004.11.1)		
(33) 優先権主張国	日本国 (JP)		
(31) 優先権主張番号	特願2005-80592 (P2005-80592)		
(32) 優先日	平成17年3月18日 (2005.3.18)		
(33) 優先権主張国	日本国 (JP)		

最終頁に続く

(54) 【発明の名称】 画像処理装置及び画像処理方法

## (57) 【特許請求の範囲】

## 【請求項 1】

第1の画像データ、前記第1の画像データに対して擬似中間調処理が施された第2の画像データ、及び前記第1の画像データに含まれる各画素の属性を表す属性データに基づいて、前記第2の画像データにおけるジャギーを改善する画像処理装置であって、

前記属性データに基づいて、スムージング処理を行うか否かを示す判別信号を出力する判定手段と、

前記判別信号に応じたスムージング処理を行って、前記第1の画像データからエッジ補正データを生成するエッジ補正生成手段と、

前記第2の画像データの画素信号と、前記エッジ補正生成手段から出力された前記エッジ補正データの画素信号とを比較して、より濃度の高い画素信号を出力する選択手段と、を備えることを特徴とする画像処理装置。

## 【請求項 2】

前記エッジ補正生成手段は、

あらかじめ定められたサイズの領域内の画素に対するスムージング処理を行うスムージング手段と、

前記第1の画像データを任意の閾値で量子化して、注目画素を含むあらかじめ定められたサイズの領域内のパターンとのマッチングを行い、一致した場合に注目画素の前記スムージング手段からのデータを出力するマッチング処理手段と、

前記判別信号と前記第1の画像データとに基づいて、エッジ検出信号を出力するエッジ検出手段と、

10

20

ジ検出手段と、

前記エッジ検出信号に応じて、前記第2の画像データに含まれる画素信号及び前記マッチング処理手段の出力信号の一方を出力する適応スマージング手段と、を含むことを特徴とする請求項1に記載の画像処理装置。

【請求項3】

前記適応スマージング処理手段は、前記エッジ検出信号が出力されていないときには前記第2の画像データに含まれる画素信号を出力し、それ以外のときには前記マッチング処理手段の出力信号を出力することを特徴とする請求項2に記載の画像処理装置。

【請求項4】

前記適応スマージング処理手段は、前記マッチング処理手段の出力信号をルックアップテーブルを参照してあらかじめ定められたビット数に変換する変換手段を含むことを特徴とする請求項2に記載の画像処理装置。 10

【請求項5】

前記エッジ検出手段は、注目画素を含むあらかじめ定められたサイズの領域内の画素値の総和とあらかじめ定められた閾値とを比較し、前記画素値の総和が前記閾値未満であるときに、エッジ検出信号を出力することを特徴とする請求項2に記載の画像処理装置。

【請求項6】

前記選択手段は、前記第2の画像データの画素信号と前記エッジ補正生成手段から出力された前記エッジ補正データの画素信号とのビット数が異なる場合、ビット数の少ない画素信号の下位ビットに「1」を詰めて同じビット数にして比較することを特徴とする請求項1に記載の画像処理装置。 20

【請求項7】

前記スマージング手段及び前記マッチング処理手段におけるあらかじめ定められたサイズの領域が、 $3 \times 3$ 以上の領域であることを特徴とする請求項2に記載の画像処理装置。

【請求項8】

前記あらかじめ定められた閾値が変更することができる、前記あらかじめ定められた閾値の変更によってジャギーの改善の程度を調整することができる特徴とする請求項5に記載の画像処理装置。

【請求項9】

前記第1の画像データの解像度があらかじめ定められた解像度と異なるときに、  
前記エッジ補正生成手段での処理が前記あらかじめ定められた解像度のデータに対する処理と同様となるように、前記第1の画像データ及び前記第2の画像データに対して変換処理を行うデータ変換手段をさらに備えることを特徴とする請求項1に記載の画像処理装置。 30

【請求項10】

第1の画像データ、前記第1の画像データに対して擬似中間調処理が施された第2の画像データ、及び前記第1の画像データに含まれる各画素の属性を表す属性データに基づいて、前記第2の画像データにおけるジャギーを改善する画像処理装置における画像処理方法であって、

前記属性データに基づいて、スマージング処理を行うか否かを示す判別信号を出力する判定工程と、 40

前記判別信号に応じたスマージング処理を行って、前記第1の画像データからエッジ補正データを生成するエッジ補正生成工程と、

前記第2の画像データの画素信号と、前記エッジ補正生成工程で出力された前記エッジ補正データの画素信号とを比較して、より濃度の高い画素信号を出力する選択工程と、を備えることを特徴とする画像処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像処理装置及び画像処理方法に関し、より詳細には、擬似中間調処理を施

50

した画像データにおける、エッジ部のジャギーをコストをかけずに改善する画像処理装置及び画像処理方法に関する。

【0002】

具体的には、電子写真プロセスの複写機やプリンタなどを対象としているが、インクジェットプリンタやディスプレイのような他のプロセスを用いた機器にも応用可能である。

【背景技術】

【0003】

従来から画像処理装置において、低解像度のプリンタで発生するジャギーを改善する技術が、いくつか提案されている。

【0004】

ここでいうジャギーとは、文字のエッジ部などで発生するガタツキのことである。その様子を図14に示した。

【0005】

図14は、従来の画像処理装置における文字処理状態を説明する模式図であり、例えば文字のエッジ部などで発生するガタツキが生じる状態を示している。

【0006】

図14において、示されている各マス目1つが1画素に相当し、この例では300dpiの密度であり、「L」字で示した部分が上述のジャギーを示している。ここでは15個の「L」字が示されている。

【0007】

このようなジャギーに対して、従来の技術ではパターンマッチングを行い、パターンに一致した箇所に画像データを付加またはジャギーの原因となる画素を除去し、ガタツキを改善していた。

【0008】

ここでいうパターンマッチングの詳細については、公知の技術であるため、詳細な説明は省略する。

【0009】

図15は、従来の画像処理装置における文字処理状態を説明する模式図であり、例えば文字のエッジ部などで発生するガタツキ、すなわち、ジャギーを改善した例を示している。

【0010】

図15の(a)は、パターンマッチングで一致した画素にハーフトーンのデータを付加した様子を示している。先ほどの「L」字で示された箇所にハーフトーンデータが付加されている。特に電子写真プロセスを有するようなプリンタでは、このようなハーフトーンデータをジャギー部に付加するだけで、画質を大きく改善できる効果がある。

【0011】

一方、図15の(b)は、パターンマッチングで一致した画素に画素分割したデータを付加した様子を示している。

【0012】

ここで、画素分割とは、300dpiの1画素を複数に分割し部分的にドットを打つ技術である。ここでは、縦に2分割する場合を例にとって示している。画素分割の技術についても公知である為、ここでの詳細な説明は省略する。

【0013】

このように、前述したスムージング処理の例としては、ジャギー部にハーフトーンのドットを付加したり、画素分割ドットを付加するものがある。しかし、逆に、フルドットをハーフトーンドットにしたり、フルドットを画素分割ドットにしたりするようなジャギー部の濃い画素を薄くするような処理もあり得ることはいうまでもない。

【0014】

従来、この種の画像処理装置における問題を改善する技術が、特許文献1に開示されている。

10

20

30

40

50

## 【0015】

一方、図16に示すような技術も知られている。この技術は、スクリーン処理のような擬似中間調処理で発生したジャギーを改善するためのものである。これは、プリンタの解像度より低い解像度で発生するジャギーであり、前述した手法では改善することができなかつたものを改善したものである。

## 【0016】

具体的には、同図(a)に示したジャギーの部分がスクリーンのような画像形成手法によって発生するものであり、プリンタ解像度(ここでは300dpiとする)より粗い解像度で画質劣化が起きていた。それを同図(b)に示すように、エッジの部分にハーフトーンドット(中間調データ)を打つ(出力する)ことで、ジャギーを改善するものである。

10

【特許文献1】特開平10-42141号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0017】

ところで、プリンタの解像度が上がってくると、はじめに述べたようなプリンタの解像度に起因するジャギーはほとんど問題でなくなってくる。すなわち、プリンタの解像度が600dpiや1200dpi程度になると、識別できるレベルのジャギーは少なくなってくる。

## 【0018】

20

しかしながら、スクリーン手法のような画像形成手法に起因して発生するジャギーは、プリンタ解像度が上がっても認識されやすい。スクリーン処理の場合、ファットニング型などと一般的に称されるようなパターンを用いて複数の画素を集中させたドットを形成することで擬似的にスクリーン解像度(線数)を表現することになる。そのため、プリンタの解像度より高いスクリーン解像度(線数)を用いることはない。例えば、600dpiの解像度を有するプリンタでスクリーン処理した画像データを出力する場合、そのスクリーン解像度(線数)は一般的に133線から175線程度であり、高くとも268線程度である。これ以上高い線数のスクリーンを用いると、電子写真プリンタの特性によって、安定した画質が得られないためでもある。なお、線数の単位は一般的にLPI(Line Per Inch)として定義されている。

30

## 【0019】

そこで、後者の画像形成手法によって発生するジャギーを改善する為の技術としては、上述のエッジの部分にハーフトーンドットを打つ方法がある。しかし、この方法は、演算が複雑となり、かつ、多くのメモリを必要とする為、この方法を実現する技術を搭載した装置のコストが高くなる傾向にある。この為、低価格な電子写真プリンタやMFPなどにも搭載可能とするためには、より簡易な構成でかつ上述したジャギーを改善できる方法が必要となる。

## 【0020】

本発明は以上のような状況に鑑みてなされたものであり、画像形成手法によって発生するジャギーを簡易な構成で改善することを目的とするものである。

40

## 【課題を解決するための手段】

## 【0021】

上記目的を達成する本発明の一態様としての画像処理装置は、第1の画像データ、前記第1の画像データに対して擬似中間調処理が施された第2の画像データ、及び前記第1の画像データに含まれる各画素の属性を表す属性データに基づいて、前記第2の画像データにおけるジャギーを改善する画像処理装置であって、

前記属性データに基づいて、スムージング処理を行うか否かを示す判別信号を出力する判定手段と、

前記判別信号に応じたスムージング処理を行って、前記第1の画像データからエッジ補正データを生成するエッジ補正生成手段と、

50

前記第2の画像データの画素信号と、前記エッジ補正生成手段から出力された前記エッジ補正データの画素信号とを比較して、より濃度の高い画素信号を出力する選択手段と、を備えている。

【0022】

すなわち、本発明では、第1の画像データ、第1の画像データに対して擬似中間調処理が施された第2の画像データ、及び第1の画像データに含まれる各画素の属性を表す属性データに基づいて、第2の画像データにおけるジャギーを改善するために、属性データに基づいて、スムージング処理を行うか否かを示す判別信号を出力し、判別信号に応じたスムージング処理を行って、第1の画像データからエッジ補正データを生成し、第2の画像データの画素信号と、エッジ補正生成工程で出力されたエッジ補正データの画素信号とを比較して、より濃度の高い画素信号を出力する。

10

【0023】

なお、エッジ補正生成手段は、あらかじめ定められたサイズの領域内の画素に対するスムージング処理を行うスムージング手段と、第1の画像データを任意の閾値で量子化して、注目画素を含むあらかじめ定められたサイズの領域内のパターンとのマッチングを行い、一致した場合に注目画素のスムージング手段からのデータを出力するマッチング処理手段と、判別信号と第1の画像データとに基づいて、エッジ検出信号を出力するエッジ検出手段と、エッジ検出信号に応じて、第2の画像データに含まれる画素信号及びマッチング処理手段の出力信号の一方を出力する適応スムージング手段と、を含んでいてもよい。

20

【0024】

適応スムージング処理手段は、エッジ検出信号が出力されていないときには第2の画像データに含まれる画素信号を出力し、それ以外のときにはマッチング処理手段の出力信号を出力する、あるいは、マッチング処理手段の出力信号をルックアップテーブルを参照してあらかじめ定められたビット数に変換する変換手段を含んでいてもよい。

【0025】

エッジ検出手段は、注目画素を含むあらかじめ定められたサイズの領域内の画素値の総和とあらかじめ定められた閾値とを比較し、画素値の総和が閾値未満であるときに、エッジ検出信号を出力してもよい。

【0026】

選択手段は、第2の画像データの画素信号とエッジ補正生成手段から出力されたエッジ補正データの画素信号とのビット数が異なる場合、ビット数の少ない画素信号の下位ビットに「1」を詰めて同じビット数にして比較するようにしてもよい。

30

【0027】

スムージング手段及びマッチング処理手段におけるあらかじめ定められたサイズの領域は、 $3 \times 3$ 以上の領域であるのがよい。さらに、あらかじめ定められた閾値が変更可能であり、あらかじめ定められた閾値の変更によってジャギーの改善の度合いを調整するようにしてもよい。

【0028】

また、上記の目的は、上述の画像処理装置の各手段に対応した工程を有する画像処理方法によっても達成される。

40

【発明の効果】

【0029】

本発明によれば、中間調処理によって発生するジャギーを、簡単な構成で改善することができる。

【0030】

これは、スクリーン処理のような画像形成処理を施した画像データ、JPEGのようなエッジ部に画質劣化がある画像データ、さらには、誤差拡散処理を施した画像データに含まれるジャギーに対して有効であり、画像劣化を確実に防止できる。

【発明を実施するための最良の形態】

【0031】

50

以下に、添付図面を参照して、本発明の好適な実施の形態を例示的に詳しく説明する。ただし、以下の実施形態に記載されている構成要素はあくまで例示であり、本発明の範囲をそれらのみに限定する趣旨のものではない。

【0032】

<第1の実施形態>

図2は、本発明に係る画像処理装置の一実施形態としての複合機の構成例を示す概略断面図であり、該複合機は機械的構成としてカラースキャナ部Aとプリンタ部Bとから構成されている。なお、ここでは、好ましい実施形態として、電子写真技術を用いたスキャナ付の複合機で説明するが、本発明が対象としている画像データは、ベクトル情報を表す画像信号である。ベクトル情報を表す画像信号とは、PCのような計算機から送られるPDLデータやPDLを解釈することで生成されるプリミティブな中間データは言うまでもなく、スキャナで読み取った画像信号をベクトル化したものなども含まれている。なお、PDLとは、Page Description Languageの略である。これらのベクトル情報は、テキストやグラフィックなどの各種オブジェクトを示すためのデータであり、最終的にビットマップデータに変換されるものである。また、以下の実施形態では、特に電子写真プロセスの複写機やプリンタなどを対象としているが、インクジェットやディスプレイのような他のプロセスを用いた機器でも適用可能である。

【0033】

図2に示すカラースキャナ部Aにおいて、原稿給送装置201Aは、原稿を最終頁から順に1枚ずつプラテンガラス202A上へ給送する。そして、原稿の読み取り動作終了後、プラテンガラス202A上の原稿を排出するものである。原稿がプラテンガラス202A上に搬送されると、ランプ203Aを点灯し、このランプ203Aを搭載したスキャナユニット204Aを移動させて原稿を露光走査する。この走査による原稿からの反射光は、ミラー205A, 206A, 207Aおよびレンズ208AによってCCDカラーイメージセンサ(以下、単に「CCD」という)209Aへ導かれる。

【0034】

そして、CCD209Aに入射した反射光は、R, G, Bの3色に色分解され色毎の輝度信号として読み取られる。さらに、CCD209Aから出力される輝度信号はA/D変換によってデジタル信号の画像データとして画像処理部(図3の304参照)に入力される。そして、シェーディング補正、階調補正、量子化(N値化)、スムージング処理などの周知の画像処理が施された後、プリンタ部B(305)へ転送される。

【0035】

図2に示すプリンタ部Bにおいて、レーザドライバ221Bは、レーザ発光部201Bを駆動するものであり、画像処理部304から出力された色毎の画像データに応じたレーザ光をレーザ発光部201Bによって発光させる。このレーザ光は感光ドラム202Bに照射され、感光ドラム202Bにはレーザ光に応じた潜像が形成される。

【0036】

そして、この感光ドラム202Bの潜像の部分には現像器203Bによって現像剤であるトナーが付着される。なお、図2では、現像器は、図示の簡略化のため、唯一一つのみが示されているが、C, M, Y, Kの色毎にトナーが用意され、それに応じて4つの現像器が設けられることは、勿論である。また、以上の構成の代わりに感光ドラムや現像器等を色毎に4組設ける構成であってもよい。

【0037】

上述のレーザ光の照射開始と同期したタイミングで、カセット204Bまたはカセット205Bの選択されたいずれかから記録紙が給紙され、転写部206Bへ搬送される。

【0038】

これにより、感光ドラム202Bに付着した現像剤を記録紙に転写することができる。現像剤が転写された記録紙は、定着部207Bに搬送され、定着部207Bの熱と圧力により現像剤の記録紙への定着が行われる。そして、定着部207Bを通過した記録紙は排出ローラ208Bによって排出され、ソータ220Bはこの排出された記録紙をそれぞれ

10

20

30

40

50

所定のピンに収納して記録紙の仕分けを行う。

【0039】

なお、ソータ220Bは、仕分けが設定されていない場合は、最上位のピンに記録紙を収納する。また、両面記録が設定されている場合は、排出口ーラ208Bのところまで記録紙を搬送した後、排出口ーラ208Bの回転方向を逆転させ、フラッパ209Bによって再給紙搬送路へ導く。多重記録が設定されている場合は、記録紙を排出口ーラ208Bまで搬送しないようにフラッパ209Bによって再給紙搬送路210Bへ導く。再給紙搬送路へ導かれた記録紙は上述したタイミングで転写部206Bへ給紙される。

【0040】

なお、色毎の潜像および現像の処理や定着は、上述の記録紙搬送機構を用いて、潜像形成等を4回分繰り返すことによって実現することは周知の通りである。

【0041】

ところで、314はネットワークケーブルであり、一般的にイーサネット（登録商標）と呼ばれるシステムである。これは、10BaseTや10Base5などの物理的なケーブルを用いてTCP/IPなどのプロトコルにより、接続される各ユニット相互の情報授受やデータの転送を行うことができる。無論、ネットワークケーブルを用いた有線に限定されたものではなく、無線を用いても同様な環境構築ができるることは言うまでもない。

【0042】

このようなネットワークケーブル314を介し、ホストコンピュータなどから送信されたPDLデータやディスプレイリストなどは、プリンタに設けられたインターフェースとしてのネットワーク信号受信部315で受信される。このネットワーク信号受信部315には、ホストコンピュータから送信されたデータを解釈して、画像を示す画素単位のピットマップデータを生成するラスタライズ部（不図示）が含まれている。また、PDLデータまたはディスプレイリストは、テキスト（文字）、フラフィック、イメージ等といったオブジェクト毎にその属性を示すデータを保持している。ラスタライズ部では、この属性データを利用して画像データをセ氏英するとともに、各画素の像域信号も生成する。像域信号については後に詳細に説明する。

【0043】

そして、ラスタライズ部で生成された画像データは、画像読み取り部309から入力されたデータと同様に、画像処理部304に入力され、矢印で示したように、フィルタ処理部1604に入力され、必要な画像処理が実行される。また、像域信号はスムージング処理部1605に入力される。カラー画像データの場合、CMYKの色毎の濃度データが生成され、色毎のデータ夫々に対して後段の処理が実行される。

【0044】

なお、ラスタライズ部はネットワーク信号受信部315の外部に設けられるものであってもよい。本実施形態では、この314を介して受信したデータを処理する構成について後述する。

【0045】

図3は、図2に含まれる画像処理装置のデータ処理構成を説明するブロック図であり、図2と同一のものには同一の符号を付してある。

【0046】

図3において、画像読み取り部309は、レンズ301、CCDセンサー302、アナログ信号処理部303等により構成される。そして、レンズ301を介してCCDセンサー302に結像された原稿画像300が、CCDセンサー302によりアナログ電気信号に変換される。変換された画像情報は、アナログ信号処理部303に入力され、サンプル&ホールド、ダークレベルの補正等が行われた後にアナログ・デジタル変換（A/D変換）される。

【0047】

このようにして変換されたデジタル信号は、図4に詳細に示す画像処理部304において、シェーディング補正401、色補正処理402、フィルタ処理403、補正処理4

10

20

30

40

50

04、画像形成処理405、スムージング処理406などが各部で行なわれる。その後、プリンタ305へ出力される。この画像形成処理405とは、入力される1画素あたりNビットの画像データを1画素あたりMビットの画像へと変換する処理を行う。具体的には、本実施の形態ではNを8、Mを4とした。画像形成処理の詳細については、公知な処理である為説明は略するが、スクリーン(ディザ)処理や誤差拡散処理などを指している。

【0048】

本願発明のポイントを含むスムージング処理406以外の処理についても、同様に周知なものであるため、説明を略する。

【0049】

ところで、図4において、ネットワーク信号受信部315からの信号が色補正処理402の後に接続されているが、これは、ネットワークを介して受信した信号がCMYKの濃度データであることを想定している為である。もし、受信した信号がRGBの輝度信号であった場合は、不図示であるが色補正処理402の前に接続されるようになることも言うまでもない。さらに、ネットワークを介して受信した信号がPDL信号の場合は、ネットワーク信号受信部315でビットマップ信号に変換した後、画像処理部304に入力されることは言うまでもない。これら処理は、正確にはネットワークを介して受信した信号はCPU310を介しているのだが、ここでは、信号の流れが分かりやすいように簡略して図4のように示してある。

【0050】

一方、図3に示すプリンタ305は、レーザ等からなる露光制御部(図示せず)、画像形成部(図示せず)、転写紙の搬送制御部(図示せず)等により構成され、入力された画像信号を転写紙上に記録する。

【0051】

また、CPU回路部310は、CPU306、ROM307、RAM308等により構成され、画像読み取り部309、画像処理部304、プリンタ部305、操作部313等を制御し、複合機のシーケンスを統括的に制御する。

【0052】

操作部313には、RAM311、ROM312が予め用意されており、UI上に文字を表示したり、ユーザが設定した情報を記憶したり、表示したりすることが可能な構成となっている。

【0053】

ユーザによって操作部313で設定された情報は、CPU回路部310を介して、画像読み取り部309、画像処理部304、プリンタ305などに送られる構成となっている。

【0054】

以上説明した流れの中で、本願発明のポイントは、画像処理部304に含まれている。以下、画像処理部内のスムージング処理部406について説明を行う。尚、以下の各構成要素については、ソフトウェアで実現したものとし、C言語で作成されたプログラムで示すが、これらは等価な機能をハードウェアやファームウェアで構築してもよい。つまり、各構成要素それまたはいくつかの構成要素がASICなどの専用ハードウェアを用いて構成されていても、ソフトウェアによって得られる効果と同様の効果を得ることができる。

【0055】

図5は、図4に示したスムージング処理部406の構成を説明する概略ブロック図であり、図4と同一のものには同一の符号を付してある。

【0056】

図5においては、1色(ImageData)に対する処理しか示していないが、図4に示したように本処理には4色分のデータが入力されている。つまり、CMYK各色、独立に処理する構成となっており、その内の1色に対する処理を図示している。

【0057】

10

20

30

40

50

まず、はじめに本願発明のポイントであるスムージング処理部 406 の概略動作フローについて図 5 を用いて説明する。

【0058】

スムージング処理部 406 には、画像形成処理（例えばスクリーン処理などの擬似中間調処理）される前の信号である  $N = 8 \text{ bit}$  の上位  $3 \text{ bit}$  と、各画素の属性を示す像域信号  $4 \text{ bit}$  と、画像形成処理された信号である  $M = 4 \text{ bit}$  とが入力される。ここでは、画像形成処理前の信号を `ImageData`（第 1 の画像データ）、像域信号を `AttributeData`（属性データ）、画像形成処理後の信号を `ScreenData`（`InDataS`）（第 2 の画像データ）で表している。

【0059】

像域信号 (`AttributeData`) とは、各画素がテキスト（文字）、イメージ（写真）、グラフィックなどの画像属性のうち、どの属性に相当する画像を構成する画素であるかを示すデータである。この信号は  $4 \text{ bit}$  であり、後述するように属性復号部 (`Attribute Decoder`) 510 により、 $8 \text{ bit}$  の像域信号に変換される。

【0060】

判定手段としての判別部 (`make_ZSG_sig`) 501 では、スムージング処理を行うか否かの判断の元となる判別信号 (`OutDataZ`) が生成される。そして、エッジ補正生成手段としての処理実行部 (`make_AST_sig`) 502 で判別信号に応じたスムージング処理を行った後、その出力を選択手段としてのセレクタ (`selector`) 503 で `ScreenData` と比較する。そして、判別信号に応じたスムージング処理がされた画素データ値と、`ScreenData` の値のうち、画素値が高い方のデータを出力データ (`OutputData`) として出力する構成となっている。

【0061】

以下、上述したスムージング処理部 406 の各部における処理について詳細に説明する。

【0062】

図 6 は、図 5 に示した属性復号部 (`Attribute Decoder`) 510 で行なわれる処理を説明する図である。図 6 の (a) において 601 (`Attribute Decoder input`) で示した部分が、属性復号部に入力される属性データ (`AttributeData`) である。この入力信号 601 の各ビットは、ラインの太さやオブジェクトの有無や文字 / 非文字やベクタ / 非ベクタなどの属性を示している。例えば、 $bit\ 4$  が線や文字の太さを示すフラグ、 $bit\ 5$  がオブジェクトの有無を示すフラグ、 $bit\ 2$  が文字 / 非文字を示すフラグ、 $bit\ 0$  がベクタ / 非ベクタを示すフラグなどである。4 つのビットが  $bit\ 0 \sim bit\ 3$  まで順に並んでいない理由は、4 ビット以上ある信号の中から任意に選択したものである為である。

【0063】

これらの信号をもとに、属性復号部 (`Attribute Decoder`) 510 では  $8 \text{ bit}$  の信号を生成している。本実施の形態で生成した信号の詳細は、図 6 の (b) のテーブルに示した通りであるが、ポイントだけ書き出すと以下のようになる。

【0064】

<code>ImageType0</code> : 00000001(0x01) : 未使用	40
<code>ImageType1</code> : 00000010(0x02) : 未使用	
<code>ImageType2</code> : 00000100(0x04) : グラフィック	
<code>ImageType3</code> : 00001000(0x08) : 線	
<code>ImageType4</code> : 00010000(0x10) : 文字	
<code>ImageType5</code> : 00100000(0x20) : 小さいグラフィック	
<code>ImageType6</code> : 01000000(0x40) : 細い線	
<code>ImageType7</code> : 10000000(0x80) : 小さい文字	

ここで、(b) のテーブルにおいて `REOS` という文字は、ラインや文字が細い / 小さいという意味を表している。この細い / 小さいについては、`PDL` などの画像データを生成し、プリンタに送信するホストコンピュータに設けられたプリントドライバで既に行な

われた判別結果を示している。プリンタドライバの設定次第で細い／小さいと判別される閾値は任意であることは言うまでもない。また、ネットワーク信号受信部 315において判別する構成であってもよい。

【0065】

以上説明したように、属性復号部 (AttributeDecoder) 510 は、4 bit の入力を 8 bit 出力に変換するテーブル変換メモリで構成されている。

【0066】

次に、判別部 (make\_ZSG\_sig) 501 の動作について、図 7A～7C を参照して説明する。図 7A は入出力される信号を示すブロック図であり、図 7B 及び 7C は判別部 501 での処理をプログラム言語 (C 言語) で記載した例を示している。

【0067】

図 7A に示すように、本実施形態の判別部 501 には、前述した属性復号部 (AttributeDecoder) 510 (なお、図 7B のプログラム中には attrdec と記載) から出力された 8 bit の信号 (AttributeDec.Sig.) と、画像形成処理前の信号である imageData の上位 3 bit の信号 InData (=InData') に応じて、図 7B に示すような処理によって判別信号 (OutDataZ) を生成している。なお、reg\_atb とは、モードセレクタ (レジスタ) である。

【0068】

InData 信号については、図 7B の A で示す部分において、任意の数値 (レジスタ: reg\_Zcheck) と比較して、大小関係に応じて判別信号 (OutDataZ) を制御する処理を行っている。また、AttributeDecoder 信号については、図 7B の B で示す部分において、任意のオブジェクトに応じて、判別信号 (OutDataZ) を制御する処理を行っている。図 7B において ThinJudge はレジスタであり、これより、任意の濃度以上でかつ任意のオブジェクトに対してのみスマージング処理をかけるように制御することが可能となっている。例えば、一定値以上の文字のみに対してスマージング処理を行うなどの制御を可能とするものである。

【0069】

ところで、本実施形態の判別部 501 は、この制御だけに限定したものではなく、他の幾つかの制御も可能な構成としてある。これら制御の切替えは、上述したモードセレクタ (レジスタ reg\_atb) の信号に応じて行なわれる。

【0070】

例えば、図 7C の C で示した部分において、reg\_atb=1 の場合は、画像形成前の信号 (InData) の大きさだけに応じて、判別信号 (OutDataZ) が制御される。言うまでもないが、reg\_Zcheck (レジスタ) に応じて任意に制御可能である。

【0071】

そして、図 7C の D で示した部分において、reg\_atb=2 の場合は、画像形成処理された信号 (InDataS) の大きさだけに応じて、判別信号 (OutDataZ) が制御される。これも同様に、reg\_Air (レジスタ) に応じて任意に制御可能である。

【0072】

最後にレジスタ reg\_atb の値がその他の場合の処理が、図 7C の E で示した部分である。これは、出力信号がすべて 0 になるため、本処理の出力位相 (タイミング) を保った状態でスルー出力を実現するために用意されている。

【0073】

次に、処理実行部 (make\_AST\_sig) 502 について図 1 を参照して説明する。なお、図 1 には、説明の便宜上セレクタ 503 を含めて示している。この部分は従来例と比較して、コストダウンの効果が一番大きいブロックである。

【0074】

図 1 に示したように、処理実行部 (make\_AST\_sig) 502 には、前述した判別部 (make\_ZSG\_sig) で生成した 1 bit の判別信号 (OutDataZ) と、画像形成処理前の上位 3 bit の信号 (InData) と、画像形成処理後 (例えばスクリーン処理などの疑似中間調処理後) の 4 bit の信号 (InDataS) とが入力され、4 bit の信号が出力される構成となっ

10

20

30

40

50

ている。これら3つの信号それぞれは、図示したような FIFOメモリ521～523に入力されており、所定のライン数だけ遅延されるようになっている。本実施形態では、このメモリのサイズを従来と比較して大きく削減している。さらに、従来と同等な効果を維持しつつ、処理系も簡素化している。

#### 【0075】

そして、FIFOメモリ521及び522を介して、判別信号(OutDataZ)と画像形成処理前の上位3bitの信号(InData)とが再生部(Zexistence)524に入力され、エッジ部の信号のみが再生される。そしてセレクタ(selector)503で、選択部(make\_select\_sig)525の出力に応じて、画像形成処理後の4bitの信号(InDataS)と、再生部(Zexistence)の出力信号とのいずれかが選択されて出力される構成となっている。  
なお、選択部(make\_select\_shig)525では、画像形成処理前の上位3bitの信号(InData)と、画像形成処理後の4bitの信号(InDataS)との差(濃度値の差)に応じて選択信号が生成される。本実施形態では処理をこのように簡略化することによって、コストダウンを実現している。具体的な処理については後述する。

#### 【0076】

まず、選択部(make\_select\_sig)525について図8A及び8Bを参照して説明する。図8Aは選択部525の入出力を示すブロック図であり、図8Bは選択部525での処理をプログラム言語(C言語)で記述した例を示している。

#### 【0077】

選択部525には、前述した通りInDataとInDataSとが入力されている。基本的には、これらの信号の大小関係を比較し、InDataの方が大きければ1を出力し、小さければ0を出力するといった処理を行う。ただし、これらInDataとInDataSとの比較は、3ビットと4ビットの比較となるため、図8Bのプログラム例に示したようなビット揃え処理を行った後に比較を行う。プログラム中では、InDataをsel\_InDataに変換し、InDataSをsel\_InDataSに変換した後、比較している。

#### 【0078】

このように、InDataとInDataSとを比較し、濃度の濃い方、即ちビット値が大きい方の信号を選択する構成にすることによって、エッジが細ったり、濃度が薄くなったりして、全体の画質レベルが下がることを防止できる。

#### 【0079】

つまり、後述する再生部(Zexistence)524で再生したエッジ信号の方が、元のInDataSのエッジ信号より薄かった場合に、再生部の出力信号を出力すると、上述したようにエッジ部が薄くなり、全体の画質レベルが下がる。これを防止するように選択部(make\_select\_sig)525が設けられている。

#### 【0080】

さらに本実施形態の特徴でもある再生部(Zexistence)524での処理を、図9のブロック図を参照して詳細に説明する。

#### 【0081】

再生部(Zexistence)524には、3ラインのInDataとOutDataZ、そして、注目ラインのInDataSとが入力されている。そして、内部に3×3のスムージング処理部(x3smooth)5241、パターンマッチング処理部(miniSST)5242、エッジ検出処理部(BorderCount&DensCheck)5243、適応スムージング処理部(AdaptiveSmoothing)5244を有し、各部での処理を行って、4bitの出力信号を生成している。以下、順に各部での処理を説明する。

#### 【0082】

まず、3×3スムージング処理部(x3smooth)5241では、InDataに含まれる3×3領域の画素の濃度平均を求めている。濃度平均を求める処理は公知のため、詳細な説明は省略するが、例えば、9画素内にある濃度1の画素(記録画素)の数を求め、9で割るといった演算を行っている。ただし、本実施の形態では、演算処理部の負荷を減らすため、3×3の領域内にある記録画素の数を求めた後、3ビットシフト(=÷8相当)を行って

10

20

30

40

50

出力信号 (smoothData) を得ている。

【0083】

次に、パターンマッチング処理部 (miniSST) 5242 では、パターンマッチングにより、プリント解像度に依存したジャギーを改善する為のスムージング処理を行っている。プリント解像度に依存したジャギーとは、例えば、プリントの解像度が 300 dpi なら、フォントやラインのガタツキとして、図 14 に示したような L 字型の部分が発生する。そこで、図 12 に示したようなパターンと、InDataを任意の閾値で 2 値化した信号 (不図示) とを  $3 \times 3$  の領域でマッチングする。そして、一致した箇所を、図 9 に示した  $3 \times 3$  スムージング処理部 (x3smooth) 5241 の出力信号 (smoothData) で置き換える処理を行っている。なお、図 12 に示したパターンは本実施の形態で用いた一例にすぎず、これに限定したものではないことは言うまでもない。このようにプリント解像度に依存したジャギーが改善された 3 bit の信号 (patternCK) が後述する適応スムージング処理部 (AdaptiveSmoothing) 5244 に入力される。10

【0084】

一方、エッジ検出処理部 (BorderCount&DensCheck) 5243 は、OutDataZ信号を用いてエッジ検出信号を生成している。図 10A はエッジ検出処理部 5243 の入出力を示しており、図 10B はエッジ検出処理部 5243 の処理をプログラム言語 (C 言語) で記述した例を示している。

【0085】

図 10A に示したように、エッジ検出処理部 5243 には、前述した OutDataZ (=DataZ) 信号 3 ラインと、InData 信号 1 ライン (注目ライン) とが入力され、エッジ検出信号 bc (=out) を生成している。具体的には、まず、図 10B において A で示した部分で、OutDataZ (=DataZ) の 1 bit 信号を用いて  $3 \times 3$  の領域内の濃度 1 の画素 (記録画素) の数をカウントする (sum=bc)。 $3 \times 3$  の領域なので合計 9 が最大になる。つまり、4 bit 信号である。なお、図 10B のプログラム中に valCheck() で示された関数は、画像の端部判定をしているだけであり、本質的な処理ではないためここでの説明は省略する。そして、上述した sum=bc を図 10B の B で示す部分で、再処理して最終的な 4 bit 信号 bc を生成して出力する構成となっている。20

【0086】

ここでは、任意のレジスタ (reg\_Zcheck) に格納された濃度値と InData (=data) に含まれる画素の濃度値が近くて、かつ InData (=data) が 0 ならば、前述のようにして求めた sum の値をクリアする (値をゼロに置換する) 処理を行っている。これにより、グラデーションなどの濃度が徐々に変化する領域において、誤ったエッジ信号により階調が不連続となるのを改善できる。つまり、図 7B で説明した InData の濃度と属性復号部の出力信号とから図 7B の A で示す処理を実行すると、グラデーションの途中で出力信号 (OutDataZ) を 1 としてしまう場合が生じる為、それをクリアする処理をここで行っている。30

【0087】

最後に、適応スムージング処理部 (AdaptiveSmoothing) 5244 の詳細について、図 11A 及び図 11B を参照して説明する。図 11A は適応スムージング処理部 5244 の入出力を示しており、図 11B は適応スムージング処理部 5244 の処理をプログラム言語 (C 言語) で記述した例を示している。40

【0088】

適応スムージング処理部 5244 には、入力信号として前述した PatternCK と InData と InDataS と bc とが入力され、4 bit の信号が出力される構成となっている。具体的には、条件 「(bc==0 || bc>=astPower)かつ (InData!=0)」 を満たす場合に、画像形成された信号 InDataS をスルー出力する。それ以外の場合には、条件 「ast\_params->sstAllOffSW==0 (レジスタ)」 を満たすときに、InDataS を LUTE というテーブルを介して出力する。上記 2 つの条件のいずれも満たさない場合には、LUTE というテーブルを介して patternCK を出力する構成となっている。

【0089】

10

20

30

40

50

ここで、LUTEはメモリで構成されたルックアップテーブルであり、3 bit 入力4 bit 出力のメモリ空間を有したものであるが、基本的には、リニアな出力変換を行うように設定されている。ただし、プリンタの記録特性に応じて、非線形な出力となるように設定してもよいのはもちろんである。また、エッジ部の出力ビット数を4 bit 以外にしたいときも、このテーブルを用いて変換可能である。

#### 【0090】

ところで、上述した第2の条件式の「astPower」は、レジスタであり、本実施形態では、この値に応じて、エッジ再処理信号への置き換え幅を調整可能であることも特徴としている。実際にエッジ再処理信号の置き換え幅を変えた画像の例を図13に示した。astPowerの値を小さくした場合 (level1) を (a) に示し、astPowerの値が大きくなるにつれて、(b) (level2)、(c) (level3) に示すようになる。このときの画像データは、(a) (b) (c) でエッジ部のハーフトーン画素数が増えている。この画素数が増すと、エッジ部の滑らかさも増すことになる。このパラメータは、ハーフトーン画素の追加によって文字が縁取られたような画像にならない程度に調整して用いられる。

本機能は、従来手法にはなかったものであり、これにより一層高機能な処理を実現している。

#### 【0091】

一方、上記の第2の条件式における「ast\_params->sstAllOffSW」もレジスタであり、上述のように0に設定されている場合は、LUTEを介してInDataが出力される構成となっている。つまり、前述したパターンマッチング処理部 (miniSST) 5242での処理結果を反映しないスマージングも行える構成となっている。パターンマッチング処理部 5242での処理を反映しないということは、プリンタ解像度より高い解像度では、スマージング処理を行わないということである。

#### 【0092】

以上説明した各部での処理を施した結果が、図1で示した再生部 (Zexistence) 524から出力され、セレクタ (selector) 503で、選択部 (make\_select\_shig) 525の出力に応じて、画素毎に画像形成処理後の4 bit の信号 (InDataS) と、再生部 (Zexistence) の出力信号とのいずれかが選択されて、スマージング処理 406の結果としてプリンタ305に出力される。出力されるイメージは、前述した図13のとおりである。

#### 【0093】

図13は、本実施形態の画像処理装置におけるエッジ処理結果を示す図であり、着目箇所は、エッジ部である。図13に示したようにスクリーンのパターンの間を埋めるようにデータが薄く、かつ、可変な幅で出力されることを特徴としている。本実施形態の構成により、この処理を従来と比較してローコストにかつ高機能に実現できる。

#### 【0094】

不図示だが、さらに、JPEGのような非可逆な圧縮処理を施した画像に対しても、問題なくエッジ部の改善ができるという特徴も有している。

#### 【0095】

その理由は、エッジ検出用のデータを作成する際、8ビット画像データの上位側3ビット信号より1ビットの像域信号を作成しているため、圧縮による微妙な劣化情報を除去できるという効果が得られるためである。

#### 【0096】

以上説明したように、本実施形態によれば、プリンタの解像度によるジャギーはもちろんのこと、175線や133線等のスクリーンを用いた疑似中間調処理によってエッジ部に発生するジャギーをも改善することが可能となる。このようなスクリーンのジャギーは、プリンタの解像度より遙かに低い解像度で発生するため、ハーフトーン文字やラインで顕著に目立っていたが、これを改善することが可能である。

#### 【0097】

さらに、エッジ再処理信号への置き換え幅を調整可能にすることで、よりラチチュードの高い処理を実現した。置き換え幅の調整により、図13に示した効果が得られる。これ

10

20

30

40

50

によって、様々なプリンタ特性に対応できるようになった。

【0098】

さらに、本実施形態を用いると、スキャナで読み込んだイメージ画像をベクトル化したものにも容易に対応可能である。なお、ベクトル化とは、ビットマップデータをアウトラインデータに変換することを意味する。

【0099】

<第2の実施形態>

以下、本発明に係る画像処理装置の第2の実施形態について説明する。以下の説明では、上記第1の実施形態と同様な部分については説明を省略し、第2の実施形態の特徴的な部分を中心に説明する。

10

【0100】

上記第1の実施形態は、入力画像信号の解像度が600dpiであったのに対し、本実施形態では入力解像度信号が1200dpi及び600dpiの両方の解像度に対応できる構成となっている。

【0101】

図17は、本願発明のポイントであるスマージング処理部406'の構成を説明する概略ブロック図であり、第1の実施形態に関して説明した図5に対応しており、同様な部分を同じ参照符号で示している。

【0102】

図17に示したスマージング処理部406'を図5のスマージング処理部406と比較する。すると、1200dpiに対応するために、1701で示すビットマスク部(ScreenBitMask)と1702で示すビット変換部(bit32\_conv)が追加となっている。また、後述するように処理実行部(make\_AST\_sig)502'も1200dpiに対応するために変更されている。

20

【0103】

これらブロックの追加および変更により、第1の実施形態と同様な基本構成で、1200dpiへの対応を実現した。このように本実施形態は、600dpiと1200dpiとの両方の解像度に同一のハード構成で対応可能とした点を特徴とする。

【0104】

以下、第1の実施形態に関して説明した図5の構成と異なる各ブロックについて詳細に説明する。

30

【0105】

図18Aはビットマスク部(ScreenBitMask)1701の入出力信号を示すブロック図であり、図18Bはビットマスク部1701での処理をプログラム言語(C言語)で記載した例を示している。

【0106】

図18Bにおいて、「ast\_params->bit\_select」で示されのはパラメータであり、処理解像度が1200dpiであるか600dpiであるかに応じて、2/3あるいは0/1が選択される構成となっている。

【0107】

これは、後述するFIFOメモリの容量を削減するために、1200dpiモードの際に上位ビットのみを使用して下位ビットをマスクするような構成となっているためである。ただし、ハード構成上の関係で、各ブロックを接続する信号線のビット数は600dpiのビット数(4bit)に合わせてある。そのため、1200dpiの処理時も出力ビット数を4bitと記載してある。繰り返しになるが、1200dpiの処理時はFIFOメモリに格納するための有効データ量を削減するようにビットマスク部は構成されている。

40

【0108】

なお、第1の実施形態において述べたのと同様に、本ブロックへ入力される信号は、スクリーニング処理或いは誤差拡散処理などによって量子化されたものである。

50

## 【0109】

次に、ビット変換部 (bit32\_conv) 1702について、図19Aの入出力信号を示すブロック図、及び図19Bのビット変換部1702での処理をプログラム言語（C言語）で記載した例を参照して説明する。

## 【0110】

図19Bにおいて、「ast\_params->dpi\_select」で示されるパラメータは、1200dpi出力時のときのみ1に設定される構成となっている。

## 【0111】

600dpi処理と共にブロックになっているため、図19Aに示すように入出力のビット数が3bitになっている。しかし、1200dpi処理時には図19Bに示すように、パラメータast\_params->bit32\_conv1~7に設定された値が出力され、2bit化されることになる。2bit化された信号は、上詰めされた状態で出力される構成となっている。そして出力された信号は、後述する処理実行部（make\_AST\_sig）502'に入力され、FIFOメモリに格納される。

## 【0112】

図20は、本実施形態の処理実行部（make\_AST\_sig）502'の構成を示すブロック図である。図1に示した第1の実施形態の処理実行部502と比較すると、本実施形態の処理実行部502'には、FIFOメモリを用いてデータを遅延させるブロックが含まれている。以下、この遅延について説明する。

## 【0113】

まず、判別信号（OutDataZ）については、1200/600dpiの解像度に依存せず、第1の実施形態と同様にFIFOメモリ521によって1bitの信号を2ライン遅延させる構成となっている。よって、1200dpi処理を行ってもメモリ容量は増加しない。

## 【0114】

一方、画像形成処理後の4bitの信号InDataSについては、600dpiでの処理時は4bit1ラインを遅延させ、1200dpiでの処理時は2bit1ラインを遅延させる構成となっている。すなわち、1つのFIFOメモリを、600dpiでの処理時は4bit1ラインを遅延させるメモリ523として用い、1200dpiでの処理時は2bit1ラインを遅延させるメモリ523'として用いる。このように、1200dpiでの処理時にはメモリの長さ方向（幅）が2倍になってしまいが深さ方向（高さ）のbit数を半分にすることで、トータルのメモリ容量が同じとなるようにしてある。これにより、InDataSに関しては、1200dpiでの処理を可能としつつトータルのメモリ容量を増大させる必要が無い。

## 【0115】

言うまでもないが、1200dpiの1ビット信号が本ブロックに入力されるのは、図18A及び図18Bに関して説明したように、ビットマスク部（ScreenBitMask）1701でパラメータast\_params->bit\_selectが3に設定された場合である。

## 【0116】

次に、画像形成処理前の3bitの信号InDataにかかるFIFOメモリ522'について説明する。600dpiでの処理時は、図に記載したように3bit信号を2ライン遅延させる構成となっている。しかし、1200dpiでの処理時は、図21に関して後述するように、入出力変換部2001によってFIFOメモリの前後でビット変換が行われるように構成されている。

## 【0117】

図21A～21Cを参照して、入出力変換部2001による1200dpiでの処理について説明する。図21Aは入出力変換部2001の構成をFIFOメモリ522'と共に示すブロック図である。

## 【0118】

まず、入力変換部（bit21\_conv）2101によってInDataから1bit信号と2bit

10

20

30

40

50

信号とが生成される。図21Bは入力変換部2101に含まれるbit21\_convモジュールでの処理をプログラム言語（C言語）で記載した例を示している。この1bitと2bitの信号は、1ライン遅延された2bit信号と、さらに1ライン遅延された1bit信号とからなることを特徴としている。つまり、注目画素から離れるとピット精度が落ちる構成となっている。その状態でFIFOメモリ522'に入力することにより、メモリ容量を600dpi処理時から増大させないようにしている。

#### 【0119】

一方、FIFOメモリ522'からの読み出し時は、bit21\_convと、bit23\_convとの2つのモジュールを含む出力変換部2102によって、3bit信号に変換される。bit21\_convモジュールでの処理は図21B、bit23\_convモジュールでの処理は図21Cにそれぞれプログラム言語（C言語）で記載された例に示される通りである。これらの出力時の変換のポイントをまとめると、前述したパラメータast\_params->dpi\_selectが1、すなわち、1200dpiが指定された場合、ast\_params->bit21\_convやast\_params->bit23\_conv1~3のパラメータに従い、各ライン遅延のデータを3bit信号に変換（置換）する構成となっている。

#### 【0120】

最後に、FIFOメモリ523又は523'からの出力された信号をピットシフトするピットシフト部(ScreenBitShift)2002について、図22Aの入出力信号を示すプロック図、及び図22Bの処理をプログラム言語（C言語）で記載した例を参照して説明する。

10

20

#### 【0121】

ピットシフト部2002での処理の詳細は図22Bに記載した通りであり、詳しい説明は省略する。しかし、ポイントとしては、入力されたInDataSのピット数が異なっても、第1の実施形態で説明した600dpi用のハード構成が共通で使えるようにピット数を4bitにそろえるようにピットシフトを行っている。

#### 【0122】

以上説明したように、本実施形態によれば、600dpiでの処理を行う第1の実施形態と同様なハード構成で、かつ、同一容量のFIFOメモリを用いて、1200dpiでの処理を行うことができる。つまり、本実施形態によれば、コストアップすることなく600dpi及び1200dpiの両方での処理が実現可能である。

30

#### 【0123】

##### <その他の実施の形態>

以上説明したように本発明は、画像形成処理後のデータを使用しているため、画像形成処理に忠実な補正ができる特徴としている。つまり、Faxのような解像度の低いプリンタにも適しているといえる。ただし、従来の手法と異なり、本発明によれば、JPEGのような非可逆圧縮法を施した画像に対しても、エッジ部の画質劣化を補うような出力が可能となるという効果が得られる。

#### 【0124】

また、スクリーン処理のような画像形成処理だけでなく、誤差拡散系の画像形成処理を施した画像に対しても効果がある。誤差拡散系処理を施したハーフトーン文字やラインは、ジャギーとは異なるガタツキがあるが、それをも改善可能とした。

40

#### 【0125】

さらに、JPEGのような非可逆な圧縮を施すことにより発生するエッジ部のモスキートノイズに対しても効果がある。つまり、圧縮により発生したエッジ部の劣化を本発明では改善可能である。

#### 【0126】

本発明は、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

#### 【0127】

例えば、本発明を画像処理装置に適用すると、スキャナで読み込んだイメージ画像、コ

50

ンピュータ機器から送信した P D L 画像の両方におけるエッジ部の画質を改善することが可能となる。

【 0 1 2 8 】

なお、本発明は、前述した実施形態の機能を実現するソフトウェアのプログラムを、システム或いは装置に直接或いは遠隔から供給し、そのシステム或いは装置のコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される場合を含む。その場合、プログラムの機能を有していれば、形態は、プログラムである必要はない。また、上述したように、上記プログラムを元に設計された A S I C などのハードウェアによっても同様の処理を実行することが可能である。

【 0 1 2 9 】

10

従って、本発明の機能や処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明のクレームでは、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

【 0 1 3 0 】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、 O S に供給するスクリプトデータ等、プログラムの形態を問わない。

【 0 1 3 1 】

20

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、 M O 、 C D - R O M 、 C D - R 、 C D - R W などがある。或いは、磁気テープ、不揮発性のメモリカード、 R O M 、 D V D ( D V D - R O M , D V D - R ) などがある。

【 0 1 3 2 】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのサイトに接続し、該サイトから本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードすることによっても供給できる。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるサイトからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせる WWW サーバも、本発明の範囲に含まれるものである。

30

【 0 1 3 3 】

また、本発明のプログラムを暗号化して C D - R O M 等の記憶媒体に格納してユーザに配布し、所定の条件をクリアしたユーザに対し、インターネットを介してサイトから暗号化を解く鍵情報をダウンロードさせ、その鍵情報を使用することにより暗号化されたプログラムを実行してコンピュータにインストールさせて実現することも可能である。

【 0 1 3 4 】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される他、そのプログラムの指示に基づき、コンピュータ上で稼動している O S などが、実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現され得る。

40

【 0 1 3 5 】

さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わる C P U などが実際の処理の一部または全部を行い、その処理によっても前述した実施形態の機能が実現される。

【 0 1 3 6 】

例えば、パーソナルコンピュータにインストールされる記録装置のドライバでこれらの

50

処理を行わせるようにする場合が、これに相当する。

【図面の簡単な説明】

【0 1 3 7】

【図1】本発明の実施形態の処理実行部の構成を示すブロック図である。

【図2】本発明に係る画像処理装置を適用可能な複合機の一例を示す概略断面図である。

【図3】図2に示した画像処理装置のデータ処理構成を説明するブロック図である。

【図4】図3に示した画像処理部の一般的構成を説明するブロック図である。

【図5】図4のスムージング処理部の構成を示すブロック図である。

【図6】図5に示した属性復号部(Attribute Decoder)510で行なわれる処理を説明する図である。

10

【図7A】図5の判別部(make\_ZSG\_sig)501の入出力信号を示すブロック図である。

【図7B】判別部501での処理をプログラム言語で記載した例を示す図である。

【図7C】判別部501での処理をプログラム言語で記載した例を示す図である。

【図8A】選択部(make\_select\_sig)525の入出力を示すブロック図である。

【図8B】選択部525での処理をプログラム言語で記載した例を示す図である。

【図9】図1の再生部(Zexistence)524の構成を示すブロック図である。

【図10A】図9のエッジ検出処理部(BorderCount&DensCheck)5243の入出力を示すブロック図である。

【図10B】エッジ検出処理部5243での処理をプログラム言語で記載した例を示す図である。

20

【図11A】図9の適応スムージング処理部(AdaptiveSmoothing)5244の入出力を示すブロック図である。

【図11B】適応スムージング処理部5244での処理をプログラム言語で記載した例を示す図である。

【図12】図9のパターンマッチング処理部(miniSST)5242で用いるパターンの例を示す図である。

【図13】本発明の実施形態の画像処理装置におけるエッジ処理の結果を示す図である。

【図14】文字のエッジ部などで発生するジャギーの様子を示した図である。

【図15】従来の画像処理装置における文字処理状態を説明する模式図である。

【図16】スクリーン処理のような擬似中間調処理で発生したジャギーを改善する従来の例を示した図である。

30

【図17】第2の実施形態におけるスムージング処理部の構成を示すブロック図である。

【図18A】図17のビットマスク部1701の入出力を示すブロック図である。

【図18B】ビットマスク部1701での処理をプログラム言語で記載した例を示す図である。

【図19A】図17のビット変換部1702の入出力を示すブロック図である。

【図19B】ビット変換部1702での処理をプログラム言語で記載した例を示す図である。

【図20】本発明の第2の実施形態の処理実行部の構成を示すブロック図である。

【図21A】図20の入出力変換部2001の構成をFIFOメモリ522と共に示すブロック図である。

【図21B】図21Aのbit21\_convモジュールでの処理をプログラム言語で記載した例を示す図である。

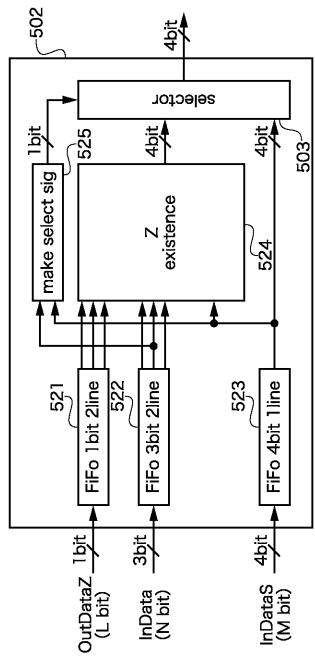
【図21C】図21Aのbit23\_convモジュールでの処理をプログラム言語で記載した例を示す図である。

【図22A】図20のビットシフト部2002の入出力を示すブロック図である。

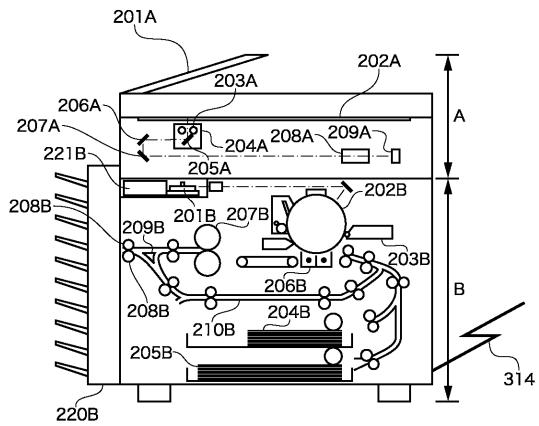
【図22B】図20のビットシフト部2002での処理をプログラム言語で記載した例を示す図である。

40

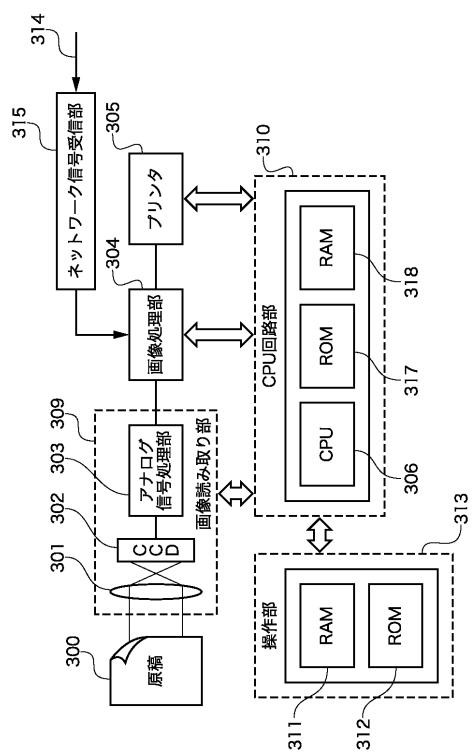
【図1】



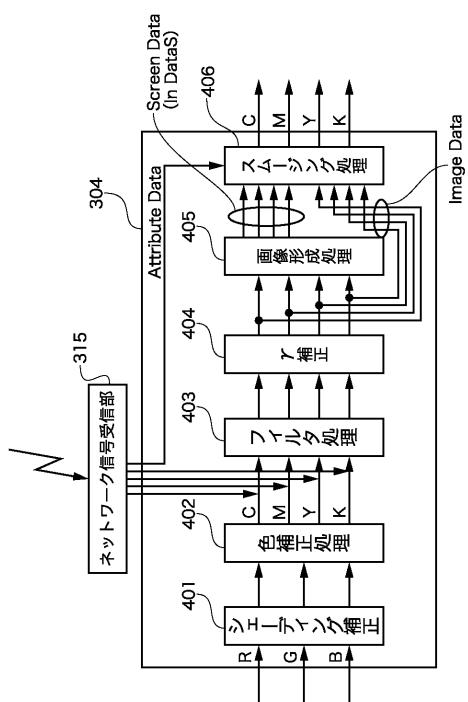
【図2】



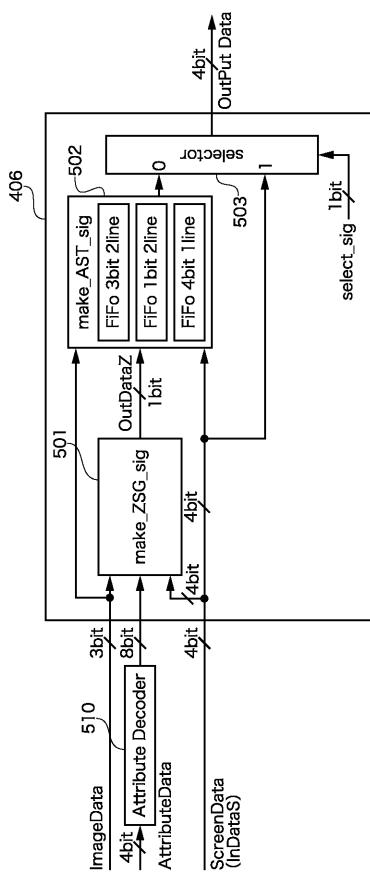
【図3】



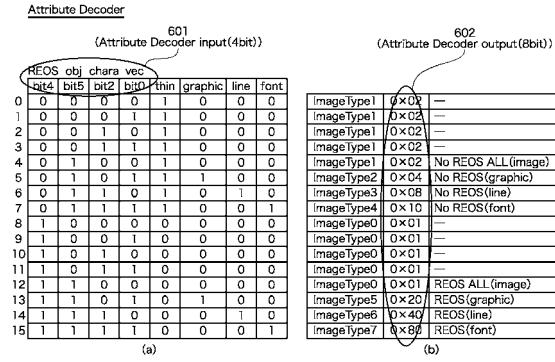
【図4】



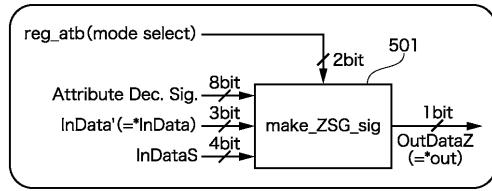
【図5】



【 四 6 】



【図7A】



【図7B】

【図7C】

```

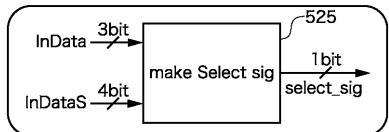
else if( reg_ab==1X
  iff( reg_Zcheck[0]<=InData[yy][xx] ) || ( reg_Zcheck[1]<=InData[yy][xx] ) || ( reg_Zcheck[2]<=InData[yy][xx] ) ) {
  OutDataZ[yy][xx]=1;
  else{
    OutDataZ[yy][xx]=0;
  }
}

else if( reg_ab==2X
  iff( (InData[yy][xx]>=reg_ab) ) {
  OutDataZ[yy][xx]=1;
  }
  else{
    OutDataZ[yy][xx]=0;
  }
}

else{
  OutDataZ[yy][xx]=0;
}
}

```

【図 8 A】



【図 8 B】

```

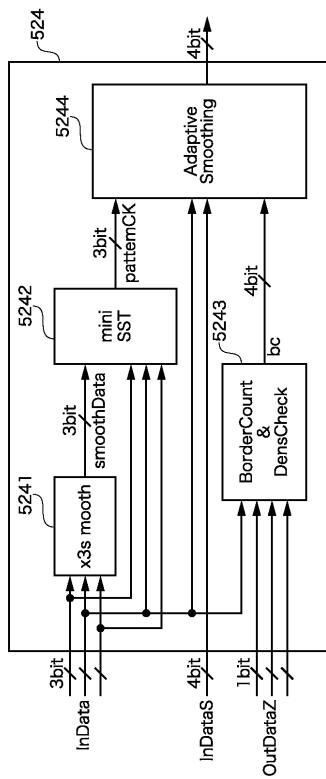
/*=====make select sig=====*/
if((ast_params->edgeLUTsel)==1)sel_InData=LUTE[InData[yy][xx]];
else
    sel_InData=(InData[yy][xx]<<1)+1;

sel_InDataS=(InDataS[yy][xx]<<(4-(ast_params->InDataSBits));
add=(unsigned char)pow(2.0, (double)(4-(ast_params->InDataSBits))-1);
sel_InDataS+=add;

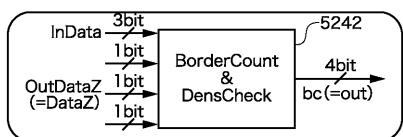
if(sel_InDataS<=sel_InData){ select_sig=1;
else{ select_sig=0;
}

```

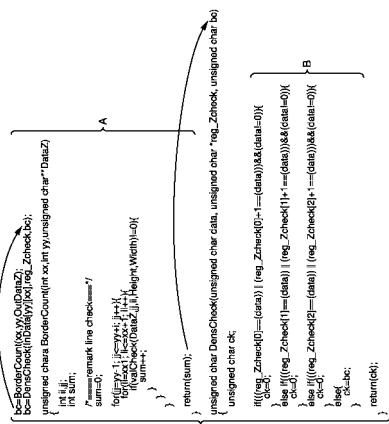
【図 9】



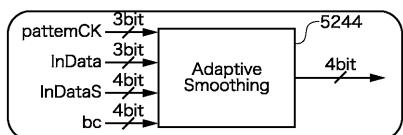
【図 10 A】



【図 10 B】



【図 11 A】



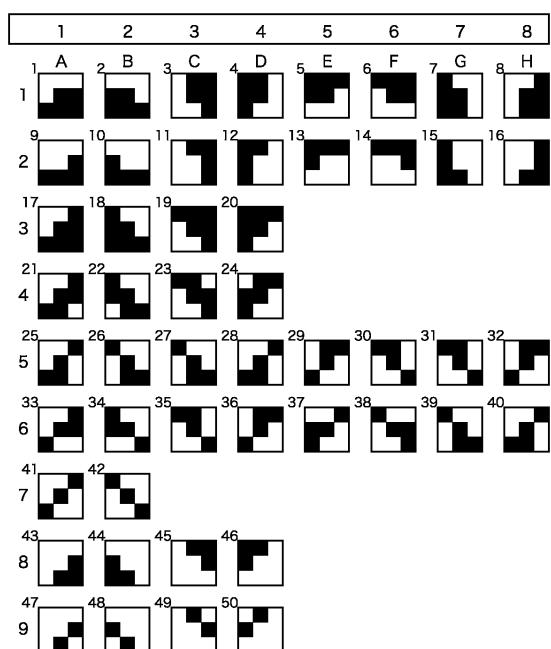
【図 11 B】

```

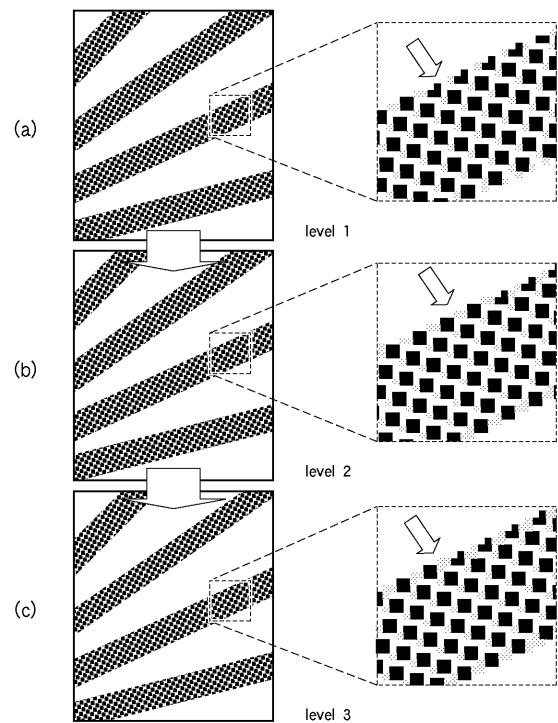
unsigned char AdaptiveSmoothing(yy,yy,InData,InDataS,ast_params,LUTE, bc,reg_Zcheck,patternCK,astPower)
{
    int xx,yy;
    unsigned char *InData;
    unsigned char **InDataS;
    AST_Params*ast_params;
    unsigned char *LUTE;
    unsigned char bc;
    unsigned char patternCK;
    unsigned char astPower;
    {
        unsigned char out;
        if(bc==0 bc>>astPower)&&(InDataS[yy][xx]==0){ //0-SST ALL OFF
            out=InDataS[yy][xx];
        }
        else{
            if(ast_params->ss1AllOff==0){ //1-SST ON(access to ss1OnOff[1-9]para.)
                out=LUTE[InDataS[yy][xx]];
            }
            else{
                out=LUTE[patternCK];
            }
        }
        return(out);
    }
}

```

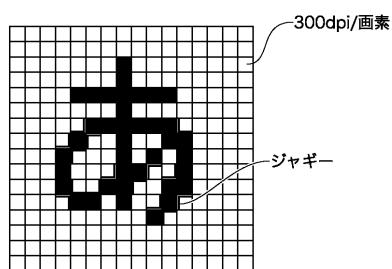
【図12】



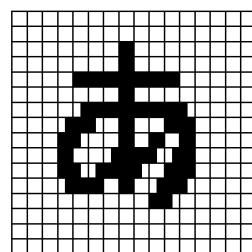
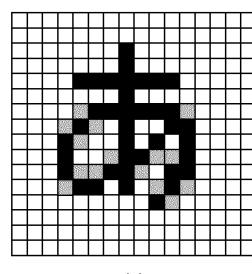
【図13】



【図14】



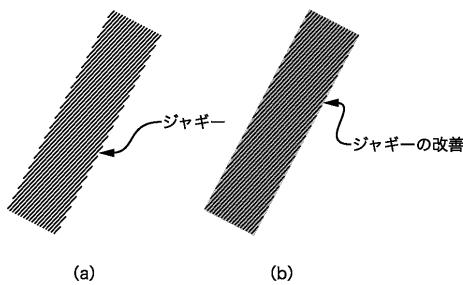
【図15】



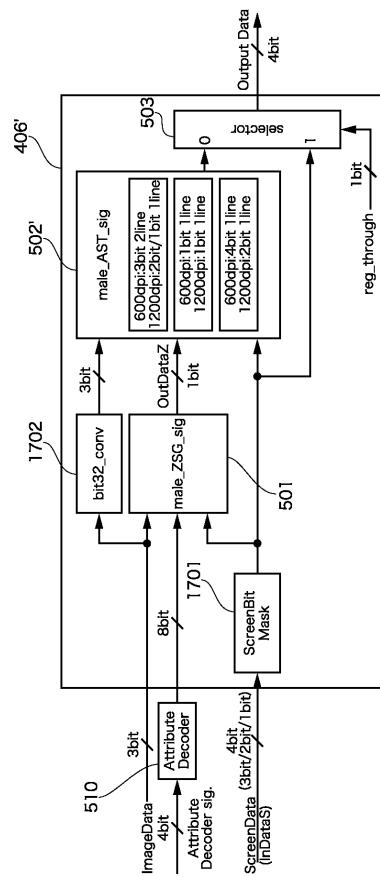
(a)

(b)

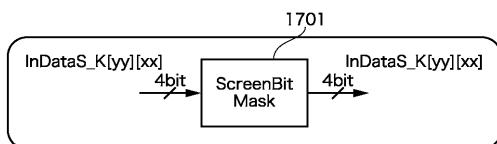
【図16】



【図17】



【図18A】



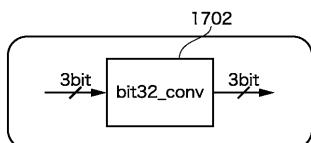
【図18B】

```

if((ast_params->bit_select)==3){ //1bit mode(600/1200dpi)
  InDataS_K[yy][xx]=(InDataS_K[yy][xx])&(0x08);
}
else if((ast_params->bit_select)==2){ //2bit mode(600/1200dpi)
  InDataS_K[yy][xx]=(InDataS_K[yy][xx])&(0x0C);
}
else if((ast_params->bit_select)==1){ //3bit mode(600dpi)
  InDataS_K[yy][xx]=(InDataS_K[yy][xx])&(0x0E);
}
[through]//4bit mode(600dpi)

```

【図19A】

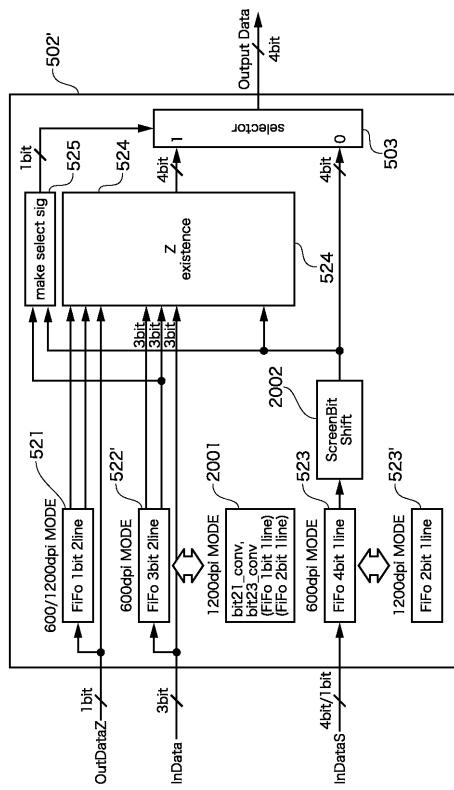


```

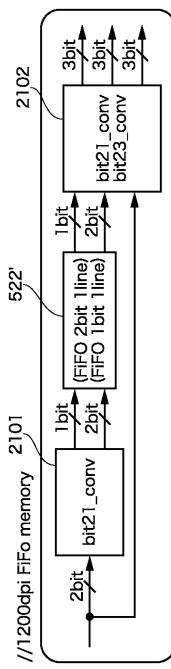
/*
=====
bit32_conv module(3bit>2bit)=====
*/
if((ast_params->bit_select)==3){
  InData_K[yy][xx]=ast_params->bit32_conv7;
}
else if((ast_params->bit_select)==2){
  if((InData_K[yy][xx]==6) InData_K[yy][xx]=ast_params->bit32_conv6;
  else if((InData_K[yy][xx]==5) InData_K[yy][xx]=ast_params->bit32_conv5;
  else if((InData_K[yy][xx]==4) InData_K[yy][xx]=ast_params->bit32_conv4;
  else if((InData_K[yy][xx]==3) InData_K[yy][xx]=ast_params->bit32_conv3;
  else if((InData_K[yy][xx]==2) InData_K[yy][xx]=ast_params->bit32_conv2;
  else if((InData_K[yy][xx]==1) InData_K[yy][xx]=ast_params->bit32_conv1;
  else InData_K[yy][xx]=0;
}

```

【図20】



【図21A】



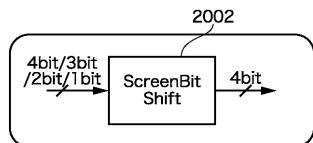
【図21B】

```
/*=====bit21_conv module=====*/
if(ast_params->dpi_select==1){
  for(yy=0; yy<Height; yy++){
    for(xx=0; xx<Width; xx++){
      if(InData[yy][xx]<=(ast_params->bit_th)){
        InData_FiFoUpDown[yy][xx]=0+
      }
      else{
        InData_FiFoUpDown[yy][xx]=(ast_params->bit21_conv)*2+1;
      }
    }
  }
}
else//600dpi
for(yy=0; yy<Height; yy++){
  for(xx=0; xx<Width; xx++){
    InData_FiFoUpDown[yy][xx]=InData[yy][xx];
  }
}
```

【図21C】

```
/*=====bit23_conv module(only remark pixel)=====*/
if(ast_params->dpi_select==1){
  for(yy=0; yy<Height; yy++){
    for(xx=0; xx<Width; xx++){
      /*=====bit11_conv module(2bit>3bit)=====*/
      if(InData[yy][xx]==3) InData_FiFoCenter[yy][xx]=ast_params->bit23_conv3;
      else if(InData[yy][xx]==2) InData_FiFoCenter[yy][xx]=ast_params->bit23_conv2;
      else if(InData[yy][xx]==1) InData_FiFoCenter[yy][xx]=ast_params->bit23_conv1;
      else InData_FiFoCenter[yy][xx]=0;
    }
  }
}
else{
  for(yy=0; yy<Height; yy++){
    for(xx=0; xx<Width; xx++){
      InData_FiFoCenter[yy][xx]=InData[yy][xx];
    }
  }
}
```

【図22A】



【図22B】

```

/*=====ScreenBitShift module(0:4bit 1:3bit 2:2bit 3:1bit)=====*/
if((ast_params.bit_select)==3){ //1bit
    for(yy=0; yy<Height; yy++){
        for(xx=0; xx<Width; xx++){
            if((InDataS_K[yy][xx]==0) InDataS_K[yy][xx]=0;
            else InDataS_K[yy][xx]=15;
        }
    }
}
else if((ast_params.bit_select)==2){ //2bit
    for(yy=0; yy<Height; yy++){
        for(xx=0; xx<Width; xx++){
            if((InDataS_K[yy][xx]==12) InDataS_K[yy][xx]=15;
            else if((InDataS_K[yy][xx]==8) InDataS_K[yy][xx]=10;
            else if((InDataS_K[yy][xx]==4) InDataS_K[yy][xx]=5;
            else InDataS_K[yy][xx]=0;
        }
    }
}
else if((ast_params.bit_select)==1){ //3bit
    for(yy=0; yy<Height; yy++){
        for(xx=0; xx<Width; xx++){
            if((InDataS_K[yy][xx]==14) InDataS_K[yy][xx]=15;
            else if((InDataS_K[yy][xx]==12) InDataS_K[yy][xx]=13;
            else if((InDataS_K[yy][xx]==10) InDataS_K[yy][xx]=11;
            else if((InDataS_K[yy][xx]==8) InDataS_K[yy][xx]=9;
            else if((InDataS_K[yy][xx]==6) InDataS_K[yy][xx]=6;
            else if((InDataS_K[yy][xx]==4) InDataS_K[yy][xx]=4;
            else if((InDataS_K[yy][xx]==2) InDataS_K[yy][xx]=2;
            else InDataS_K[yy][xx]=0;
        }
    }
}

```

---

フロントページの続き

(51)Int.Cl.	F I	
B 4 1 J 2/52 (2006.01)	B 4 1 J 3/00	A
B 4 1 J 5/30 (2006.01)	B 4 1 J 5/30	Z

審査官 加内 慎也

(56)参考文献 特開平10-336454 (JP, A)  
特開平06-296236 (JP, A)  
特開2000-125134 (JP, A)  
特開2003-032480 (JP, A)  
特開2001-298620 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H 04N 1/40 - 1 / 409  
H 04N 1/46  
H 04N 1/60