

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4143302号
(P4143302)

(45) 発行日 平成20年9月3日(2008.9.3)

(24) 登録日 平成20年6月20日(2008.6.20)

(51) Int.Cl.

H04N 1/405 (2006.01)

F 1

H04N 1/40

B

請求項の数 37 (全 45 頁)

(21) 出願番号 特願2002-6025 (P2002-6025)
 (22) 出願日 平成14年1月15日 (2002.1.15)
 (65) 公開番号 特開2003-208607 (P2003-208607A)
 (43) 公開日 平成15年7月25日 (2003.7.25)
 審査請求日 平成16年12月14日 (2004.12.14)

(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
 (74) 代理人 100090538
 弁理士 西山 恵三
 (74) 代理人 100096965
 弁理士 内尾 裕一
 (72) 発明者 山形 茂雄
 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
 (72) 発明者 谷岡 宏
 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

最終頁に続く

(54) 【発明の名称】 画像処理装置、画像処理方法、制御プログラム及び記録媒体

(57) 【特許請求の範囲】

【請求項 1】

1 ラインが X 画素からなる Y ライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理装置において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを有する演算手段と、

1 ラインの画像データを N 画素毎に分割して入力する入力手段と、

前記分割して入力された N 画素データを N ライン分記憶する記憶手段と、

前記記憶手段に記憶された N ライン分の N 画素の画像データのうち N 個の画像データを前記演算手段の N 個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理手段とを有する 画像処理装置であって、

前記 N 個の要素プロセッサーに供給され並列に処理される N 個の画像データは、それぞれ異なるラインの入力画像データであることを特徴とする画像処理装置。

【請求項 2】

前記 N 個の要素プロセッサーが並列に処理する画像処理手段は、誤差拡散処理である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 N 個の要素プロセッサーを有する演算手段は、N 個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、隣接する要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力さ

10

20

れた画像データを処理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N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理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 5】

前記N個の要素プロセッサーを有する演算手段は、1ラインを構成するX画素をN画素毎に分割したブロック数をB L Kとした場合、少なくともB L K × N個の誤差記憶手段を有する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 6】

Yラインから構成される入力画像データをNライン毎に分割したバンド数をB N Dとした場合、前記少なくともB L K × N個の誤差記憶手段に記憶された誤差データは、異なるB N Dに属する画像データを処理する際に利用されることを特徴とする請求項5に記載の画像処理装置。

【請求項 7】

1ラインがX画素からなるYライン分の入力画像データを SIMD型プロセッサーを用いて処理する画像処理方法において、

N(X > N > 1、Y > N > 1)個の並列動作可能な要素プロセッサーを用いて演算する演算工程と。 20

1ラインの画像データをN画素毎に分割して入力する入力工程と、

前記分割して入力されたN画素データをNライン分記憶する記憶工程と、

前記記憶工程によって記憶したNライン分のN画素の画像データのうちN個の画像データを前記N個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理工程とを有する画像処理方法であって、

前記N個の要素プロセッサーに供給され並列に処理されるN個の画像データは、それぞれ異なるラインの入力画像データであることを特徴とする画像処理方法。

【請求項 8】

前記N個の要素プロセッサーを用いて並列に処理する画像処理工程は、誤差拡散処理を実行することを特徴とする請求項7に記載の画像処理方法。 30

【請求項 9】

前記N個の要素プロセッサーを用いて演算を行なう演算工程は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出させる工程を有し、各要素プロセッサーに対し、隣接する要素プロセッサーが算出した誤差値に基づいて入力された画像データを処理させることを特徴とする請求項8に記載の画像処理方法。

【請求項 10】

前記N個の要素プロセッサーを用いて演算を行なう演算工程は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出させる工程を有し、各要素プロセッサーに対し、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて入力された画像データを処理させることを特徴とする請求項8に記載の画像処理方法。 40

【請求項 11】

前記N個の要素プロセッサーを用いて演算を行なう演算工程は、1ラインを構成するX画素をN画素毎に分割したブロック数をB L Kとした場合、少なくともB L K × N個の誤差を記憶する誤差記憶工程を有することを特徴とする請求項8に記載の画像処理方法。

【請求項 12】

Yラインから構成される入力画像データをNライン毎に分割したバンド数をB N Dとした場合、前記誤差記憶工程で記憶する少なくともB L K × N個の誤差データは、異なるB N Dに属する画像データを処理する際に利用されることを特徴とする請求項11に記載の画像処理方法。

【請求項 13】

請求項7乃至12のいずれか1項に記載の画像処理方法をコンピュータに実行させるための制御プログラム。

【請求項14】

請求項13に記載の制御プログラムを記録したことを特徴とする記録媒体。

【請求項15】

1ラインがX画素からなるYライン分の入力画像データを SIMD型プロセッサーを用いて処理する画像処理装置において、

N(X>N>1、Y>N>1)個の並列動作可能な要素プロセッサーを有する演算手段と、

1ラインの画像データをN画素毎に分割して入力する入力手段と、

前記分割して入力されたN画素データをN×M(Mは自然数)ライン分記憶する手段と、

前記記憶する手段にMライン分のN画素の画像データのうちN個の画像データを前記演算手段のN個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理手段とを有する画像処理装置であって、

前記N個の要素プロセッサーに供給され並列に処理されるN個の画像データは、それ異なるラインの入力画像データであることを特徴とする画像処理装置。

【請求項16】

前記N個の要素プロセッサーのそれぞれは、1ライン中の連続する複数の画素の画像データを処理することを特徴とする請求項15に記載の画像処理装置。

【請求項17】

前記入力手段は、1ラインの画像データを1ライン中の奇数画素位置のN画素分の画像データと、偶数画素位置のN画素分の画像データとに分割して入力することを特徴とする請求項15に記載の画像処理装置。

【請求項18】

前記N個の要素プロセッサーが並列に処理する画像処理手段は、誤差拡散処理であることを特徴とする請求項15に記載の画像処理装置。

【請求項19】

前記N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、隣接する要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理することを特徴とする請求項18に記載の画像処理装置。

【請求項20】

前記N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理することを特徴とする請求項18に記載の画像処理装置。

【請求項21】

前記N個の要素プロセッサーを有する演算手段は、1ラインを構成するX画素をN×M画素毎に分割したブロック数をBLKとした場合、少なくともBLK×N×M個の誤差記憶手段を有することを特徴とする請求項18に記載の画像処理装置。

【請求項22】

Yラインから構成される入力画像データをNライン毎に分割したバンドの数をBNDとした場合、前記少なくともBLK×N×M個の誤差記憶手段に記憶された誤差データは、異なるBNDに属する画像データを処理する際に利用されることを特徴とする請求項21に記載の画像処理装置。

【請求項23】

前記N個の要素プロセッサーのそれぞれは、1ライン入力画像データのライン方向に連続するM画素の画像データを処理することを特徴とする請求項15に記載の画像処理装置

10

20

30

40

50

。

【請求項 2 4】

1 ラインが X 画素からなる Y ライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理方法において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを用いて演算する演算工程と、

1 ラインの画像データを N 画素毎に分割して入力する入力工程と、

前記分割して入力された N 画素データを N × M (M は自然数) ライン分記憶する記憶工程と、

前記記憶工程によって記憶した M ライン分の N 画素の画像データのうち N 個の画像データを前記 N 個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理工程とを有することを 画像処理方法であって、
10

前記 N 個の要素プロセッサーに供給され並列に処理される N 個の画像データは、それぞれ異なるラインの入力画像データであることを特徴とする画像処理方法。

【請求項 2 5】

前記 N 個の要素プロセッサーのそれぞれは、1 ライン中の連続する複数の画素の画像データを処理することを特徴とする請求項 2 4 に記載の画像処理方法。

【請求項 2 6】

前記入力工程は、1 ラインの画像データを 1 ライン中の奇数画素位置の N 画素分の画像データと、偶数画素位置の N 画素分の画像データとに分割して入力することを特徴とする
20 請求項 2 4 に記載の画像処理方法。

【請求項 2 7】

前記 N 個の要素プロセッサーを用いて並列に処理する画像処理工程は、誤差拡散処理であることを特徴とする請求項 2 4 に記載の画像処理方法。

【請求項 2 8】

前記 N 個の要素プロセッサーを用いて演算を行なう演算工程は、N 個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出させる工程を有し、各要素プロセッサーに対し、隣接する要素プロセッサーが算出した誤差値に基づいて入力された画像データを処理させることを特徴とする請求項 2 7 に記載の画像処理方法。
30

【請求項 2 9】

前記 N 個の要素プロセッサーを用いて演算を行なう演算工程は、N 個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出させる工程を有し、各要素プロセッサーに対し、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて入力された画像データを処理させることを特徴とする請求項 2 7 に記載の画像処理方法。
。

【請求項 3 0】

前記 N 個の要素プロセッサーを用いて演算を行なう演算工程は、1 ラインを構成する X 画素を N × M 画素毎に分割したブロック数を B L K とした場合、少なくとも B L K × N × M 個の誤差を記憶する誤差記憶工程を有することを特徴とする請求項 2 7 に記載の画像処理方法。
40

【請求項 3 1】

Y ラインから構成される入力画像データを N ライン毎に分割したバンドの数を B N D とした場合、前記誤差記憶工程で記憶する少なくとも B L K × N × M 個の誤差データは、異なる B N D に属する画像データを処理する際に利用されることを特徴とする請求項 3 0 に記載の画像処理方法。

【請求項 3 2】

請求項 2 4 乃至 3 1 のいずれか 1 項に記載の画像処理方法をコンピュータに実行させるための制御プログラム。

【請求項 3 3】

請求項 3 2 に記載の制御プログラムを記録したことを特徴とする記録媒体。
50

【請求項 3 4】

1 ラインが X 画素からなる Y ライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理装置において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを有する演算手段と、

1 ラインの画像データを N 画素毎に分割して入力する入力手段と、

前記分割して入力された N 画素データを N ライン分記憶する記憶手段と、

前記記憶手段に記憶された N ライン分の N 画素の画像データのうち異なるラインの画像データを含む N 個の画像データを前記演算手段の N 個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理手段とを有することを特徴とする画像処理装置。 10

【請求項 3 5】

1 ラインが X 画素からなる Y ライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理方法において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを用いて演算する演算工程と、

1 ラインの画像データを N 画素毎に分割して入力する入力工程と、

前記分割して入力された N 画素データを N ライン分記憶する記憶工程と、

前記記憶工程によって記憶した N ライン分の N 画素の画像データのうち異なるラインの画像データを含む N 個の画像データを前記 N 個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理工程とを有することを特徴とする画像処理方法。 20

【請求項 3 6】

1 ラインが X 画素からなる Y ライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理装置において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを有する演算手段と、

1 ラインの画像データを N 画素毎に分割して入力する入力手段と、

前記分割して入力された N 画素データを N × M (M は自然数) ライン分記憶する手段と、 30

前記記憶する手段に M ライン分の N 画素の画像データのうち異なるラインの画像データを含む N 個の画像データを前記演算手段の N 個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理手段とを有することを特徴とする画像処理装置。

【請求項 3 7】

1 ラインが X 画素からなる Y ライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理方法において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを用いて演算する演算工程と、

1 ラインの画像データを N 画素毎に分割して入力する入力工程と、 40

前記分割して入力された N 画素データを N × M (M は自然数) ライン分記憶する記憶工程と、

前記記憶工程によって記憶した M ライン分の N 画素の画像データのうち異なるラインの画像データを含む N 個の画像データを前記 N 個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理工程とを有することを特徴とする画像処理方法。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は、デジタル信号で表される画像データを SIMD 型のプロセッサーに基づいて 50

処理する画像処理装置、画像処理方法、制御プログラム及び記録媒体に関する。

【0002】

【従来の技術】

現在、コピー機、ファクシミリ、プリンター、スキャナーといった画像処理装置の複合機として構成された、いわゆるMFP (Multi Function Printer) と呼ばれる画像処理装置がある。このようなMFPの画像処理部に、SIMD (Single Instruction stream Multiple Data stream) 型のプロセッサーを使用することにより、高速、かつプログラマブルに画像を処理する技術が例えば特開平8-315126号公報に記載されている。

【0003】

また、従来、MFPでは、スキャナーにて読み取られた多値の原稿画像データを2値プリンタに出力する際には、原稿画像データの中間調を擬似的に表現するための、誤差拡散処理を行なうものがある。誤差拡散処理は、注目画素を2値化する際に発生した2値化誤差に基づいて、後に2値化される注目画素の多値データの補正が行なわれる逐次処理である。

【0004】

このような誤差拡散処理を SIMD 型の演算処理部と、逐次的な処理を補助演算部を用いて行なう技術が特開2001-274993号公報に記載されている。

【0005】

【発明が解決しようとする課題】

しかしながら上記従来例では、誤差拡散に代表される逐次処理を行なうために、SIMD型演算処理部の他に補助演算部を必要とするため、回路規模の増大、コストアップの要因となっていた。さらには、SIMD型演算処理部と、前記補助演算部とのインターフェース回路ブロックの増加、インターフェース制御等も必要となり、非効率的な構成となっていた。

【0006】

本発明は上述した課題に鑑みてなされたものであり、その目的は、逐次処理を行なうための補助的な演算処理部を利用することなく SIMD 型演算処理部を用いて、例えば誤差拡散処理等の画像処理を行なうことが可能な画像処理装置、画像処理方法、制御プログラム及び記録媒体を提供することにあり、さらには、ラスタースキャンによる1ラインの入力画像データの画素数が、SIMD型プロセッサーが有する演算処理部の個数以上の場合であっても、SIMD型プロセッサーが有する複数の演算処理部を効率よく動作させながら逐次処理が可能な画像処理装置、画像処理方法、制御プログラム及び記録媒体を提供することにある。

【0007】

【課題を解決するための手段】

上述した課題を解決し、目的を達成するために、本発明に係わる画像処理装置は、1ラインがX画素からなるYライン分の入力画像データを SIMD 型プロセッサーを用いて処理する画像処理装置において、

N (X > N > 1 , Y > N > 1) 個の並列動作可能な要素プロセッサーを有する演算手段と、

1ラインの画像データをN画素毎に分割して入力する入力手段と、

前記分割して入力されたN画素データをNライン分記憶する記憶手段と、

前記記憶手段に記憶されたNライン分のN画素の画像データのうちN個の画像データを前記演算手段のN個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理手段とを有する画像処理装置であって、

前記N個の要素プロセッサーに供給され並列に処理されるN個の画像データは、それぞれ異なるラインの入力画像データであることを特徴とする。

【0009】

また、この発明に係わる画像処理装置において、前記N個の要素プロセッサーが並列に処理する画像処理手段は、誤差拡散処理であることを特徴としている。

10

20

30

40

50

【 0 0 1 0 】

また、この発明に係わる画像処理装置において、前記N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、隣接する要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理することを特徴としている。

【 0 0 1 1 】

また、この発明に係わる画像処理装置において、前記N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理することを特徴としている。10

【 0 0 1 2 】

また、この発明に係わる画像処理装置において、前記N個の要素プロセッサーを有する演算手段は、1ラインを構成するX画素をN画素毎に分割したブロック数をB L Kとした場合、少なくともB L K × N個の誤差記憶手段を有することを特徴としている。

【 0 0 1 3 】

また、この発明に係わる画像処理装置において、Yラインから構成される入力画像データをNライン毎に分割したバンド数をB N Dとした場合、前記少なくともB L K × N個の誤差記憶手段に記憶された誤差データは、異なるB N Dに属する画像データを処理する際に利用されることを特徴としている。20

【 0 0 1 4 】

また、本発明に係わる画像処理方法は、1ラインがX画素からなるYライン分の入力画像データをS I M D型プロセッサーを用いて処理する画像処理方法において、

N (X > N > 1, Y > N > 1) 個の並列動作可能な要素プロセッサーを用いて演算する演算工程と、

1ラインの画像データをN画素毎に分割して入力する入力工程と、

前記分割して入力されたN画素データをNライン分記憶する記憶工程と、

前記記憶工程によって記憶したNライン分のN画素の画像データのうちN個の画像データを前記N個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理工程とを有する画像処理方法であって、30

前記N個の要素プロセッサーに供給され並列に処理されるN個の画像データは、それぞれ異なるラインの入力画像データであることを特徴とする。

【 0 0 1 6 】

また、この発明に係わる画像処理方法において、前記N個の要素プロセッサーを用いて並列に処理する画像処理工程は、誤差拡散処理を実行することを特徴としている。

【 0 0 1 7 】

また、この発明に係わる画像処理方法において、前記N個の要素プロセッサーを用いて演算を行なう演算工程は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出させる工程を有し、各要素プロセッサーに対し、隣接する要素プロセッサーが算出した誤差値に基づいて入力された画像データを処理させることを特徴としている。40

【 0 0 1 8 】

また、この発明に係わる画像処理方法において、前記N個の要素プロセッサーを用いて演算を行なう演算工程は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出させる工程を有し、各要素プロセッサーに対し、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて入力された画像データを処理させることを特徴としている。

【 0 0 1 9 】

また、この発明に係わる画像処理方法において、前記N個の要素プロセッサーを用いて演50

算を行なう演算工程は、1ラインを構成するX画素をN画素毎に分割したブロック数をB L Kとした場合、少なくともB L K × N個の誤差を記憶する誤差記憶工程を有することを特徴としている。

【0020】

また、この発明に係わる画像処理方法において、Yラインから構成される入力画像データをNライン毎に分割したバンド数をB N Dとした場合、前記誤差記憶工程で記憶する少なくともB L K × N個の誤差データは、異なるB N Dに属する画像データを処理する際に利用されることを特徴としている。

【0021】

また、本発明に係わる制御プログラムは上記の画像処理方法をコンピュータに実行させることを特徴としている。 10

【0022】

また、本発明に係わる記録媒体は、上記の制御プログラムを記録したことを特徴としている。

【0023】

また、本発明に係わる画像処理装置は、1ラインがX画素からなるYライン分の入力画像データをS I M D型プロセッサーを用いて処理する画像処理装置において、

N(X > N > 1, Y > N > 1)個の並列動作可能な要素プロセッサーを有する演算手段と、

1ラインの画像データをN画素毎に分割して入力する入力手段と、 20

前記分割して入力されたN画素データをN × M(Mは自然数)ライン分記憶する手段と、

前記記憶する手段にMライン分のN画素の画像データのうちN個の画像データを前記演算手段のN個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理手段とを有する画像処理装置であって、

前記N個の要素プロセッサーに供給され並列に処理されるN個の画像データは、それぞれ異なるラインの入力画像データであることを特徴とする。

【0025】

また、この本発明に係わる画像処理装置において、前記N個の要素プロセッサーのそれぞれは、1ライン中の連続する複数の画素の画像データを処理することを特徴としている。 30

【0026】

また、この本発明に係わる画像処理装置において、前記入力手段は、1ラインの画像データを1ライン中の奇数画素位置のN画素分の画像データと、偶数画素位置のN画素分の画像データとに分割して入力することを特徴としている。

【0027】

また、この本発明に係わる画像処理装置において、前記N個の要素プロセッサーが並列に処理する画像処理手段は、誤差拡散処理であることを特徴としている。

【0028】

また、この本発明に係わる画像処理装置において、前記N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、隣接する要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理することを特徴としている。 40

【0029】

また、この本発明に係わる画像処理装置において、前記N個の要素プロセッサーを有する演算手段は、N個の要素プロセッサー毎に誤差拡散処理にて発生した量子化誤差を誤差値として算出する手段を有し、各要素プロセッサーは、少なくとも直前に該要素プロセッサーが算出した誤差値に基づいて各要素プロセッサーに入力された画像データを処理することを特徴としている。

【0030】

10

20

30

40

50

また、この本発明に係わる画像処理装置において、前記N個の要素プロセッサーを有する演算手段は、1ラインを構成するX画素をN×M画素毎に分割したブロック数をBLKとした場合、少なくともBLK×N×M個の誤差記憶手段を有することを特徴としている。

【0031】

また、この本発明に係わる画像処理装置において、Yラインから構成される入力画像データをNライン毎に分割したバンドの数をBNDとした場合、前記少なくともBLK×N×M個の誤差記憶手段に記憶された誤差データは、異なるBNDに属する画像データを処理する際に利用されることを特徴としている。

【0032】

また、この本発明に係わる画像処理装置において、前記N個の要素プロセッサーのそれぞれ、1ライン入力画像データのライン方向に連続するM画素の画像データを処理することを特徴としている。 10

【0033】

また、本発明の画像処理方法は、1ラインがX画素からなるYライン分の入力画像データをSIMD型プロセッサーを用いて処理する画像処理方法において、

N(X>N>1、Y>N>1)個の並列動作可能な要素プロセッサーを用いて演算する演算工程と、

1ラインの画像データをN画素毎に分割して入力する入力工程と、

前記分割して入力されたN画素データをN×M(Mは自然数)ライン分記憶する記憶工程と、 20

前記記憶工程によって記憶したMライン分のN画素の画像データのうちN個の画像データを前記N個の要素プロセッサーにそれぞれ供給し、各要素プロセッサーに同種の演算を並列に実行させる画像処理工程とを有することを画像処理方法であって、

前記N個の要素プロセッサーに供給され並列に処理されるN個の画像データは、それ異なるラインの入力画像データであることを特徴とする。

【0034】

また、本発明に係わる制御プログラムは上記の画像処理方法をコンピュータに実行させることを特徴としている。

【0035】

また、本発明に係わる記録媒体は、上記の制御プログラムを記録したことを特徴としている。 30

【0036】

【発明の実施の形態】

(画像処理装置の全体説明)

まず、本実施の形態にかかる画像処理装置の原理について説明する。図1は、この発明の本実施の形態にかかる画像処理装置の構成を機能的に示すブロック図である。図1において、画像処理装置は、以下に示す5つのユニットを含む構成である。

【0037】

上記5つのユニットとは、画像データ制御ユニット100と、画像データを入力する画像データ入力ユニット101と、画像を蓄積する画像メモリーを制御して画像データの書き込み/読み出しをおこなう画像メモリー制御ユニット102と、画像データに対し加工編集等の画像処理を施す画像処理ユニット103と、画像データを転写紙等に書き込む画像書込ユニット104である。 40

【0038】

上記各ユニットは、画像データ制御ユニット100を中心に構成されている。すなわち、画像データ入力ユニット101、画像メモリー制御ユニット102、画像処理ユニット103、画像書込ユニット104は、いずれも画像データ制御ユニット100に接続されている。以下、この各ユニットについて、それぞれ説明する。

【0039】

画像データ制御ユニット100によりおこなわれる処理は以下のようである。 50

【0040】

制御データバスインターフェース処理、全体システム制御、ローカルバス制御処理（システム・コントローラを起動させるためのROM、RAM、アクセス制御処理）、画像データ入力ユニット101とのインターフェース処理、画像メモリ制御ユニット102とのインターフェース処理、画像処理ユニット103とのインターフェース処理、画像書き込みユニット104とのインターフェース処理、ネットワーク制御処理等である。

【0041】

画像データ入力ユニット101によりおこなわれる処理は以下である。

【0042】

システム・コントローラとのインターフェース制御処理、光学系による原稿反射光の読み取り処理、CCD（Charge Coupled Device：電荷結合素子）等を用いた電気信号への変換処理、A/D変換器でのデジタル化処理、シェーディング補正処理（光源の照度分布ムラを補正する処理）、読み取り系の濃度特性を補正する処理、ネットワークを介して入力されるPDL画像データのラスタライズ処理等である。 10

【0043】

画像メモリー制御ユニット102によりおこなわれる処理は以下である。

【0044】

システム・コントローラとのインターフェース制御処理、メモリー部への書き込み・読み出し処理、メモリー・モジュールへのアクセス制御処理（複数のユニットからのメモリー・アクセス要求の調停処理）等である。 20

【0045】

画像処理ユニット103によりおこなわれる処理は以下である。

【0046】

色変換処理、色補正処理、MTF補正処理、平滑化処理、主走査方向の任意変倍処理、濃度変換（変換処理：濃度調整キーに対応）、単純二値化処理、各種擬似中間調処理、ドット配置位相制御処理（ジャギー補正）、像域分離処理（色判定、属性判定、適応処理）、密度変換処理、等である。

【0047】

画像書き込みユニット104によりおこなわれる処理は以下である。

【0048】

画像信号のパルス制御処理、パラレルデータとシリアルデータのフォーマット変換処理、等である。 30

【0049】

（ディジタル複合機のハードウェア構成）

次に、本実施の形態にかかる画像処理装置がディジタル複合機を構成する場合のハードウェア構成について説明する。図2は本実施の形態にかかる画像処理装置のハードウェア構成の一例を示すブロック図である。

【0050】

図2のブロック図において、本実施の形態にかかる画像処理装置は、読み取りユニット201と、PDL処理ユニット202と、画像データ制御部203と、画像処理プロセッサー204と、作像ユニット205、メモリ制御部206、メモリモジュール207、ネットワーク制御部214、ワーキングメモリ216とを備える。また、本実施の形態にかかる画像処理装置は、制御用データバス208を介して、システムコントローラ209と、ROM210と、RAM211、操作パネル212とを備える。さらに、ネットワーク213を介して、パーソナルコンピュータ215に接続されている。 40

【0051】

上記した構成のうち、画像処理プロセッサー204は、画像に基づいて作成されたディジタル信号である画像データを顕像として出力できるように処理し、複数の画像形成動作を実現できるプログラマブルな画像処理手段である。また、画像データ制御部203は、画像データを伝送するデータバスと画像処理プロセッサー204による画像処理に用いられ 50

る処理ユニット間の画像データ伝送を一括して管理する画像データ伝送管理手段であり、読み取りユニット201、PDL処理ユニット202、画像処理プロセッサー204、メモリ制御部206、作像ユニット205、ネットワーク制御部214間のデータ伝送管理を行なう。なお、本発明は、主に画像処理プロセッサー204に関するもので、画像処理プロセッサー204の構成については、図3以降の図面を用いて詳細に説明するものとする。

【0052】

また、本実施の形態にかかる画像処理装置は、画像データ記憶管理手段として、画像メモリー制御部206に接続されるメモリー・モジュール207を備える。

【0053】

ここで、上記各構成部と、図1に示した各ユニット100～104との関係について説明する。すなわち、読み取りユニット201およびPDL処理ユニット202により、図1に示した画像データ入力ユニット101の機能を実現する。また同様に、画像データ制御部203、システムコントローラ209、ROM210、RAM211、操作パネル212、ネットワーク制御部214により、画像データ制御ユニット100の機能を実現する。また同様に、画像処理プロセッサー204、ワーキングメモリ216により画像処理ユニット103の機能を実現する。

10

【0054】

また同様に作像ユニット205により画像書込ユニット104を実現する。また同様に、メモリー制御部206およびメモリー・モジュール207により画像メモリー制御ユニット102を実現する。

20

【0055】

尚、システムコントローラ209は、制御用データバス208を介して接続されたROM210に記憶された制御プログラムに基づき動作するものであり、RAM211をワーク用メモリとして使用している。また、読み取りユニット201、PDL処理ユニット202、画像データ制御部203、画像処理プロセッサー部204、作像ユニット205、メモリ制御部206、ネットワーク制御部214、操作パネル212は、制御用データバス208を介して、システムコントローラ209に動作が制御されるものである。

【0056】

各構成部の内容について説明する。原稿を光学的に読み取る読み取りユニット201は、ランプとミラーとレンズ、受光素子から構成され、原稿に対するランプ照射の反射光をミラーおよびレンズにより受光素子に集光する。

30

【0057】

受光素子、たとえばCCDにおいて電気信号に変換された画像データはデジタル信号に変換された後、読み取りユニット201より出力（送信）される。

【0058】

PDL処理ユニット202は、ネットワーク213に接続されたパーソナルコンピュータ215より出力されたPDL画像データをビットマップ画像へラスタライズするユニットである。ネットワーク213を介して入力されたPDL画像データがネットワーク制御部214を介して、PDL処理ユニット202に入力されると、PDL処理ユニット202は、入力されたPDL画像データに基づいたラスタライズを行ない、ビットマップ画像データを出力（送信）する。以上のように、読み取りユニット201、PDL処理ユニット202より出力（送信）された画像データは画像データ制御部203に入力（受信）される。

40

【0059】

読み取りユニット201、PDL処理ユニット202より画像データ制御部203が受信した画像データは、画像データ処理プロセッサー部204、または、メモリ制御部206に出力される。

【0060】

まず、画像処理プロセッサー部204に出力される場合の動作について説明する。

【0061】

50

画像処理プロセッサ部 204 に入力された画像データは、ワーキングメモリー 216 を用いながら画像処理プロセッサー部 204 にて処理が行なわれた後、再度画像データ制御部 203 に出力される。画像処理プロセッサー部 204 より、画像データ処理部 203 に入力された画像データは、メモリ制御部 206 に出力され、メモリ制御部 206 を介して、メモリモジュール 207 に記憶される。

【0062】

画像処理プロセッサー部 204 による 1 画面分の画像データの処理が終了し、1 画面分の処理済みデータが、メモリモジュールに記憶された後、メモリー制御部 206 は、メモリー・モジュール 207 に対する画像データの読み出しを行ない読み出された画像データを画像データ制御部 203 を介して、作像ユニット 205 に出力し、プリント出力を得る。あるいは、メモリモジュール 207 より読み出された画像データを画像データ制御部 203 より、ネットワーク制御部 214 に出力し、ネットワーク 213 を介して、パーソナルコンピュータ 215 に出力するように動作する。

10

【0063】

次に、読み取りユニット 201、PDL 処理ユニット 202 より画像データ制御部 203 が受信した画像データをメモリ制御部 206 に出力する場合の動作について説明する。画像データ制御部 203 より、メモリ制御部 206 に入力された画像データは、メモリモジュール 207 に記憶される。次に、メモリ制御部 206 は、メモリモジュール 207 より、記憶された画像データを読み出し、画像データ制御部 203 を介して、画像処理プロセッサー部 204 に出力する。画像処理プロセッサー部 204 では、入力された画像データを処理し、処理後の画像データを、再度画像データ制御部 203、メモリ制御部 206 を介して、メモリモジュール 207 に記憶する。画像処理プロセッサー部 204 による 1 画面分の画像データの処理が終了し、1 画面分の処理済みデータが、メモリモジュール 207 に記憶された後、メモリー制御部 206 は、メモリー・モジュール 207 に対する画像データの読み出しを行ない読み出された画像データを画像データ制御部 203 を介して、作像ユニット 205 に出力し、プリント出力を得る。あるいは、メモリモジュール 207 より読み出された画像データを画像データ制御部 203 より、ネットワーク制御部 214 に出力し、ネットワーク 213 を介して、パーソナルコンピュータ 215 に出力するように動作する。

20

【0064】

30

上記動作例は、読み取りユニット 201、PDL 処理ユニット 202 より出力された画像データに対する処理を画像処理プロセッサー部 204 により行ない、1 画面分の処理済み画像データがメモリモジュール 207 に記憶された後、メモリモジュール 207 より、処理済画像データの読み出しを行ない、作像ユニット 205、あるいは、ネットワーク制御部 214 に出力する例を示したが、処理済み画像データの記憶が 1 画面分終了する前に、処理済みの画像データをメモリモジュール 207 より読み出しを開始するように制御してもよい。

【0065】

さらには、メモリモジュール 207 に画像データを記憶させない動作例について説明する。

40

【0066】

読み取りユニット 201、PDL 処理ユニット 202 より画像データ制御部 203 が受信した画像データは、画像データ制御部 203 より、画像処理プロセッサー部 204 に出力される。画像処理プロセッサー部 204 では、入力された画像データに所定の処理を行ない、画像データ制御部 203 に出力する。画像処理プロセッサー部 204 より、画像データ制御部 203 に入力された画像データは、画像データ制御部 203 を介して、作像ユニット 205、ネットワーク制御部 214 に出力される。

【0067】

メモリー・モジュール 207 に処理済みの画像データ 1 面分を記憶する場合の動作例としては、1 枚の原稿について複数枚を複写する場合に、読み取りユニット 201 を 1 回だけ動作

50

させ、読み取りユニット 201 により読み取った画像データをメモリー・モジュール 207 に記憶し、記憶された画像データを複数回読み出すという方法がある。

【0068】

メモリー・モジュール 207 に画像データを記憶させない動作例としては、1枚の原稿を1枚だけ複写する場合がある。読み取り画像データに対する処理済みデータを直接作像ユニット 205 に出力すればよいので、メモリー・モジュール 207 へのアクセスをおこなう必要はない。

【0069】

尚、本装置の全体の動作は、操作パネル 212 より入力された、画像処理装置がおこなうべき処理に基づいてシステムコントローラ 209 により制御される。操作パネル 212 からは、処理の種類（複写、送信、画像読み込み、プリント等）および処理の枚数等を入力することができる。

【0070】

図 3 は、図 2 に示した本実施の形態の画像処理プロセッサー部 204 の構成を説明するための図である。図示した画像処理プロセッサー部 204 は、FIFO メモリ 301、307 と、演算処理ユニット 300 から構成され、演算処理ユニット 300 は、入力レジスタ 302、出力レジスタ 304、SIMD 型のデータ演算処理部 303 からなる SIMD プロセッサー 308 と、制御プロセッサー部 305、外部メモリインターフェース 306 を有している。なお、SIMD とは、複数のデータに対し、単一の命令を並列に実行させるもので、本実施の形態では、128 個の PE によりデータ演算処理部 303 が構成されている。FIFO メモリ 301 は、読み取りユニット 201、または、PDL 処理ユニット 202 より入力される画像データ 1 ライン分(7168 画素)の容量を有するファーストイン、ファーストアウトのメモリであり、書き込みと読み出しが独立に制御される。画像データ制御部 203 のデータバス A より入力された画像データは、FIFO メモリ 301 に入力され、データ演算処理部 303 が有する PE の数と等しいレジスタ数で構成された入力レジスタ 302 に 128 個分の画像データとして 56 分割して入力される。FIFO メモリ 301 より、入力レジスタ 302 に入力された画像データは、データ演算処理部 303、外部メモリインターフェース 306 に出力される。データ演算処理部 303 に入力された画像データは、データ演算処理部 303 にて所定の処理が行なわれ、処理後の画像データが、出力レジスタ 304、外部メモリインターフェース 306 に出力される。尚、外部メモリインターフェース 306 には、データ演算処理部 303 で処理された中間データを出力することも、可能な構成となっている。出力レジスタ 304 は、入力レジスタ 302 と同様に、データ演算処理部 303 が有する PE 数と等しいレジスタ数で構成されている。出力レジスタ 304 の出力画像データは、画像データ 1 ライン分の容量を有する FIFO メモリ 307 に入力される。FIFO メモリ 307 は、書き込みと読み出しが独立に制御されるファーストイン、ファーストアウトのメモリである。FIFO メモリ 307 より出力される画像データ信号は、データバス B を介して画像データ制御部 203 に出力される。さらに、SIMD プロセッサー 308 及び外部メモリインターフェイス 306 は、図 2 に示した制御用データバス 208 に接続された制御プロセッサー部 305 と接続されている。制御プロセッサー部 305 は、データ演算処理部 303 の PE に対する命令の供給、各 PE のステータスの判断、各 PE に接続されたメモリ、レジスタへのデータの入出力等の制御及び外部メモリインターフェイス部 306 を制御し、SIMD プロセッサー 308 の内部メモリーやレジスターとワーキングメモリー 216 間のデータ制御を行なうものである。尚、制御プロセッサー 305 と SIMD プロセッサー 308 は互いに異なる処理を独立に実行することが可能である。

【0071】

次に、SIMD 型プロセッサーを構成するデータ演算処理部 303 を含む演算処理ユニット 300 の概略ブロックを図 4 に示し説明する。

【0072】

図 4 中、305 は、図 3 に示した制御プロセッサー部を示し、制御プロセッサー 401 お

10

20

30

40

50

および、制御プロセッサ及び SIMD プロセッサーの動作を制御するプログラムが格納されたプログラムメモリ 402、データメモリ 403 より構成される。404 は、SIMD プロセッサー 308 を構成するひとつの PE を示し、前述したように、本実施の形態における SIMD プロセッサー 308 は 128 個の PE (PE0 ~ PE127) により構成されている。

【0073】

図 4 に示したように、それぞれの PE は、8 ビットの演算ユニット (ALU) 405 と、8 ビットのレジスタ 16 本から構成される汎用レジスタ 406、ALU の演算動作を実施するか否かを制御するマスクレジスタ 407、演算途中のデータを格納する PE レジスタ 408、出力レジスタ 409、入力レジスタ 408、2K バイトの容量を有するメモリ 411 から構成される。ALU 405、PE レジスタ 408 は、隣接する PE 間における同一構成要素との接続がなされており、データの入出力が可能な構成となっている。また、出力レジスタ 409、入力レジスタ 410 も隣接する PE 間の同一構成要素との接続が行なわれており、128 段のシフトレジスタとして動作する。尚、図 3 中の入力レジスタ 302、出力レジスタ 304 に相当するブロックには同一符号を付して図 4 に示してある。

【0074】

また、各 PE におけるメモリ 411 は、外部メモリインターフェースを介して、データバス C より、ワーキングメモリー 216 に接続される。

【0075】

尚、同一 PE を構成する ALU 405、汎用レジスタ 406、マスクレジスタ 407、PE レジスタ 408、出力レジスタ 409、入力レジスタ 410、メモリ 413 は、任意のブロック間におけるデータの入出力が可能な構成となっており、例えば、メモリ 411 から PE レジスタ 408 へのデータ入出力、PE レジスタ 408 から外部メモリインターフェース 306 へのデータ入出力が可能な構成となっている。

【0076】

各 PE に対する命令の供給は、制御プロセッサー 401 より、命令供給バス 413 を介して各 PE に同一内容で与えられ、全ての PE が同一の命令に従った動作を行なうように制御されるが、各 PE に与える処理対象のデータを異ならせることにより、各 PE が、異なる処理対象データに対する演算処理を並列に行なうように制御される。たとえば、画像データ 1 ライン中の 128 画素の内容を各画素ごとに PE レジスタに配置し、同一の命令コードで PE レジスタに対する演算処理をさせれば、1 画素ずつ逐次処理するよりも短時間で 128 画素分の処理結果が得られる。

【0077】

各 PE の ALU 405 における演算結果、および、PE レジスタ 408 の内容は、隣接する PE 間で入出力可能な構成となっていることにより、隣接 PE の PE レジスタ 408、および、ALU 405 の演算結果を参照した演算処理が各 PE で行なうことも可能な構成となっている。さらに、各 PE のメモリ 411 および、入力レジスタ 410、出力レジスタ 409、PE レジスタ 408、マスクレジスタ 407、汎用レジスタ 406 は、メモリ / レジスタアクセスバス 414 を介して制御プロセッサ 401 に接続され、メモリ、および、各レジスタデータの入出力が、制御プロセッサ 401 により制御される。

【0078】

また、制御プロセッサ 401 は、制御用データバス 208 を介して、図 2 中のシステムコントローラ 209 と制御データの入出力が可能な構成となっている。さらに、制御プロセッサ 401 の動作を制御するプログラムメモリ 402、および、データメモリ 403 は、制御用データバス 208 を介してシステムコントローラ 209 よりアクセス可能な構成となっており、システムコントローラ 209 により、画像処理ユニット 300 で行なう処理内容に応じて、制御プロセッサ 401 の動作を制御するプログラムメモリ 402 の書き換えが可能な構成となっている。

【0079】

(第 1 の実施の形態)

10

20

30

40

50

本発明を誤差拡散処理に適用した第1の実施の形態を以下に詳細に説明する。

【0080】

図5は、図2中の読み取りユニット201より出力されるA4原稿画像データ配置を示す。図中の主走査方向が、読み取りユニット201に設けられたCCDの画素並び方向を表しこれをライン方向と呼ぶ。また、主走査(ライン)方向の読み取り領域を副走査方向に順次走査することにより、2次元的な原稿画像データの読み取り動作が行なわれる。なお、本実施の形態では、読み取りユニット201の読み取解像度は600dpiと設定されており、A4原稿画像は、主走査方向には、7168画素、副走査方向には、4992画素の2次元データとなる。上記は、読み取りユニット201より出力される原稿画像データの例を示したが、PDL処理ユニット202より出力される画像データが、A4サイズのプリントデータに相当する場合には、図5に示した画像データと同一配置となる。以下では、読み取りユニット201から出力された画像データを処理する例を説明するが、PDL処理ユニット202からの出力画像データを処理する場合であっても同様である。
10

【0081】

読み取りユニット201から出力されたA4原稿画像データは、画像データ制御部203に入力され、図3に示した、画像処理プロセッサー部204の FIFOメモリ301に入力される。図6に FIFOメモリ301の動作の概要を示す。

【0082】

FIFOメモリ301は、1ライン分(7168画素)の容量を有するラインメモリであり、読み取りユニット201が出力する原稿画像データに同期したクロックにより、FIFOメモリ301に入力画像データをライトする。リードクロックは、画像処理ユニット300より出力されるクロックであり、FIFOメモリに128画素分の画像データがライトされた後の、リード開始(0)のタイミングより、128画素分の画像データのリード動作が開始される。なお、本実施の形態では、FIFOメモリ301へのライトクロックが20MHz、リードクロックは、100MHzと設定されており、ライトクロックに対してリードクロックが高速なものとなっている。FIFOメモリ301より読み出された画像データは、FIFO301のリードクロックと同期したシフトクロックにより図3中の入力レジスタ301に入力される。入力レジスタ301に128画素分の画像データが入力されると、入力レジスタに入力された128画素分の画像データは、図3中の外部メモリインターフェース306を介してワーキングメモリ216に記憶される。なお、この間も連続的にFIFO301には入力画像データのライト動作が行なわれている。さらに、FIFOメモリに256画素分の画像データがライトされた後のリード開始(1)のタイミングより、128画素分の画像データのリード動作が開始され、上述したのと同様の動作にて、128~256画素目の画像データが、ワーキングメモリ216に記憶される。
20
30

【0083】

以上の動作を繰り返し、1ライン7168画素の画像データは、128画素づつ、56回に分けてワーキングメモリ216に記憶される。さらに、連続するラインの画像データも同様にして、ワーキングメモリ216に、128画素単位で記憶される。なお、FIFOメモリ301のライトクロックに対するリードクロックの速度は、FIFOメモリに入力される画像データを取りこぼしなく、128画素単位で、入力レジスタに入力し、ワーキングメモリ216に転送することが可能な速度であれば、上記速度に限定されるものではない。
40

【0084】

図7は、ワーキングメモリ216に記憶された画像データとワーキングメモリのアドレスとの関係を模式的に示した図である。ワーキングメモリのアドレスは、連続する128画素分のデータを1ラインとするデータブロック毎に0から割り振られているものであり、ワーキングメモリのラインアドレス0~55に記憶された画像データが、入力された画像データの先頭1ラインに相当するものである。なお、入力レジスタ302よりワーキングメモリ216へのデータ転送の起動および、ワーキングメモリのアドレス制御は、図3中の制御プロセッサー305により動作が制御されるものである。
50

【 0 0 8 5 】

次に、図3中のデータ演算処理部303における動作を図8のフローを用いて説明する。Step_1にて、128ライン分の入力画像データがワーキングメモリ216に取り込まれるまで待機する。ワーキングメモリ216に128ライン分の入力画像データ取り込みが終了するとフローは、Step_2に進む。なお、入力画像のワーキングメモリへの取り込みは、以下のフローを実施している間もA4サイズの原稿画像に相当する4992ライン分が取り込まれるまで継続して行なわれている。Step_2にて変数の初期化が行なわれる。

ここで、変数に関して説明する。演算ユニット300にて用いられる変数としては、データ演算処理部の各PEが並列に持つ変数と、並列に持たない変数がある。各PEが並列に持つ変数とは、各PEそれぞれに割り当てられた変数でありこれをsep型変数と呼ぶ。sep型変数は各PE毎に異なる値として使用することが可能となっており、sep型で宣言された変数は、ひとつの変数につき本実施の形態のデータ演算処理部303を構成するPEの個数と等しい128個が設けられることになる。各PEが並列に持たない変数を用いた演算を各PEが並列に行なう場合には、各PEが共通の変数值を参照するのに対して、各PEが並列に持つ変数を用いた演算を各PEが並列に行なう場合には、変数名が同一だが、各PEがそれぞれ異なる変数值を参照することが可能となる。

【 0 0 8 6 】

Step_2では、変数N(ワーキングメモリのラインアドレス)、L(データ転送回数)、BND(バンド)、BLK(ブロック)の初期化、および、sep型変数i(つなぎ誤差のインデックス)、sep型変数ERRRB[0]～ERRRB[55](つなぎ誤差データ56個)の初期化を行なう。ここで、BND(バンド)とは、入力画像データの副走査方向の128ラインを1バンドとする変数、BLK(ブロック)とは、入力画像データの主走査方向の128画素を1ブロックとする変数であり、A4サイズの原稿画像データ；主走査画素数=7168、副走査ライン数=4992に対しては、BNDは、0～38、BLKおよびiは、0～55の値をとる。また、sep型変数ERRRB[0]～ERRRB[55]は、2値化処理により発生した2値化誤差の副走査方向への分配成分を記憶するメモリであり、各PEそれぞれにつき56個が設けられている。56個とは上述した、主走査方向に配置されるBLK数と等しいものとなっている。2値化誤差の副走査方向への分配は、副走査方向に連続する同一BLK間にて各PEが行なうものである。

【 0 0 8 7 】

次にStep_3にて、ワーキングメモリ216のラインアドレスNの128画素分のデータをデータ演算処理部303の各PEが有するメモリ411に転送する。L%128は、Lを128で割り算した余りの数値を示し、Lを128で割り算した余りがメモリ411の転送先アドレスとなる。Lを128で割り算した余りとは、Lの下位7ビットの値と等価である。

【 0 0 8 8 】

図9の(a)は、L=0の時に、ワーキングメモリ216よりメモリ411に転送された画像データの配置を示す。この場合には、最初の主走査128画素分の画像データが、各PEが有するメモリ411のアドレス0に、1画素づつ記憶される。

【 0 0 8 9 】

次にsep型変数Kに、(L_PENUM)を128で割り算した余りを代入する。この演算は各PEにて並列実行されるものである。PENUMは、プロセッサエレメントPE0に対して0、PE1に対して1、……、PE127に対して127が割り当てられたシステム定数であり、各PEに対応するsep型変数Kの値は、Lの値に応じて図10のように算出される。

【 0 0 9 0 】

次にフローはstep_4に進み、各PEにおいて並列に、PENUM=0、かつBLK=0か否かが判定され、PENUM=0、かつBLK=0の場合には、step_5に、そうでない場合には、step_6に分岐する。PENUM=0は、プロセッサエレメントPE0に割り当てられたシステム定数であり、PE0以外のプロセッサエレメントのPENUMが、

10

20

30

40

50

0' と等しくなることはなく、P E 1 ~ P E 1 2 7 のプロセッサエレメントにおいては常にStep_6 が実行されることになる。

【 0 0 9 1 】

Step_5 では、プロセッサエレメント P E 0 に対応したsep型変数 E R h に 0 が代入される。Step_6 では、P E 1 ~ P E 1 2 7 のプロセッサエレメントに対応したsep型変数 E R h にsep型変数 E R R ' / 2 を代入する。ここで、変数 E R h は注目画素を 2 値化した際の 2 値化誤差の主走査方向への分配値を示し、後述するsep型変数 E R v は、副走査方向への誤差分配値を示す。図 1 1 に注目画素を 2 値化した際に発生する 2 値化誤差の分配方法を示す。E R R は、注目画素を 2 値化した際に発生する 2 値化誤差が記憶されたsep型変数であり、E R R の 1 / 2 が、主走査方向への分配誤差 E R h となり、副走査方向への分配誤差 E R v は、E R R - E R R / 2 となる。また、step_6 に示した変数 E R R ' はデータ演算処理部 3 0 3 中の各 P E について、左に隣接する P E が有する変数 E R R と同一変数を示し、P E 1 に対しては、P E 0 の変数 E R R 、P E 1 2 7 に対しては、P E 1 2 6 の変数 E R R に相当する。さらに、P E 0 に対しては、P E 1 2 7 の変数 E R R が相当することになる。これらの動作は、図 4 中のリング状に接続された P E レジスタに変数 E R R の値を設定することにより、全ての P E が同時に隣接する P E の変数 E R R を参照することにより実行される。10

【 0 0 9 2 】

Step_5 の処理は、主走査方向の先頭画素に対する処理に相当し、主走査方向の画素に分配される 2 値化誤差 E R h に所定値 ' 0 ' を設定するものである。以上のように、Step_5 、Step_6 の処理により、主走査方向の隣接画素より分配される 2 値化誤差 E R h の算出が行なわれる。20

【 0 0 9 3 】

次にStep_7 にてsep型変数 K が ' 0 ' 以下か否かが判定され、Step_8 、またはStep_9 の処理が行なわれる。K が ' 0 ' 以下の場合には、Step_8 にて、副走査方向への隣接画素に対する分配誤差を表すsep型変数 E R v に E R R B 「 i 」 が代入され、Step_7 の条件を満たさない場合には、Step_9 にて、E R R - E R R / 2 が代入される。ここで、E R R B [i] とは、2 値化処理により発生した 2 値化誤差の副走査方向への分配成分を記憶する変数であり、各 P E が有するメモリ 4 1 1 に設けられている。ここで、B N D = 0 かつ K = 0 となる P E が処理を行なうのは先頭主走査ラインの画像データに相当する。この場合には、先行する主走査ラインがないことより、注目画素の上部に隣接する画素より分配される 2 値化誤差 E R v を ' 0 ' として、画像端部の 2 値化処理を行うようにしてよい。また、B N D が ' 0 ' でない場合には、後述するStep_1 7 にて設定された 2 値化誤差の副走査方向への分配成分が E R v に設定されることになる。30

【 0 0 9 4 】

Step_9 では、各 P E が自身のプロセッサエレメントに属するsep型変数 E R R を参照して、副走査方向に隣接する画素が発生した 2 値化誤差の分配成分である E R v を算出する。

【 0 0 9 5 】

次にStep_1 0 にてsep型変数 D A T A [K] に隣接画素が発生した 2 値化誤差 E R h 、 E R v を加算し、誤差補正したsep型変数 D を算出する。ここで、D A T A [K] は、各 P E のメモリ 4 1 1 のアドレス K (Step_3 にて算出) に記憶された入力画像データを示している。40

【 0 0 9 6 】

次に、Step_1 1 にて、D が 1 2 7 よりも大きいか否かが判別され、大きい場合には、Step_1 2 にて、sep型変数 O U T に 2 5 5 を、大きくない場合には、Step_1 3 にて O U T に 0 が代入される。Step_1 2 、Step_1 3 により誤差補正された 8 ビットの入力画像データが 0 または、2 5 5 に 2 値化処理される。

【 0 0 9 7 】

Step_1 4 では、sep型変数 K が 0 以上かつ 1 2 7 以下か否かが判別され、Step_1 5 また

50

はStep_18に分岐する。変数Kが0以上127以下の範囲でないPEが算出した2値化処理結果は、Step_15、Step_16、Step_17が実行されないことにより破棄されることになる。

【0098】

Step_15では、前述したメモリ411のDATA[K]の領域に2値化済みデータOUTを記憶し、2値化誤差ERRにD-OUTを代入し、Step_16に分岐する。

【0099】

Step_16では、Kが127に等しいか否かが判別され、等しい場合にはStep_17に、等しくない場合にはStep_18に分岐する。Step_17は、隣接するバンドの先頭ラインの画素データに対する2値化誤差の分配値をERRB[i]に記憶する。さらに、iに1を加算した値を56で割り算した余りをiに設定する処理を行なう。
10

【0100】

次に、Step_18では、Lに1を加算する。

【0101】

次にフローはStep_19に進み、Lが127より大きいか否かが判別され、大きくない場合にはStep_20に進みNに56が加算され、Step_27を介してStep_3に分岐する。

【0102】

ここまで処理では、図10に示すとおり、sep型変数Kが0以上となるのは、PE0に対応した変数のみであるので、Step_15では、PE0の2値化処理のみ有効となり、図9の(a)に黒背景で示した画素位置データ(0,0)の2値化処理のみが終了し入力画像データに変わって2値化済データが記憶されている。また、2値化誤差を記憶するsep型変数ERRに関しても、PE0に対応する変数ERRにはStep_15にて2値化誤差が記憶されている。次にStep_20からStep_27を介してStep_3にフローが分岐し、Step_3を実行した時のデータ演算処理部303のメモリ411に転送されるデータは、図9の(Bb)に示したものとなる。また、この場合には、L=1となっていることより、PE0では、K=1、PE1では、K=0となる。(図10参照)。
20

【0103】

Step_4の判定において、PE1に対するフローはStep_6に分岐する。PE1の処理では、Step_6にて直前のStep_15が実行された際にPE0のERRに記憶された2値化誤差、すなわち、PE0が入力データ(0,0)を2値化した際に生じた2値化誤差ERRをERR'として参照し、入力データ(0,1)に対する誤差補正值ERhを算出する。
30

【0104】

Step_4におけるPE0に対する処理では、前回と同様にStep_5に分岐しPE0に対応するERhは'0'となる。

【0105】

次に、Step_7では、PE0に対応する変数Kが'1'となっていることより、PE0に対するフローは、Step_9に分岐する。Step_9では、PE0に対応したsep型変数ERRRを参照し、ERVを算出する。このときにPE0が使用するERRRの値は、前回のStep_15の処理にて、PE0が入力データ(0,0)を2値化した際に算出された2値化誤差であり、Step_9でPE0が算出したERVは、入力データ(0,0)から入力データ(1,0)に分配される2値化誤差に相当するものである。
40

【0106】

また、PE1に対する変数Kは'0'となっていることより、PE1に対するフローは、Step_7よりStep_8に進む。ここで、PE1に対応するsep型変数i=0、ERRB[i]=0によりERVには、初期設定値である'0'が設定されることになる。

【0107】

以降フローが進みStep_14の判定にて0以上127以下の変数Kを有するプロセッサー要素は、PE0、とPE1ということになり、Step_15にて、PE0、PE1が算出した2値化結果が記憶されている変数OUTが、図9(b)中の入力画像(1,0)、(0,1)が記憶されていたアドレスに有効な2値化結果として記憶されることになる
50

。以上のように、 $L = 1$ の場合には P E 0、P E 1 の 2 個のプロセッサー要素が同時に入力画像データの 2 値化処理を行なうことになり。さらに、Step_1 8 にて L が加算され、上記フローを繰り返す。

【0108】

以上説明したように、L が加算されるにつれて、有効な 2 値化結果を算出するプロセッサー要素の数も増加していくことになる。

【0109】

次に $L = 127$ となって、Step_2 7 を介してから Step_3 に分岐した場合について説明する。

【0110】

$L = 127$ の場合に、Step_3 にてラインメモリアドレス N の 128 画素データを内部メモリ 411 に転送した後の内部メモリ 411 の画素配置を図 9 の (c) に示す。また、この時の sep 型変数 K の値は、図 10 より P E 0 ~ P E 127 まで全ての P E に対して 0 以上 127 以下の値となり、全ての P E が並列に有効な 2 値化結果を算出することになる。このとき各 P E が並列に 2 値化処理を行なう画素位置は、図 9 (c) に斜め線にて示されている。 $L = 127$ の時にフローが Step_1 6 に至ると P E 0 のプロセッサー要素に対する変数 K が 127 と等しいことにより、P E 0 に対する処理は、Step_1 6 より Step_1 7 に分岐し、Step_1 7 の処理が実行される。Step_1 7 では、P E 0 が演算した 2 値化誤差の副走査方向への分配成分を算出し E R R B [i] に設定する。ここでは、i の値は、'0' となっていることより E R R B [0] に 2 値化誤差の副走査方向への分配成分が設定され、P E 0 に対応する sep 型変数 i には 1 が加算される。

10

【0111】

次に、Step_1 8 より Step_1 9 に至ると、Step_1 9 より Step_2 1 に進む。

【0112】

Step_2 1 では、変数 L を 128 で割り算した余りの値で示されるアドレス（この場合は、'0'）のメモリ 411 の値をワーキングメモリ 216 の対応するラインアドレスに転送する。対応するラインアドレスとは、図 9 (c) に示すとおり、メモリ 411 のアドレス '0' に設定されている 2 値化済みの原稿画像データに対応する (0, 0) より (0, 127) が記憶されているラインアドレスに一致したものである。この場合には、ワーキングメモリのラインアドレス '0' が転送先のアドレスとして設定される。メモリ 411 からワーキングメモリ 216 へのデータ転送は、各 P E が並列で動作し、図 9 (c) に示した、2 値化済のデータ 128 画素分がワーキングメモリに転送され入力画像データ (0, 0) より、(0, 127) の 128 画素分が記憶されていたワーキングメモリ 216 の領域に、対応する 2 値化済データが上書きされることになる。

20
30

【0113】

次にフローは Step_2 2 に進み、変数 L を 128 で割り算した余りが '0' か否かが判定される。この場合には、 $L = 128$ となっているのでフローは Step_2 3 に進む。Step_2 3 にて B L K は B L K = 1 となり、Step_2 4 より、Step_2 6 に分岐する。Step_2 6 では、N に B L K + B N D × 56 × 128 の値が代入される。この場合は、N = 1 となる。そして Step_2 6 より Step_2 7 を介して Step_3 にフローは分岐し、フローを繰り返す。

40

【0114】

Step_3 では、ワーキングメモリのラインアドレス 1 に対応するデータが、内部メモリ 411 のアドレス '0' に転送される。この時の内部メモリの内容を図 9 の (d) に示す。

【0115】

図 9 (d) に示すように、内部メモリのアドレス 0 には原稿画像データの先頭ライン 128 ~ 255 画素のデータが記憶されることになる。

【0116】

また、図 10 に示した $L = 128$ に対応する各 P E の変数 K の値より、この場合に各 P E が並列に 2 値化動作を行なう原稿画像データの配置は図 9 中の線で示したライン上、別の図示では、図 12 中の B の 128 画素となっている。このような場合に P E 0 が担当する

50

画素データ(0、128)に対する主走査方向からの2値化誤差分配値は、Step_6において、P E 1 2 7 の2値化誤差が記憶されている変数E R R を図4のリング上に接続されたP E レジスタを使用して、E R R 'として参照することにより、E R h に設定されることになる。さらに、Step_1 7 では、P E 1 に対応するE R R B [0] に副走査方向への2値化誤差の分配値が設定される。

【0117】

上記のような動作によりフローを繰り返す。

【0118】

次に、L = 7 1 6 7 の時の内部メモリ4 1 1 に転送された画像データを図9(e)に示す。この場合に各P E が並列に2値化動作を行なう原稿画像データの配置は、図9(e)の10
ライン上、別の図示では、図12中のCで示したライン上の128画素となっている。L = 7 1 6 7 の場合に、Step_3 よりフローを実行し、Step 2 1 に至った場合には、内部メモリ4 1 1 のアドレス0の128画素の2値化処理済みデータがワーキングメモリ2 1 6 の対応するラインアドレスに転送されることになる。これにより、原稿画像データの最初の1ライン：7 1 6 8 画素の2値化処理が全て終了したことになる。

【0119】

さらにフローは、Step_2 2 の判定を行ない、 $L = (7 1 6 7 + 1) \% 1 2 8 = 0$ より、Step_2 3 に進む。Step_2 3 、Step_2 4 からは、B L K = (5 5 + 1) \% 5 6 = 0 より、Step_2 5 に進む。Step_2 5 では、B N D に1が加算され、Step_2 6 より、Step_2 7 に分岐する。Step_2 6 では、N に、 $N = B N D \times 5 6 \times 1 2 8$ が設定される。この場合には、B N D = 1 となるので、N = 7 1 6 8 が設定され、Step_2 7 を介して、Step_3 に分岐する。この時Step_3 の実行後の内部メモリ4 1 1 に転送されている画像データを図9(f)に示す。図に示すように内部メモリ4 1 1 のアドレス0には、原稿画像データの128ラインの先頭から128画素が設定されていることになる。

【0120】

さらに、フローが進みStep_7 に至ると、P E 0 に対するsep型変数K はK = 0 となっていることにより、P E 0 に対するフローは、Step_8 に進み、P E 0 に対応したE R R B [0] が、副走査方向の隣接画素より分配される2値化誤差として、E R v に設定される。この値は、図12中Aのライン上の画像データを2値化処理した際に、Step_1 7 にてP E 0 に対応するE R R B [0] に設定された隣接する副走査方向へ分配される2値化誤差30
に相当するものである。

【0121】

同様にして、B N D = 1 の先頭ラインを処理する際には、それぞれのP E が対応する副走査方向の隣接画素より分配された2値化誤差であるE R R B [0] をE R v として使用することにより、B N D = 0 の最終ライン画素からB N D = 1 の先頭ライン画素への2値化誤差伝播が行なわれることになる。これを繰り返すことにより、B N D の最終ライン画素から隣接する副走査画素であるB N D + 1 の先頭画素への2値化誤差伝播が行なわれることになる。

【0122】

以上のフローを繰り返して、L = 2 7 9 5 5 1 の時の、内部メモリ4 1 1 に転送された画像データを図9(g)に示す。L = 2 7 9 5 5 2 の時の、内部メモリ4 1 1 に転送された画像データを図9(h)に示す。これらは、いずれもB N D = 3 8 、B L K = 5 5 となる最終ブロックに対する処理となる。L = 2 7 9 5 5 1 では、P E 0 ~ P E 1 2 7 までの全てのP E が有効な2値化処理を行なうが、L = 2 7 9 5 5 2 の場合には、P E 0 が行なう2値化処理は無効画素に対するものとなる。以降、無効画素を2値化するP E は順次増加していく。

【0123】

次にL = 2 7 9 6 7 8 の時の内部メモリ4 1 1 に転送された画像データを図9(i)に示す。

【0124】

10

20

30

40

50

この場合には、P E 1 2 7 のみが有効な 2 値化処理を行なうことになる。この処理を行なったフローにおけるStep_2_7 の判定にて、 $L = (279678 + 1) = 279679$ となっていることよりフローを終了する。

【0125】

以上のようにして、A 4 原稿画像 1 面分である 7168×4992 画素の 2 値化処理が終了し、2 値化結果がワーキングメモリ 216 に記憶されることになる。

【0126】

ワーキングメモリメモリ 216 に記憶された 2 値化済データは、図 3 に示した演算ユニット 300 の外部メモリインターフェース 306 を介して出力レジスタ 304 に入力される。次に、ワーキングメモリメモリ 216 より読み出された 2 値化済データは、出力レジスタ 304 から FIFO メモリ 307 に書き込まれる。FIFO 307 へのデータ書き込みと読み出しの関係を図 13 に示す。10

【0127】

FIFO メモリ 307 は、1 ライン分 (7168 画素) の容量を有するラインメモリであり画像処理ユニット 300 より出力されるクロックにより出力レジスタ 304 より出力された画像データを FIFO メモリ 307 にライトする。リードクロックは、作像エンジン 205 の動作に同期したクロックとなっている。なお、本実施の形態では、FIFO メモリ 307 へのライトクロックが 100MHz 、リードクロックは、 20MHz と設定されており、リードクロックに対してライトクロックが高速なものとなっている。FIFO メモリ 307 に 0 画素目の画像データがライトされた後の、リード開始 (0) のタイミングより、0 画素目の画像データの読み出しが開始される。図 13 に示したように、FIFO 307 への 2 値化済データの書き込みは、128 画素単位で行なわれる。また、読み出しへは、作像ユニット 205 動作に同期したクロックにより、1 ライン毎に連続した画像データとなるように行なわれる。20

【0128】

FIFO 307 から読み出された 2 値化済データは、作像ユニット 205 に入力され作像エンジン 205 入力データに応じたプリント処理をおこなわれることになる。

【0129】

以上のようにして読み取りユニット 201 により読みとられた A 4 サイズの原稿画像データの 2 値化処理が行なわれプリント出力が得られることになる。30

【0130】

本実施の形態では、読み取りユニット 201 より読み取られた画像データを図 7 に示したように、1 ラインを連続する 56 アドレスに記憶しているが、記憶アドレスは、これに限定されるものではなく、図 14 に示すように、主走査方向の先頭より、128 画素単位で、連続するラインを連続するアドレスに記憶するようにしてもよい。

【0131】

さらに、読み取りユニット 201 より画像処理プロセッサ - 204 に入力される画像クロック速度に対して、画像処理プロセッサ - の動作クロックが十分に高速で、リアルタイム処理が可能な場合には、入力画像データ読み取りユニット 201 より読み取られた画像データをワーキングメモリ 216 に記憶することなく 2 値化処理を行なう構成としてもよい。その場合には、図 3 の入力レジスタ 302 より入力された画像データをデータ演算処理部 303 の内部メモリ 411 に取り込み、2 値化処理が終了した画素位置の内部メモリを 2 値化データに書きかえるようにし、図 12 に示した 1 バンド分の 2 値化処理が終了した時点より、内部メモリの 2 値化済データを出力レジスタを介して、作像ユニット 205 に出力させるようにする構成としてもよい。40

【0132】

さらに、本実施の形態の誤差拡散処理における誤差データは、図 11 に示した主走査、副走査方向それぞれ 1 画素に分配されているが、誤差拡散の方法はこれに限定されるものではなく図 15 に示すような場合であっても適用可能なことは自明である。図 15 には誤差を配分する画素を夫々 3 画素 (パターン 1)、4 画素 (パターン 2)、7 画素 (パターン

3)、9画素(パターン4)の例を示し、又各画素位置への誤差配分率の例を示したが、これらの例に限定される物では無い。図中*印は夫々のPEが今同時に処理しようとする注目画素位置であり、誤差は該注目画素位置を含まず、注目画素位置に近い画素から約10画素程度の画素に配分すれば良好な画像が得られる。

【0133】

尚、図中X印は各誤差の配分演算で生じた演算誤差を吸収する画素位置であり、誤差の総和から他の画素への配分誤差の総和を減算した余り誤差を割り当てる。

【0134】

又、図15に示すパターンで誤差を配分する場合、先に説明した繋ぎ誤差ERRBは各PEで各バンドに対して数バイト必要になる。

10

【0135】

以上説明した如く、本発明の第1の実施の形態によれば、逐次処理を行うための補助的な演算処理部を利用することなく、逐次処理の代表例である誤差拡散処理を SIMD型演算部用いて行うことが可能となり、補助的な演算処理部のコスト削減効果がある。

【0136】

さらには、1ラインの画素数、および、ライン数が、SIMD型演算部が有すプロセッサーエレメントの数を超える2次元画像データ入力信号に対しても、2次元画像データを分割して、SIMD型演算部のメモリに記憶し、処理を行うことにより、各プロセッサエレメントには、異なる画素位置の画像データを供給して、各プロセッサーエレメントを並列動作させることができとなり、SIMD型プロセッサーの並列処理能力を有効利用した、高速な処理を行えるという効果がある。

20

【0137】

さらには、分割した領域間においても、誤差拡散処理にて発生した誤差の分配を行い、誤差補正処理を行うことにより入力信号の濃度を保存した高品位な誤差拡散処理画像が得られるという効果がある。

【0138】

(第2の実施の形態)

第2の実施の形態では、更に高画質を得るために誤差拡散処理にて発生した誤差データを拡散する画素位置の変更例に関して説明する。

【0139】

30

図16に、第2の実施の形態にて拡散される誤差データの画素位置を示す。

【0140】

第2の実施の形態は、図11、図15に示した誤差データの拡散位置の例では行っていない注目画素に隣接する左斜め下、図16の(a)中Aにて示す画素位置へ誤差の拡散を行う場合に適用される。

【0141】

第1の実施の形態と同様にして、図5に示した原稿画像データは、図3中の FIFOメモリ301に入力される。本実施の形態では、7168画素で構成される1ラインデータを偶数番目と奇数番目の画素に振り分けて、演算処理ユニット300中の入力レジスタ302に入力する。図17に、FIFOメモリ301の構成を示し、図18にFIFOメモリ301より入出力される画像データとの関係を示す。図17中1700はデータ制御部203より入力される原稿画像データを奇数番目の画素の偶数番目の画素に振り分けるセレクタ、1701はセレクタ1700より出力された偶数番目の画素データが入力されるFIFO1メモリ、1702はセレクタ1700より出力された奇数番目の原稿画像データが入力されるFIFO2メモリ、1703はFIFO1メモリ1701、FIFO2メモリ1702の出力データが入力され、どちらかの入力データを出力するセレクタであり、セレクタ1703の出力信号は、図3中の入力レジスタ302への入力信号となる。

40

【0142】

図18は、図17で構成されるFIFOメモリ301の動作を説明するタイムチャートである。7168画素で構成される1ラインのデータは、0ラインの0画素目より順に、デ

50

ータ制御部203よりスイッチ1700に入力され、原稿画像データに同期したクロックにより、偶数番目の画素データ0、2、4、……7166が、 FIFO1メモリ1701にライトされ、奇数番目の画素データ1、3、5、……7167が、 FIFO2メモリ1701にライトされる。 FIFO1メモリ1701に0～254番目までの128画素分の画像データのライトが終了すると、リード開始タイミング(0)より、 FIFO1メモリ1701から128画素分の画像データのリードが開始され、リードされた画像データは、セレクタ1703を介して、入力レジスタ302へ入力される。なお、 FIFO1メモリ1701、 FIFO2メモリ1702に対するリードクロックは、画像処理ユニット300より出力されるクロックであり、前述したのと同様に、ライトクロックに対して、高速なクロックとなっている。

10

【0143】

FIFO1メモリ1701より0～254番目までの128画素の画像データのリードが終了すると、次に、 FIFO2メモリ1702より、原稿画像データの1～255番目までの128画素分の画像データのリードが開始され、 FIFO2より読み出された奇数番目の画像データは、セレクタ1703を介して、入力レジスタ202へ入力される。 FIFO2メモリ1702から128画素分の画像データのリード動作が終了すると、再度、 FIFO1メモリ1701より128画素分の原稿画像データのリード動作が開始され、256、258、……510までの、原稿画像データが、セレクタ1703を介して、入力レジスタに入力される。以降同様の動作を繰り返し、原稿画像データは、偶数番目の128画素と奇数番目の128画素が交互に、入力レジスタ302に入力され、入力レジスタ302に入力された画像データは、前述したようにワーキングメモリ216に記憶される。図19に、ワーキングメモリ216に記憶される原稿画像データとワーキングメモリのアドレスとの関係を示す。図19に示すように、ワーキングメモリ216には、偶数番目の画像データと奇数番目の画像データが128画素ずつ順に記憶される。

20

【0144】

以下に第2の実施の形態の動作に関して図20を用いて詳細に説明する。なお、第1の実施の形態における図8の説明と重複する動作に関しては、説明を簡略化する。

【0145】

Step_101にて、128ライン分の入力画像データがワーキングメモリ216に取り込まれるまで待機する。ワーキングメモリ216に、128ライン分の入力画像データ取り込みが終了すると、フローは、Step_102に進み、変数の初期化が行なわれる。

30

【0146】

Step_102では、変数N(ワーキングメモリのラインアドレス)、L(データ転送回数)、BND(バンド)、BLK(ブロック)の初期化、および、sep型変数i(つなぎ誤差のインデックス)、sep型変数ERRBE[0]～ERRBE[27](偶数番目画素用つなぎ誤差データ28個)、および、sep型変数ERRBO[0]～ERRBO[27](奇数番目画素用つなぎ誤差データ28個)の初期化を行う。ここで、BND(バンド)とは、入力画像データの副走査方向の128ラインを1バンドとする変数、BLK(ブロック)とは、入力画像データの主走査方向の256画素を1ブロックとする変数である。ここでは、A4サイズの原稿画像データ；主走査画素数=7168、副走査ライン数=4992とし、BNDは、0～38、BLKおよびiは、0～27の値をとる。また、sep型変数ERRBE[0]～ERRBE[27]、ERRBO[0]～ERRBO[27]は、各バンドの最終ラインの2値化処理により発生した2値化誤差を記憶するメモリであり、各PEそれぞれにつきブロック数に等しい28個が設けられている。また、2値化誤差の配分は、図16(a)に示した注目画素周辺の4画素に対して行なわれ、右斜め下、左斜め下、下に隣接する3画素それぞれに、注目画素が発生した2値化誤差の1/4が分配され、注目画素の右に隣接する1画素には、上記3画素に分配した残りの誤差が分配される。

40

【0147】

次にStep_103にて、ワーキングメモリ216のラインアドレスNの128画素分のデ

50

ータをデータ演算処理部303の各PEが有するメモリ411の‘L%256’のアドレスに転送する。また、sep型変数Kの算出を行う。Kの算出値とLの関係を図21に示す、図21中、Kの値が、0以上、255以下のPEが有効な2値化処理を行い、これ以外の、Kの値を有するPEが算出した2値化データは、無効データとして破棄される。また、Kが偶数のPEは、主走査ライン方向の偶数番目の画像データを、Kが奇数のPEは奇数番目の画像データを処理するように制御され、K=0のPEは、各BNDの先頭ラインの偶数画素位置、K=1のPEは、各BNDの先頭ラインの奇数画素位置、K=254のPEは、各BNDの最終ラインの偶数画素位置、K=255のPEは、各BNDの最終ラインの奇数画素位置の画像データの処理を行うことになる。

【0148】

10

次にフローはstep_104に進み、PENUM=0、かつ、L%2=0、かつBLK=0の場合には、step_105に、そうでない場合には、step_1060に分岐する。Step_105では、PENUM=0の条件より、プロセッサーエレメントPE0に対応したsep型変数ERhに0が代入される。Step_1060では、L%2=0か否かが判別される。L%2=0とは、Lが偶数の場合に相当し、Lが偶数の場合には、Step_1062に分岐することになる。

【0149】

ここで、以降のStepにて使用されるsep型変数ERRRE[0]、ERRRE[1]、ERRRO[0]、ERRRO[1]に関して説明する。

【0150】

20

本実施の形態では、後述するように、各PEはそれぞれ、主走査方向に連続する2画素のデータの2値化処理するように制御され、上記sep型変数ERRRE[0]、ERRRE[1]、ERRRO[0]、ERRRO[1]は、それぞれのPEが2値化処理を行った際に発生する2値化誤差を記憶する領域として使用される。ERRRE[0]、ERRRE[1]には、各PEが偶数番目の画素データを処理した際に発生する2値化誤差が記憶され、ERRRE[0]には、直前に処理された偶数画素データの2値化誤差、ERRRE[1]には、その前に処理された偶数画素番目の2値化誤差が記憶されている。同様に、ERRRO[0]、ERRRO[1]には、それぞれ、各PEが奇数番目の画素データを処理した際に発生した2値化誤差が記憶されているものである。また、ERRRE'[0]、ERRRE'[1]は、各PEに対して左に隣接したPEが有するERRRE[0]、ERRRE[1]に相当し、ERRRE''[0]は、右に隣接したPEが有するERRRE[0]、ERRRE[1]に相当するものである。同様に、ERRRO'[0]、ERRRO'[1]は、各PEの左に隣接するPEが有するERRRO[0]、ERRRO[1]に相当し、ERRRO''[0]、ERRRO''[1]は、それぞれ右に隣接するPEが有するERRRO[0]、ERRRO[1]に相当するものである。第1の実施の形態にて説明したのと同様にして、PE0の左隣接PEには、PE127が相当し、PE127の右隣接PEには、PE0が相当する。Step_105、Step_1061、Step_1062では、各PEに対して、左隣接画素から分配される2値化誤差がERhに代入される。Step_105は、原稿画像データの主走査先頭画素を処理する際に実行されるステップを示し、ここでは、左隣接画素からの分配誤差を‘0’としている。

30

【0151】

40

次に、Step_107にて、Kが1以下か否かが判別される。Kが1以下でない場合は、Step_1090に分岐し、Lが奇数か偶数かによって、Step_1091、Step_1092が実行される。Step_1091、Step_1092は、それぞれ、図16(b)中の1、2、3で示した画素位置から分配される2値化誤差データの総計をERVに代入するステップである。また、Step_107にてKが1以下と判定された場合には、Step_1080にて、Lが奇数か偶数かによって、Step_1081、Step_1082が実行される。Step_1081、Step_1082は、各バンドの最初の主走査ラインの画像データを処理する際に実行されるステップであり、前バンドの最終主走査ラインの画像データを処理した際に発生した2値化誤差を参照して、各バンドの最初の主走査ラインの画像データに対する分配誤差を算出す

50

るステップである。なお、最初のバンドの先頭主走査ラインの画像データに対しては、先行する主走査ライン画像データより分配される2値化誤差はないことより、Step_1081、Step_1082で算出されるERVは、'0'と設定されるように制御される。さらに、各バンドにおける先頭ブロックにおいて、PE0が処理する偶数画素位置は、主走査先頭画素に相当し、図16(b)に示した、'1'の方向より2値化誤差を分配する画素が存在しないことより、この場合には、Step_1092におけるERRO'[1]/4を'0'に、さらに、Step_1082におけるERRBO'[i]/4を'0'に設定するように制御して、画像端部の2値化処理を行うようにしてもよい。

【0152】

次にStep_110にてsep型変数DATA[K]に隣接画素が発生した2値化誤差ERh、ERVを加算し、誤差補正したsep型変数Dを算出する。ここで、DATA[K]は、各PEのメモリ411のアドレスKに記憶された入力画像データを示している。

10

【0153】

次に、Step_111、Step_112、Step_113にて2値化処理が行われ、Step_114にて、Kが0以上、255以下か否かが判別される変数Kが0以上255以下の範囲でない場合は、Step_118に分岐し、2値化結果は破棄される。Step_115では、メモリ411のDATA[K]の領域に、2値化済みデータOUTを記憶し、Step_1150に分岐する。

【0154】

Step_1150では、Lが奇数か偶数かが判別され、奇数の場合には、Step_1151に分岐し、ERRO[1]にERRO[0]を代入し、ERRO[0]に2値化誤差D-OUTを代入する。Lが偶数の場合には、Step_1152に進み、ERRRE[1]にERRRE[0]を代入し、ERRRE[0]に2値化誤差D-OUTを代入する。後述するが、Lが偶数の場合に各PEが処理する画素位置は、主走査方向の偶数番目の画素位置に対応し、Lが奇数の場合に各PEが処理する画素位置は、主走査方向の奇数番目の画素位置に対応している。Step_1151、Step_1152により、処理を行った画素位置に応じた2値化誤差の記憶が行われることになる。

20

【0155】

次に、Step_116にてKが254か否かが判定され、254に等しい場合にはStep_1160に分岐し、2値化誤差ERRRE[0]をERRBE[i]に代入し、Step_118に分岐する。Step_117では、Kが255か否かが判定され、255に等しい場合にはStep_1170に分岐し、2値化誤差ERRO[0]をERRBE[i]に代入し、iに1を加算して、28で割り算した余りをiに代入する処理を行いStep_118に進む。Kが254に等しいPEが処理した画素位置は、各BNDの最終主走査ラインの偶数画素位置、Kが255に等しいPEが処理した画素位置は、各BNDの最終主走査ラインの奇数画素位置となることより、Step_1160、Step_1170により、各BNDの最終主走査ラインの画像データを処理した際に発生した2値化誤差が、ERRBE[i]、ERRBO[i]に記憶されることになる。

30

【0156】

次にStep_118にて、Lに1が加算され、Step_119にて、Lが254よりも大きいか否かが判別され、大きくない場合にはStep_1190に分岐する。Step_1190では、Lが奇数か偶数かが判別され、偶数の場合にはStep_120へ、奇数の場合にはStep_1191に分岐する。ここで、判別されるLは、次に処理を行うライン番号に相当し、次に処理するラインが奇数の場合には、Nに1を加算し、偶数の場合には、55を加算する。Nの値は、Step_103の処理にて、内部メモリに転送されるワーキングメモリのアドレスとなっており、ここで、アドレスの変更が順次行われることになる。

40

【0157】

次にフローは、Step_127に分岐し、さらに、Step_103に進んでフローを繰り返す。

【0158】

ここで、Step_102における初期化以降の動作より、Step_103以降のフローを繰り返

50

す際の動作に関して説明する。図22は、Lの値に応じて内部メモリ411の内容を示した図面である。図22(a)はL=0の時の場合を示す。初めて、Step_103を実行した際には、N=L=0により、ワーキングメモリのラインアドレス0の128画素のデータが、内部メモリ411のアドレス0に転送される。また、図21に示すように、L=0の場合に0以上255以下のKの値を有するPEは、PE0のみとなるので、L=0の場合にはPE0のみが有効な2値化処理を行うことになる。また、PE0に対しては、Step_105により主走査方向の左隣接画素より分配される2値化誤差ERhは'0'となり、さらに、Step_1082には図示していないが、前述したように先頭BNDの先頭ライン画素データであることにより、この場合には、Step_1082にてERvも'0'とされる。

10

【0159】

Step_110では、PE0は、K=0により、内部アドレスのアドレス'0'のデータ、原稿画像データの(0, 0)の画像データがDに代入され、Step_111、Step_112、Step_113にて2値化処理が行われる。さらに、Step_115にて、2値化結果が、内部アドレスのアドレス'0'に代入される。次に、Step_1152にて、2値化誤差が、ERR[E]に記憶される。

【0160】

以上のようにして、図22の(a)に示すように、L=0の場合には、原稿画像データの(0, 0)のデータのみの2値化結果が内部メモリ411のアドレス0に記憶されることになる。次に、Step_118にて、L=1となり、フローを進むと、フローは、Step_1191に至る。Step_1191では、Nに1が加算され、フローを進む。

20

【0161】

このようにして、再度Step_103を実行した際の内部メモリ411の様子を図22の(b)に示す。N=1となっていることにより、図19のワーキングメモリのラインアドレス1の128画素データが内部メモリに転送される。図21より、この場合にも、有効なKの値を有するPEは、PE0のみであることより、PE0のみが有効な2値化処理を行うことになる。Step_103よりフローが進むと、Step_104よりStep_1060をへて、Step_1061にフローは進む。ここでは、ERhに、ERR[E][0]-3*(ERR[E][0]/4)が代入される。ここで、参照されるERR[E][0]とは、PE0が最初に処理を行った2値化処理にて発生した2値化誤差であり、ここで、図16(a)のBで示す左隣接画素より分配される2値化誤差の算出が行われる。さらに、フローが進むとStep_115にて、PE0が算出した2値化結果が、内部アドレスのアドレス'1'に代入され、Step_1151にて、2値化誤差が、ERR[0]に記憶される。

30

【0162】

Step_118にて、Lに1が加算され、L=2となり、Step_119、Step_1190を経てStep_1120にフローが至る。ここでは、Nに55が加算され、N=56となり、フローが進む。次にStep_103にフローが進んだ際には、ワーキングメモリのラインアドレスN=56の128画素データが、内部メモリ411のアドレス2に転送される。さらに、図21に示すように、PE1の変数Kが0となり、有効な2値化処理を行うPEとして、PE1が加わることになる。以上のようにして、フローを繰り返すごとに、有効な2値化結果を算出するPEの数が増加し、L=254の場合には、PE0~PE127のすべてのPEが有効な2値化結果を算出するように動作する。実施の形態1にて説明したように、PE0~PE127は、同時動作を行うプロセッサーエレメントであることより、L=254以降は、PE0~PE127すべてのPEが同時に有効な2値化処理を行うように動作する。

40

【0163】

ここで、図23に、Lの増加に対応して2値化処理が行われる原稿画像データの配置と、2値化処理の順番を示す。図中 PE0~PE127の下方向に示される原稿画像データは、対応する同一のPEにより処理される画素位置であることを示している。また、上述した、BNDとBLKの関係も示している。各BNDは、原稿画像データの128ライン

50

に対応し、各 B L K は、主走査画素数 256 画素に対応したものとなる。

【0164】

ここで、 $L=0$ の時に処理される 2 値化データは、図中 '0' で示された画素位置のデータであり、これは、原稿画像の (0, 0) 位置データに相当する。

【0165】

$L=1$ の時に処理される 2 値化データは、図中 '1' で示された画素位置データであり、これは、原稿画像の (0, 1) の位置データに相当する。

【0166】

$L=2$ の時に処理される 2 値化データは、図中 '2' で示された画素位置データであり、これは、原稿画像の (1, 0)、(0, 2) の位置データに相当する。

10

【0167】

以下同様にして、 $L=M$ の時に処理される原稿画像の画素位置データには 'M' の値が記入してある。図 23 に示すように、本実施の形態では、主走査方向の原稿画像データを 2 画素一組として、同一の P E が処理を行うように構成されている。図 23 に示したように、 $L=0, 1$ では、有効な 2 値化処理を行う P E は、P E 0 のみであるが、 L が増加するにつれて、有効な 2 値化処理を行う P E の数が増加し、 $L=254$ 以降は、128 個の P E がすべて有効な 2 値化処理を行うことになる。

【0168】

ここで、図 23 中の $L=5$ の時に処理される画像データに対する 2 値化誤差の分配について詳細に説明する。 $L=5$ の時に処理される画像データは、図中 '5' の番号を付した画素位置のものである。ここで '5' の番号を付した画素位置に注目すると、図 16 (b) に示した注目画素位置に対して 2 値化誤差を分配する周辺の 4 画素の画素位置データの番号がすべて 5 より小さくなっていることより、すでに 2 値化処理が終了していることがわかる。

20

【0169】

左隣接画素の番号 '4' の値が付された画素位置からの分配誤差は、上述した Step_1061 にて、

$$E R h = E R R E [0] - 3 * (E R R E [0] / 4)$$

として算出される。図 23 中番号 '4' が付された左隣接画素位置の 2 値化処理は、番号 '5' が付された P E と同一番号の P E が、直前に行った偶数番目の画像データに対する処理であることより、Step_1061 では、同一 P E が有する偶数番目の画素位置を処理した際に発生した 2 値化誤差である E R R E [0] (Step_1152 にて設定済み) より左隣接画素より分配される 2 値化誤差の算出を行っている。

30

【0170】

さらに Step_1091 では、図 23 中 '2'、'3'、'4' と番号が付された 1 ライン上部の隣接画素位置より分配される 2 値化誤差の総和を、

$$E R v = E R R E [1] / 4 + E R R O [0] / 4 + E R R E '' [0] / 4 \text{ として算出する。}$$

【0171】

ここで、E R R E [1] とは、図 23 中 '2' の番号が付された画素位置データを処理した際に発生した 2 値化誤差であり、これは、同一 P E が偶数番目の画素を処理した際に発生した 2 値化誤差として、Step_1152 にて設定されたものである。また、E R R O [0] は、図 23 中 '3' の番号が付された画素位置データを処理した際に発生した 2 値化誤差であり同一 P E が、直前に処理した奇数番目の画素より発生した 2 値化誤差として、Step_1151 にて設定されたものであり、さらに、E R R E '' [0] は、図 23 中 '4' の番号が付された画素位置データを処理した際に発生した 2 値化誤差であり、右隣接 P E が、直前に処理した偶数番目の画素より発生した 2 値化誤差として、Step_1151 にて設定されたものである。

40

【0172】

上記の関係は、図 23 中の番号 '5' を付した画素位置データに対する 2 値化誤差の分配

50

に関して説明したが、図23中の奇数番号を付した画素位置データに対する2値化誤差の分配に関しても同様である。

【0173】

次に図23中の番号'6'を付した画素位置データに対する2値化誤差の分配について説明する。番号'6'を付した画素位置データに対しては、番号'3'、'4'、'5'が付された画素位置を処理した際に発生した2値化誤差が分配されそれぞれ、すでに2値化処理は終了していることがわかる。

【0174】

左隣接画素の番号'5'が付された画素位置からの分配誤差は、Step_1062にて、

$$ERh = ERRO'[0] - 3 * (ERRO'[0] / 4) \quad 10$$

として算出される。図23中番号'5'が付された左隣接画素位置の2値化処理は、番号'6'が付されたPEの左隣接PEが、直前に行なった奇数番目の画像データに対する処理であることより、Step_1062では、左隣接PEが有する奇数番目の画素位置を処理した際に発生した2値化誤差であるERRO'[0](Step_1151にて設定済み)より左隣接画素より分配される2値化誤差の算出を行っている。

【0175】

さらにStep_1092では、図23中'3'、'4'、'5'と番号が付された1ライン上部の隣接画素位置より分配される2値化誤差の総和を、

$$ERV = ERRO'[1] / 4 + ERRE[0] / 4 + ERRO[0] / 4 \quad 20$$

として算出する。

【0176】

ここで、ERRO'[1]とは、図23中'3'の番号が付された画素位置データを処理した際に発生した2値化誤差であり、これは、左隣接PEが奇数番目の画素を処理した際に発生した2値化誤差として、Step_1151にて設定されたものであり、ERRE[0]は、図23中'4'の番号が付された画素位置データを処理した際に発生した2値化誤差であり同一PEが、直前に処理した偶数番目の画素より発生した2値化誤差として、Step_1152にて設定されたものであり、さらに、ERRO'[0]は、図23中'5'の番号が付された画素位置データを処理した際に発生した2値化誤差であり、同一PEが、直前に処理した奇数番目の画素をより、発生した2値化誤差として、Step_1151にて設定されたものである。 30

【0177】

上記の関係は、図23中の番号'6'を付した画素位置データに対する2値化誤差の分配について説明したが、図23中の偶数番号を付した画素位置データに対する2値化誤差の分配に関しても同様である。

【0178】

次にL=254の場合の動作について説明する。L=254にて、Step_103を実行すると、図21に示すように、PE127の変数Kが'0'となり、PE0～PE127までの128個のPEすべてが有効な2値化処理を実行することになる。図22(d)、および、図23に示すように、このときPE127が2値化を実施する画素位置は、最初の主走査ラインの255画素目であり、これは、原稿画像データの(0、254)の位置のデータとなる。また、PE0が2値化を実施するのは、原稿画像データの(127、0)の位置データとなる。L=254にて、フローが進むとPE0のsep型変数Kが254と等しいことにより、PE0に対する処理は、Step_116より、Step_1160にフローは進み、Step_1160にて、ERRBE[0]にERRE[0]が代入される。ここで、ERRE[0]とは、PE0がこのときに処理した原稿画像データの(127、0)の2値化誤差値であり、これが、ERRBE[0](このときi=0)に代入されることになる。これは、図23中のB0で示した、BND=0、BLK=0における最終主走査ライン中のPE0が担当する偶数画素位置にて発生した2値化誤差値となる。 40

【0179】

次にフローは、Step_118にて、L=254+1とされて、Step_119より、Step_12

50

1にフローは至る。ここで、内部メモリ411のアドレス(L - 255) % 256 の128画素のデータが、ワーキングメモリの対応するラインアドレスに転送される。この場合には、L=255となっていることより、内部メモリ411のアドレス'0'の128画素データが、ワーキングメモリのラインアドレス'0'に転送される。図22の(d)に示すように、この場合には、内部メモリ411の内部アドレス'0'の128画素データは、すべて2値化が終了していることにより、主走査方向の先頭より、偶数番目の原稿画像に対応する2値化結果が、128画素分ワーキングメモリに転送されることになる。次に、フローは、Step_122より、Step_1190に至り、L=255により、Step_1191にて、N=N+1 = 7112 + 1 = 7113となり、フローを繰り返す。この場合Step_103では、ワーキングメモリのラインアドレス7113の128画素データが、内部メモリのアドレス'255'に転送され、PE0に対する変数Kは、'255'となり、以降のフローが繰り返される。図22の(e)にL=255の時の内部メモリの画素データ配置を示す。この場合には、PE0に対する処理は、Step_117より、Step_1170にフローが至りここで、ERRBO[i] (このとき i=0) にERRO[0]が代入される。ここで、ERRO[0]とは、PE0がこのときに処理した原稿画像データの(127、1)の2値化誤差値であり、これが、ERRBO[0]に代入されることになる。これは、図23中のB1で示した、BND=0、BLK=0における最終主走査ライン中のPE0が担当する奇数画素位置にて発生した2値化誤差値となる。また、ここで、PE0が有するsep型変数iには1が加算される。以降フローが進むと、Step_118にて、L=255+1=256となり、Step_121にて、内部メモリアドレスの1ライン目の2値化結果をワーキングメモリに転送後、Step_122より、Step_123にフローは至る。ここで、BLKに1が加算され、この場合には、BLK=1となる。次にStep_124により、Step_126にて、N=1×2+0×28×256=2となり、Step_127を介して、フローは、Step_103に至る。

【0180】

Step_103では、L=256、N=2となっていることより、ワーキングメモリのラインアドレス'2'に記憶された128画素のデータが、内部メモリのアドレス'0'に転送されることになる。このときの内部メモリの内容を図22の(f)に示す。このとき、内部メモリのアドレス'0'には、原稿画像データの先頭主走査ラインの(0、256)～(0、512)までの偶数が素位置の128画素が記憶されていることになる。また、このときに、PE0に対するsep型変数KがK=0となっていることより、PE0は原稿画像データの(0、256)の画素位置の画像データの2値化処理を行うことになる。すなわち、PE0がL=256の時に2値化処理を行うのは、図23のCの画素位置に対応する原稿画像データということになる。すなわち、PE0が処理する原稿画像位置は、BND=0、BLK=1の先頭主走査ラインの先頭画素となる。この場合に、PE0が処理する画素位置に対する2値化誤差の配分ERhは、Step_1062によりPE127が有する2値化誤差メモリERRO'[0]より算出される。以降フローを繰り返し、L=257の場合の内部メモリの内容を図22(g)に示す。

【0181】

また、L=7167の場合の動作に関して以下に説明する。L=7167の場合の内部メモリの内容を図22(h)に示す。また、図23よりこの時に各PEが処理を行う画素位置は、BND=0のBLK=27のブロックとなり、PE0は、最終主走査ラインの奇数画素目である(127、6915)の原稿画像データの処理を行うことになる。この場合に、前述したのと同様にして、Step_103より、Step_117にフローが至ると、PE0のsep型変数K=255であることより、Step_1170にフローが進む。この場合のPE0に属するsep型変数iは変数BLKと等しい数値である'27'となっていることより、2値化誤差ERRBO[0]がERRBO[27]に記憶され、i=(27+1)%28=0が設定される。さらにフローが進みStep_121にフローが至ると、L=7168により、(L-255)%256=1により、内部メモリのアドレス'1'の2値化結果がワーキングメモリのアドレス'54'に転送され、フローは、Step_122より、L=7168

10

20

30

40

50

$\% 256 = 0$ により、Step_123に至る。ここで、 $B L K = (27 + 1) \% 28 = 0$ となり、Step_124より、Step_125に至り、 $B N D = B N D + 1 = 0 + 1 = 1$ となる。また、Step_126により、 $N = B L K \times 2 + B N D \times 28 \times 256 = 0 \times 2 + 1 \times 28 \times 256 = 7168$ となり、Step_127よりStep_103に分岐する。Step_103でワーキングメモリのラインアドレス $N = 7168$ の128画素データが、内部メモリのアドレス $L \% 256 = 7168 \% 256 = 0$ に転送される。このときの内部メモリの内容を図22(i)に示す。

【0182】

図22(i)、および、図23に示すように、このときPE0が処理する原稿画像のデータ位置は、(128, 0)となり、原稿画像の129ライン目の主走査の先頭画像データなる。この場合の2値化誤差ERh、および、ERVの算出し関して以下に説明する。
10

【0183】

Step_103より、Step_104に至るとPE0のフローは、PENUM=0、 $L = 7168$ 、 $B L K = 1$ により、Step_104に至り、 $ERh = 0$ が設定される。図23に示すように、このときPE0が処理する原稿画像データは129ライン目の主走査先頭画素位置にあることより、左隣接画素からの分配誤差ERhは'0'に設定される。次にStep_107にて、Kが1以下か否かが判別される。図21に示すように、この場合には、 $L = 7168$ であることより、PE0のsep型変数Kは $K = 0$ となっており、PE0のフローはStep_1082に至る。

【0184】

Step_1082では、

$ERV = ERRBO'[i] / 4 + ERRBE[i] / 4 + ERRBO[i] / 4$ を算出する。
20

【0185】

ここで、PE0に関するsep型変数iは、Step_1170で*i=0*に設定されていることにより、

$ERV = ERRBO'[0] / 4 + ERRBE[0] / 4 + ERRBO[0] / 4$ が代入される。

【0186】

ここで、 $ERRBO'[0]$ とは、PE0の左隣接画素のPEとして、PE127が有するsep型変数 $ERRBO[0]$ に相当するが、 $B L K = 0$ の場合PE0は、主走査の先頭画素を処理することより $ERRBO'[0] = 0$ が代入される。また。 $ERRBE[0]$ は、図23中のB0の位置の画素が発生した2値化誤差を示し、PE0が、 $L = 254$ の時に処理した画素位置が発生した2値化誤差であり、これは、Step_1160にて、 $ERRBE[0]$ に設定されていたものである。さらに、 $ERRBO[0]$ は、図23中のB1の位置の画素が発生した2値化誤差を示し、PE0が、 $L = 255$ の時に処理した画素位置が発生した2値化誤差であり、これは、Step_1170にて、 $ERRBO[0]$ に設定されていたものである。以下同様にして、フローを繰り返す。
30

【0187】

ここで、 $B N D = 1$ 、 $B L K = 0$ の場合における主走査先頭画素以外の2値化誤差の分配に関して $L = 7169$ の場合を例に説明する。 $L = 7169$ の場合、フローはStep_103、Step_104より、Step_1060より、Step_1061に至る。ここで、 ERh の算出が行われる。これは、上述したのと同様にして、各PEそれぞれが、偶数画素位置を処理した際に発生した2値化誤差が記憶されている同一PEのsep型変数 $ERRE[0]$ を参照して実施される。次に、フローは、Step_107に至る。このとき、PE0のsep型変数Kは、'1'となっていることより、PE0の処理は、Step_107より、Step_108に至り、 $L = 7169$ であることより、Step_1081に至る。Step_1081では、 $ERV = ERRBE[0] / 4 + ERRBO[0] / 4 + ERRBE'[0] / 4$ を算出する。

【0188】

10

20

30

40

50

ここで、ERRBE[0]とは、ERRBE[0]は、図23中のB0の位置の画素が発生した2値化誤差を示し、PE0が、L=254の時に処理した画素位置が発生した2値化誤差であり、これは、Step_1160にて、ERRBE[0]に設定されていたものである。さらに、ERRBO[0]は、図23中のB1の位置の画素が発生した2値化誤差を示し、PE0が、L=255の時に処理した画素位置が発生した2値化誤差であり、これは、Step_1170にて、ERRBO[0]に設定されていたものである。さらに、ERRBE'[0]は、図23中のB2の位置の画素が発生した2値化誤差を示し、PE1が、L=256の時に処理した画素位置が発生した2値化誤差であり、これは、Step_1160にて、ERRBO[0]に設定されていたものである。以上のように、ここで、算出されるERVは、図16(b)中の画素位置'1'、'2'、'3'で示す3画素から分配される2値化誤差の総和となる。
10

【0189】

同様にして、図23に示した、BND=1~38の先頭主走査ラインの各画素は、ひとつ上のBNDの最終主走査ラインを処理した際に発生した2値化誤差を記憶したsep型変数ERRBE[i]、ERRBO[i](Step_1160、または、Step_1170にて記憶)を参照することにより、ERVとして、図16(b)中の画素位置'1'、'2'、'3'で示す3画素から分配される2値化誤差の総和を算出することになる。

【0190】

以下フローを繰り返し、L=279806に達すると、Step_127にて、フローを終了する。また、この場合には、図23に示した原稿画像データのすべての2値化処理が終了することになる。
20

【0191】

以上説明したように、原稿画像データの2値化処理が終了すると、A4原稿画像1面分である7168×4992画素の2値化処理が終了し、2値化結果がワーキングメモリ216に記憶されることになる。

【0192】

ワーキングメモリ216に記憶された2値化済データは、ワーキングメモリのラインアドレス'0'より128画素毎に図3に示した、演算ユニット300の外部メモリインターフェース306を介して、出力レジスタ304に入力される。次に、ワーキングメモリメモリ216より読み出された2値化済データは、出力レジスタ304から FIFOメモリ307に書き込まれる。FIFO307のブロック図を図24に示す。図24の2300はスイッチであり、出力レジスタ304から出力される信号を128画素毎に FIFO1(偶数画素データ)、2301と FIFO2(奇数画素データ)2302に振り分けて出力する。FIFO1、FIFO2は、1/2ライン分(3584画素)の容量を有するラインメモリである。FIFO1、FIFO2の出力は、セレクタ2304に入力され、セレクタ2304では、2入力信号を1画素毎に切り換えて、画像データ制御部203へ出力する。図25を用いて図24のFIFOメモリ307の動作に関して説明する。
30

【0193】

画像処理ユニット300より出力されるクロックにより、出力レジスタ304より出力された画像データを128画素単位で、FIFO1、FIFO2にライトする。リードクロックは、作像エンジン205の動作に同期したクロックとなっている。なお、本実施の形態では、FIFO1、FIFO2へのライトクロックが100MHz、リードクロックは、20MHzと設定されており、リードクロックに対してライトクロックが高速なものとなっている。まず、出力レジスタ304の出力信号は、スイッチ2300より、FIFO1に接続され、0~254画素目までの、128画素がライトされるとスイッチ2300が切り替わり、FIFO2に1~255画素目までの128画素をライトする。以降128画素毎にスイッチを切り換えなら、0ライン目の偶数画素位置2値化済みデータは、FIFO1へ、奇数画素位置2値化済みデータFIFO2へ、書き込まれることになる。また、FIFO2に255画素目の2値化済みデータの書き込みが終了すると、リード開始(0)のタイミングより、0画素目の画像データの読み出しが、作像ユニット205動作
40
50

に同期したクロックにより開始される。図25に示すように、 FIFO1、 FIFO2からの2値化済みデータの読み出しへは、 FIFO1、 FIFO2からのリード動作を画素毎に切り換えることにより、セレクタ2304より出力される2値化済み画像データは、0、1、2、……7167という1ライン毎に連続したデータとなる。以降これを繰り返し、7168ライン分の2値化済みデータが、セレクタ2304より画像データ制御部203へ出力されることになる。

【0194】

以上のように、 FIFO307から読み出された2値化済データは、作像ユニット205に入力され、作像エンジン205入力データに応じたプリント処理をおこなわれることになる。

10

【0195】

以上のようにして、読み取ユニット201により読みとられたA4サイズの原稿画像データの2値化処理が行なわれ、プリント出力が得られることになる。

【0196】

さらに、読み取ユニット201より画像処理プロセッサ-204に入力される画像クロック速度に対して、画像処理プロセッサ-の動作クロックが十分に高速で、リアルタイム処理が可能な場合には、入力画像データ読み取ユニット201より読み取られた画像データをワーキングメモリ216に記憶することなく2値化処理を行なう構成としてもよい。その場合には、図3の入力レジスタ302より入力された画像データをデータ演算処理部303の内部メモリ411に取り込み、2値化処理が終了した画素位置の内部メモリを2値化データに書きかえるようにし、図12に示した1バンド分の2値化処理が終了した時点より、内部メモリの2値化済データを出力レジスタを介して、作像ユニット205に出力させるようにする構成としてもよい。

20

【0197】

または、ワーキングメモリ216に入力画像データ読み取ユニット201より読み取られた画像データを所定ライン数(128ライン以上)記憶する領域と、2値化済みデータを所定ライン数記憶する領域を設けて、それぞれに記憶する画像データを更新しながら、A4サイズの原稿画像データ全面の2値化処理を行うようにしてもよい。ただし、この場合には、図20のStep_121における、内部メモリの2値化結果を転送するワーキングメモリの転送先アドレスは、対応する入力画像データが記憶されている領域とは異なり、上記、2値化済みデータを所定ライン数記憶する領域となる。

30

【0198】

さらに、本実施の形態の誤差拡散処理における誤差データは、図16に示した隣接4画素に分配されているが、誤差拡散の方法はこれに限定されるものではなく、図26に示すような場合であっても適用可能なことは自明である。図26には誤差を配分する画素を10画素とした場合の例を示し、又 各画素位置への誤差配分率の例を示したが、これらの例に限定される物では無い。図中*印は夫々のPEが今同時に処理しようとする注目画素位置であり、誤差は該注目画素位置を含まず、注目画素位置に近い画素から10画素程度の画素に配分すれば、良好な画像が得られる。

40

【0199】

尚、図中X印は各誤差の配分演算で生じた演算誤差を吸収する画素位置であり、誤差の総和から他の画素への配分誤差の総和を減算した余り誤差を割り当てる。また、図16の誤差分配時に設けた、各PEが有する、BND間の誤差分配用のsep型変数ERRBE[0]～ERRBE[27]、ERRBO[0]～ERRBO[27]のそれぞれ $28 \times 2 = 56$ 個の変数は、図26に示すパターンで誤差を配分する場合には $56 \times 2 = 112$ 個が必要になる。

【0200】

さらには、図23に示した、注目画素位置の番号よりも大きな番号が付された任意の画素位置は、注目画素の2値化処理が終了後に2値化処理が行われる画素位置を示したおり、これらの画素位置への誤差配分が可能なことは、自明である。

50

【0201】

また、本実施の形態では、0画素目より順番に入力される主走査方向の原稿画像データを奇数番目と偶数番目に振り分ける処理を図17に示したFIFOメモリ301にて行っているが、奇数番目と偶数番目画素の振り分けを行う手段は、これに限定されるものではない。たとえば、ワーキングメモリ216への原稿画像データの入力は、第1の実施の形態にて説明した図7に示した形式で行い、ワーキングメモリ216より、内部メモリ411へのデータ転送時に、奇数番目と偶数番目の原稿画像データの振り分けを行うようにしてもよい。

【0202】

または、ワーキングメモリ216より、内部メモリ411への転送は、0画素番目より順番に行い、内部メモリ411への転送後に、内部メモリ411の内容を操作して、奇数番目と偶数番目画素の振り分けを行うようにしてもよい。10

【0203】

または、ワーキングメモリ216より、内部メモリ411への転送は、0画素番目より順番に行い、奇数番目と偶数番目の原稿画像データの振り分けは、各PEが内部メモリ411をアクセスする際に行うようにしてもよい。

【0204】

さらには、図7の形式にてワーキングメモリ216に記憶された原稿画像データを、ワーキングメモリ216の内容を操作することにより、奇数番目と偶数番目、それぞれの並びに変更された原稿画像データの並びをワーキングメモリ216に形成するようにしてもよい。20

【0205】

同様に、奇数番目と偶数番目に振り分けられた原稿画像データの2値化処理結果を連続する並びデータに変換する手段も、図24に示したFIFOメモリ307による手段に限定されるものではない。たとえば、2値化処理結果を内部メモリ411より、ワーキングメモリ216へ転送する際に、奇数番目と偶数番目に振り分けられた原稿画像データの2値化処理結果を連続する並びデータに変換するようにしてもよいし、内部メモリ411の内容を操作して、内部メモリ411に連続する並びのデータ列を形成した後に、ワーキングメモリ216への転送を行うようにしてもよい。さらには、各PEが奇数画素と偶数画素に振り分けて処理を行った2値化結果を、連続するデータ並びとなるように、各PEが内部メモリ411にライトするようにしてもよい。30

【0206】

または、ワーキングメモリ216に転送された、奇数番目と偶数番目に振り分けられた原稿画像データの2値化処理結果をワーキングメモリ216の内容を操作することにより、連続する並びデータに変更された原稿画像データの並びをワーキングメモリ216に形成するようにしてもよい。

【0207】

このように第2の実施の形態によれば、第1の実施の形態の効果に加え、第1の実施の形態では誤差の拡散が行なえなかった注目画素の斜め下に隣接する画素(図16のA)への誤差拡散を行うことが可能となることにより、誤差拡散処理後の画像データに発生する不自然なテクスチャーを抑制することが可能となり、画像品質を向上させる効果がある。40

【0208】**(第3実施の形態)**

第3の実施の形態では、更に高画質を得るために注目画素の周辺12画素へ誤差配分を行う場合の実施の形態を説明する。

【0209】

図27に誤差の配分方向と、配分比率を示す。図中*印は夫々のPEが今同時に処理しようとする注目画素位置であり、誤差は該注目画素位置を含まず、注目画素位置に近い画素から12画素に配分する。尚、図中X印は各誤差の配分演算で生じた演算誤差を吸収す50

る画素位置であり、誤差の総和から他の画素への配分誤差の総和を減算した余り誤差を割り当てる。

【0210】

第2の実施の形態では、各P Eがそれぞれ、主走査方向に連続する2画素の処理を担当していたのに対して、第3の実施の形態では、各P Eがそれぞれ、主走査方向の連続する3画素の処理を担当することになる。

【0211】

第3の実施の形態におけるワーキングメモリ216に記憶される原稿画像データとワーキングメモリのアドレスとの関係を図28に示す。図28に示すように、ワーキングメモリ216には、主走査方向の3の倍数の画素位置データと、3の倍数+1の画素位置データ、3の倍数+2の画素位置データが、128画素ずつ順に記憶される。なお、本実施の形態では主走査方向の画素数を7196画素とした例を示している。10

【0212】

第3の実施の形態は、第1、第2の実施の形態と同様に、図28に示すワーキングメモリのアドレス'0'より原稿画像データを128画素ずつ、順番にデータ演算処理部303の内部メモリ411に転送し、内部メモリに転送された画像データを各P Eが並列に2値化処理を行うように動作する実施の形態である。

【0213】

なお、本実施の形態の詳細な処理フローは、第1の実施の形態、および、第2の実施の形態の説明より、容易に、類推可能なことより、詳細な動作説明は省略する。20

【0214】

図29に、主走査7296画素、副走査4992ラインのA4サイズの原稿画像データの2値化処理の順番と、各P Eが処理を担当する画素位置、および、各画素位置に対する処理の順番を示す。図29に示すように、図27に示した2値化誤差が分配される注目画素周辺の12画素領域の2値化処理の順番は、注目画素の2値化処理の順番より遅れた順番となっていることより、図27に示した2値化誤差の分配が行われることがわかる。

【0215】

また、第3の実施の形態では、各P Eがそれぞれ、主走査方向の連続する3画素の処理を担当しているが、これに限定されるものではなく4画素または、5画素としてもよい。

【0216】

各P Eそれぞれが処理を担当する主走査方向の連続する画素数を増加させることにより、より広範囲な画素位置への2値化誤差の分配が可能となる。30

【0217】

このように第3の実施の形態によれば、第2の実施の形態の誤差拡散領域に対して、さらに注目画素の副走査方向に隣接するラインにおける主走査方向の2画素手前の画素位置への誤差拡散が可能となる。これにより、誤差拡散領域の対称性を向上させることができなり、誤差拡散処理後の画像データのテクスチャーをより好みのものに改善することができる。これより、更なる高画質化が図れるという効果がある。

【0218】

(他の実施の形態)

さらに、本実施の形態では入力画像データを2値化する誤差拡散処理を例に説明しているが、2値化処理に限定されるものではなく、3値化、4値化等の多値誤差拡散処理にも適用可能なことは、明らかである。40

【0219】

さらには、本発明が適用される画像処理としては、誤差拡散に限定されたものではなく、各種誤差拡散法に従う改良された方式はもとより、前に行なった処理結果を利用して注目画素の処理が行なわれるいわゆる逐次処理を行なう場合にも適用可能なことは自明である。

【0220】

また、各実施形態の目的は、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50

ムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

10

【0221】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カ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込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0222】

【発明の効果】

以上説明したように本発明によれば、逐次処理を行なうための補助的な演算処理部を利用することなく SIMD 型演算処理部を用いて、例えば誤差拡散処理等の画像処理を行なうことが可能となる。さらには、ラスタースキャンによる 1 ラインの入力画像データの画素数が、 SIMD 型プロセッサーが有する演算処理部の個数以上の場合であっても、 SIMD 型プロセッサーが有する複数の演算処理部を効率よく動作させながら逐次処理が可能となる。

20

【0223】

又、良好な画質を得るために画素位置へ 2 値化誤差を補正できる為、高品位な画像が得られる。

【図面の簡単な説明】

【図 1】本発明の実施の形態にかかる画像処理装置の構成を機能的に示したブロック図である。

30

【図 2】本実施の形態にかかる画像処理装置のハードウェア構成を示したブロック図である。

【図 3】図 2 に示した画像処理プロセッサー部 204 の構成を説明するための図である。

【図 4】 SIMD 型プロセッサーを構成する演算処理ユニット 300 の構成を示したブロック図である。

【図 5】図 2 中の読み取りユニット 201 より出力される A4 原稿画像データを示した図である。

【図 6】図 3 に示した FIFO メモリ 301 の動作を説明するための図である。

【図 7】ワーキングメモリ 216 に記憶された画像データとワーキングメモリのアドレスとの関係を示した図である。

40

【図 8】図 3 のデータ演算処理部 303 における動作を示したフローチャートである。

【図 9】本実施の形態における 2 値化処理を説明するための図である。

【図 10】各 PE に対する、sep 型変数 K の値を示した図である。

【図 11】2 値化誤差の分配方法を示した図である。

【図 12】本実施の形態における 2 値化処理を説明するための図である。

【図 13】FIFO メモリ 307 の動作を説明するための図である。

【図 14】ワーキングメモリ 216 に記憶された画像データとワーキングメモリのアドレスとの関係を示した図である。

【図 15】2 値化誤差の配分方法を示した図である。

【図 16】第 2 の実施の形態における 2 値化誤差の配分方法を示した図である。

50

【図17】FIFOメモリ301の構成を示した図である。

【図18】FIFOメモリ301の動作を説明するためのタイムチャートである。

【図19】第2の実施の形態における原稿画像データとワーキングメモリのアドレスとの関係を示した図である。

【図20】第2の実施の形態における処理を示したフローチャートである。

【図21】第2の実施の形態における各PEに対する、sep型変数Kの値を示した図である。

【図22】Lの値に応じた、内部メモリ411の内容を示した図である。

【図23】Lの増加に対応した、2値化処理が行なわれる原稿画像データの配置と2値化処理の順番を示した図である。 10

【図24】FIFOメモリ307の構成を示した図である。

【図25】FIFOメモリ307の動作を説明するためのタイムチャートである。

【図26】2値化誤差の配分方法を示した図である。

【図27】第3の実施の形態における2値化誤差の配分方法を示した図である。

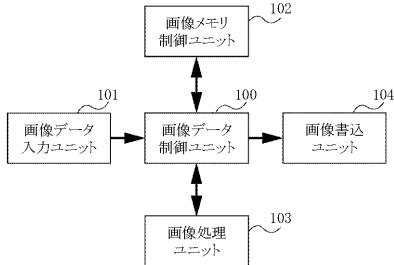
【図28】第3の実施の形態におけるワーキングメモリ216に記憶される原稿画像データとワーキングメモリのアドレスとの関係を示した図である。

【図29】第3の実施の形態における2値化処理を説明するための図である。

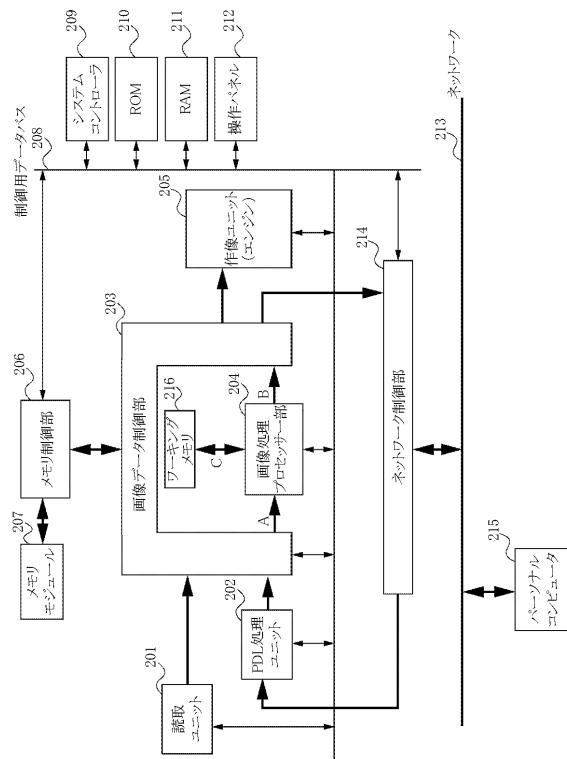
【符号の説明】

201	読み取りユニット	
202	PDL処理ユニット	20
203	画像データ制御部	
204	画像処理プロセッサー部	
205	作像ユニット	
206	メモリ制御部	
207	メモリモジュール	
208	制御用データバス	
209	システムコントローラ	
210	ROM	
211	RAM	
212	操作パネル	30
213	ネットワーク	
214	ネットワーク制御部	
215	パーソナルコンピュータ	
216	ワーキングメモリ	

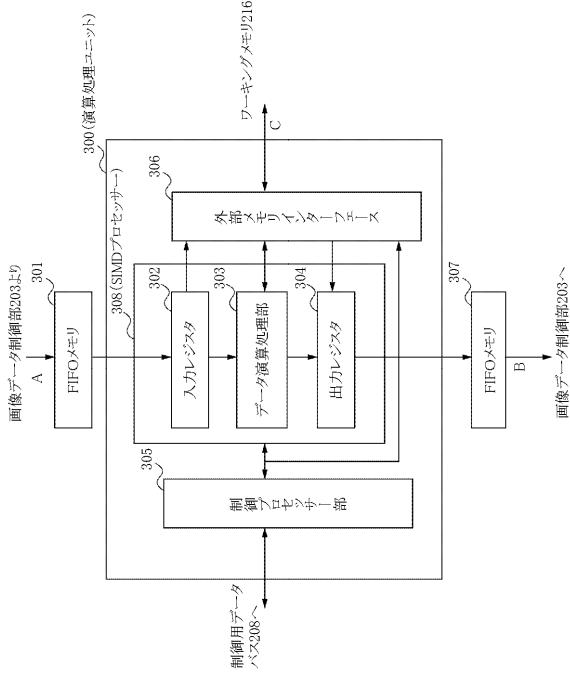
【 図 1 】



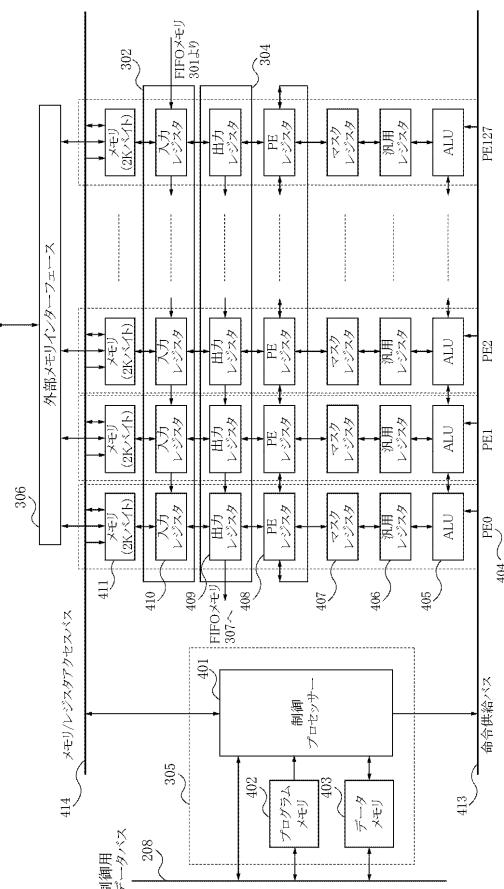
【 図 2 】



【 四 3 】



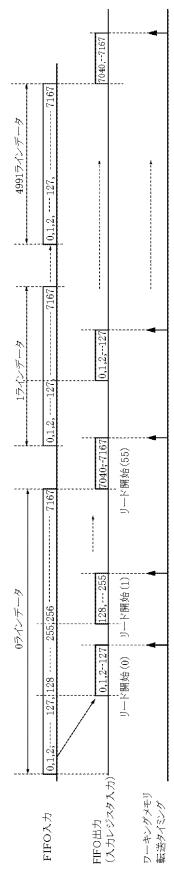
【 図 4 】



【図5】

画像信号					
主走査方向					
(0,0)	(0,1)	(0,2)	(0,126)	(0,127)
(1,0)	(1,1)	(1,2)	(1,126)	(1,127)
(2,0)	(2,1)	(2,2)	(2,126)	(2,127)
(3,0)	(3,1)	(3,2)	(3,126)	(3,127)
⋮	⋮	⋮	⋮	⋮	⋮
(4989,0)	(4989,1)	(4989,2)	(4989,126)	(4989,127)
(4990,0)	(4989,1)	(4989,2)	(4990,126)	(4990,127)
(4991,0)	(4989,1)	(4989,2)	(4991,126)	(4991,127)

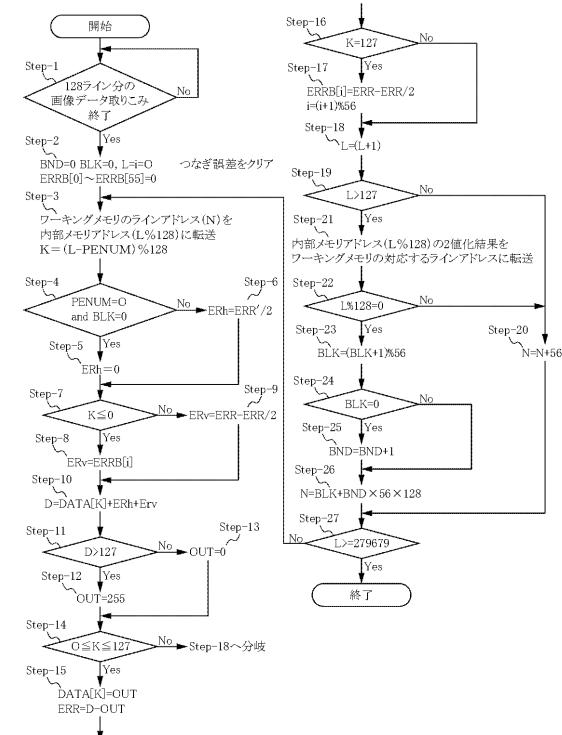
【図6】



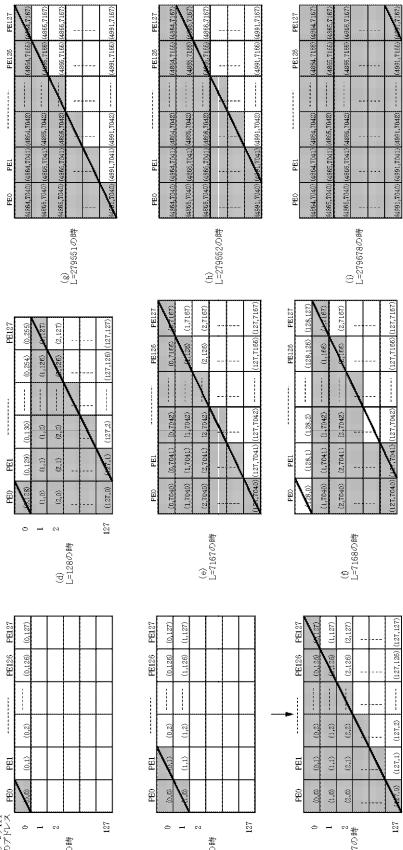
【図7】

ワーキングメモリラインアドレス					
0	(0,0)	(0,1)	(0,2)	...	(0,126)
1	(0,128)	(0,129)	(0,130)	...	(0,254)
2	(0,256)	(0,1)	(0,2)	...	(0,126)
⋮	⋮	⋮	⋮	⋮	⋮
54	(0,6912)	(0,6913)	(0,6914)	...	(0,7038)
55	(0,7040)	(0,7041)	(0,7042)	...	(0,7166)
56	(1,0)	(1,1)	(1,2)	...	(1,126)
57	(1,128)	(1,129)	(1,130)	...	(1,254)
58	(1,256)	(1,257)	(1,258)	...	(1,382)
⋮	⋮	⋮	⋮	⋮	⋮
110	(1,6912)	(1,6913)	(1,6914)	...	(1,7038)
111	(1,7040)	(1,7041)	(1,7042)	...	(1,7166)
⋮	⋮	⋮	⋮	⋮	⋮
7112	(127,0)	(127,1)	(127,2)	...	(127,126)
7113	(127,128)	(127,129)	(127,130)	...	(127,254)
7114	(127,256)	(127,257)	(127,258)	...	(127,382)
⋮	⋮	⋮	⋮	⋮	⋮
7166	(127,6912)	(127,6913)	(127,6914)	...	(127,7038)
7167	(127,7040)	(127,7041)	(127,7042)	...	(127,7166)
⋮	⋮	⋮	⋮	⋮	⋮
279496	(4991,0)	(4991,1)	(4991,2)	...	(4991,126)
279497	(4991,128)	(4991,129)	(4991,130)	...	(4991,254)
279498	(4991,256)	(4991,257)	(4991,258)	...	(4991,382)
⋮	⋮	⋮	⋮	⋮	⋮
279550	(4991,6912)	(4991,6913)	(4991,6914)	...	(4991,7038)
279551	(4991,7040)	(4991,7041)	(4991,7042)	...	(4991,7166)
⋮	⋮	⋮	⋮	⋮	⋮

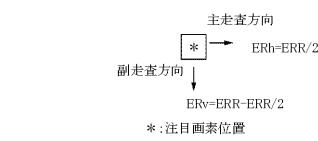
【図8】



【図9】



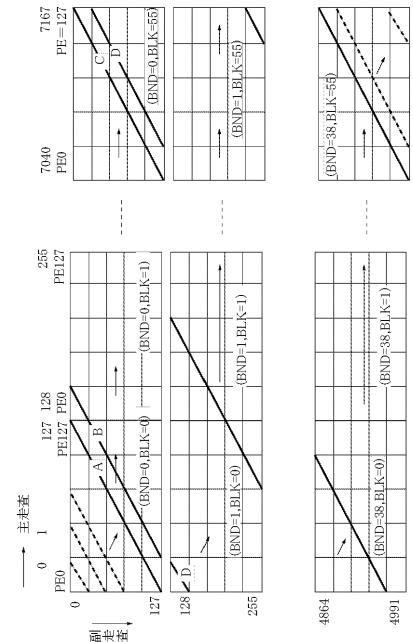
【図11】



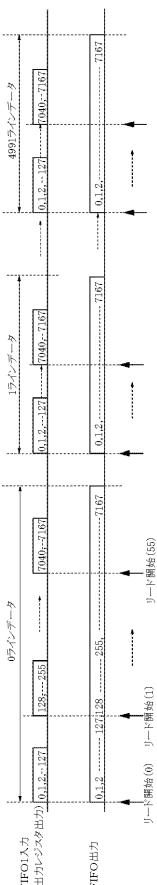
【図10】

	PE0	PE1	PE2	PE12	PE13	PE14	PE15	PE16	PE17
0	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
1	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
2	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120	0.120
L=128の時									
(a)									
(b)									
(c)									
(d)									
(e)									
(f)									
(g)									
(h)									
(i)									
(j)									
(k)									
(l)									

【図12】



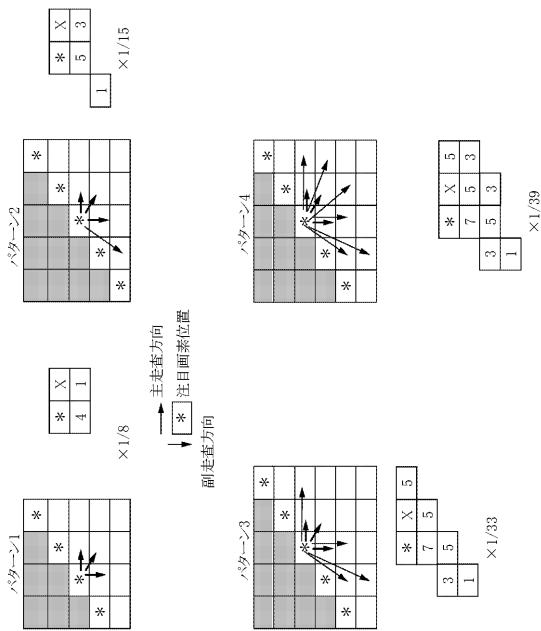
【図 1 3】



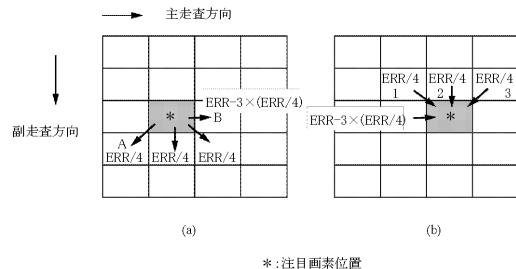
【図 1 4】

0	(0,0)	(0,1)	(0,2)	...	(0,126)	(0,127)
1	(1,0)	(1,1)	(1,2)	...	(1,126)	(1,127)
2	(2,0)	(2,1)	(2,2)	...	(2,126)	(2,127)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
4991	(4991,0)	(4991,1)	(4991,2)	...	(4991,126)	(4991,127)
4992	(0,128)	(0,129)	(0,130)	...	(0,254)	(0,255)
4992	(1,128)	(1,129)	(1,130)	...	(1,254)	(1,255)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
9983	(4991,128)	(4991,129)	(4991,130)	...	(4991,254)	(4991,255)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
274560	(0,7040)	(0,7041)	(0,7042)	...	(0,7166)	(0,7167)
274561	(1,7040)	(1,7041)	(1,7042)	...	(1,7166)	(1,7167)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
279551	(4991,7040)	(4991,7041)	(4991,7042)	...	(4991,7166)	(4991,7167)

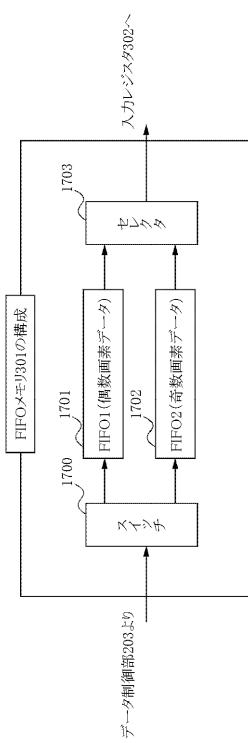
【図 1 5】



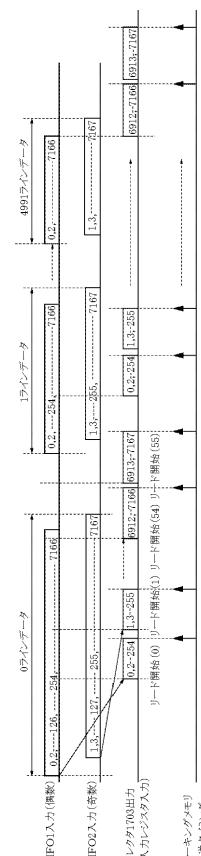
【図 1 6】



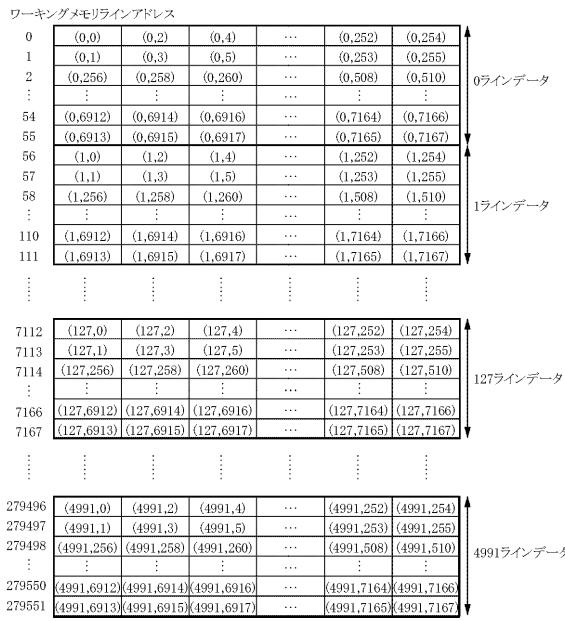
【図 17】



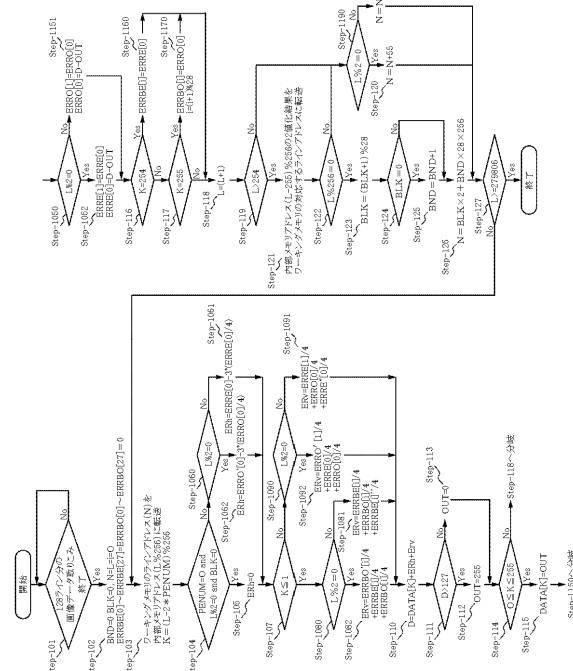
【図18】



【図19】



【図20】

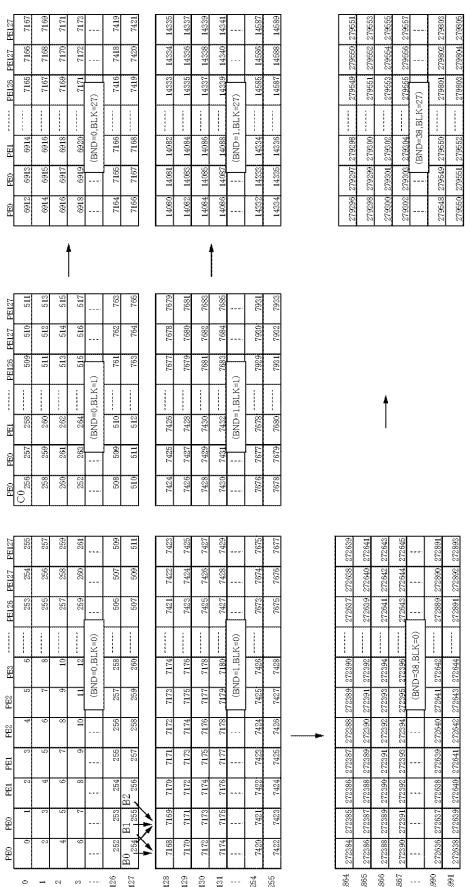


【 図 2 1 】

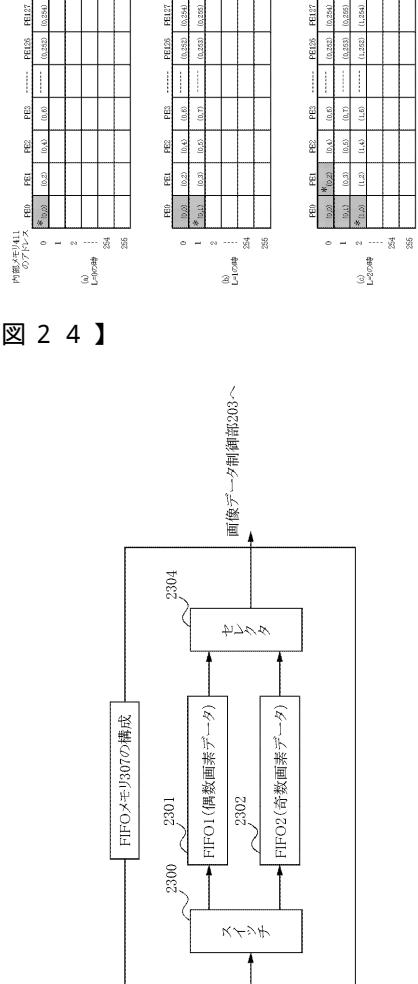
L		PE0	PE1	PE2	PE126	PE127
0	K=	0	-2	-4	-252	-254
1	K=	1	-1	-3	-251	-253
2	K=	2	0	-2	-250	-252
:	K=	:	:	:	:	:
:	K=	:	:	:	:	:
253	K=	253	251	249	1	-1
254	K=	254	252	250	2	0
255	K=	255	253	251	3	1
256	K=	0	254	252	4	2
:	K=	:	:	:	:	:
511	K=	255	253	251	3	1
512	K=	0	254	252	4	2
:	K=	:	:	:	:	:
7167	K=	255	253	251	3	1
7168	K=	0	254	252	4	2
7169	K=	1	255	253	5	3
:	K=	:	:	:	:	:
279804	K=	252	250	248	0	254
279805	K=	253	251	249	1	255

【 図 2 2 】

【 図 2 3 】

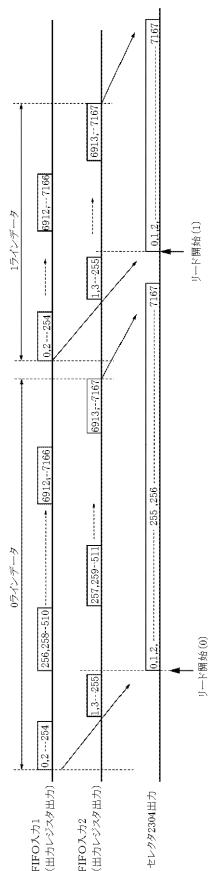


【 図 2 4 】

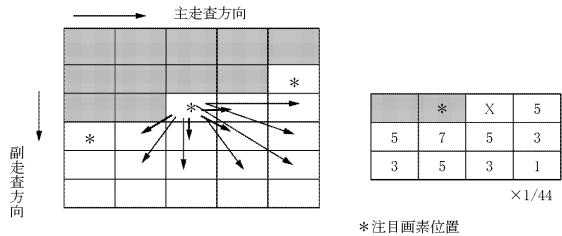


* は並列処理される画素位置を示す。

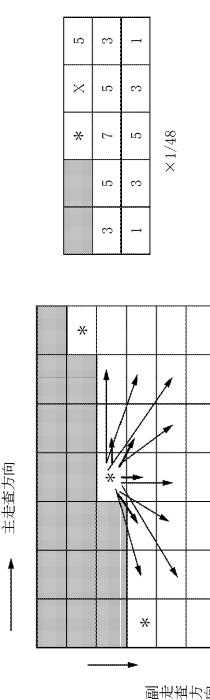
【図25】



【図26】



【図27】



【図28】

ワーニングメモリラインアドレス						
0	(0,0)	(0,3)	(0,6)	...	(0,378)	(0,381)
1	(0,1)	(0,4)	(0,7)	...	(0,379)	(0,382)
2	(0,2)	(0,5)	(0,8)	...	(0,380)	(0,383)
:	:	:	:	...	:	:
55	(0,6913)	(0,6916)	(0,6919)	...	(0,7291)	(0,7294)
56	(0,6914)	(0,6917)	(0,6920)	...	(0,7292)	(0,7295)
57	(1,0)	(1,3)	(1,6)	...	(1,378)	(1,381)
58	(1,1)	(1,4)	(1,7)	...	(1,379)	(1,382)
59	(1,2)	(1,5)	(1,8)	...	(1,380)	(1,383)
:	:	:	:	...	:	:
111	(1,6913)	(1,6916)	(1,6919)	...	(1,7291)	(1,7294)
112	(1,6914)	(1,6917)	(1,6920)	...	(1,7292)	(1,7295)
...

0ラインデータ
1ラインデータ
4991ラインデータ

【図29】

フロントページの続き

(72)発明者 竹林 学
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

審査官 渡邊 聰

(56)参考文献 特開2001-126057(JP,A)
特開2001-169106(JP,A)
特開2001-186290(JP,A)
特開2001-188901(JP,A)

(58)調査した分野(Int.Cl., DB名)
H04N 1/405