

(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 I

H04N 1/40

B

請求項の数 37 (全 45 頁)

(21) 出願番号	特願2002-6025 (P2002-6025)	(73) 特許権者	000001007
(22) 出願日	平成14年1月15日(2002.1.15)		キヤノン株式会社
(65) 公開番号	特開2003-208607 (P2003-208607A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成15年7月25日(2003.7.25)	(74) 代理人	100090538
審査請求日	平成16年12月14日(2004.12.14)		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にそれぞれ供給し、各要素プロセッサ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を有する演算手段は、N個の要素プロセッサ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、各要素プロセッサは、隣接する要素プロセッサが算出した誤差値に基づいて各要素プロセッサに入力さ

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を有する演算手段は、N 個の要素プロセッサ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、各要素プロセッサは、少なくとも直前に該要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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 S I M D 型プロセッサ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にそれぞれ供給し、各要素プロセッサ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、各要素プロセッサに対し、隣接する要素プロセッサ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、各要素プロセッサに対し、少なくとも直前に該要素プロセッサ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】

50

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 $N \times M$ (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にそれぞれ供給し、各要素プロセッサ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 16】

前記 N 個の要素プロセッサのそれぞれは、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を有する演算手段は、N 個の要素プロセッサ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、各要素プロセッサは、隣接する要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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を有する演算手段は、N 個の要素プロセッサ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、各要素プロセッサは、少なくとも直前に該要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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を有する演算手段は、1 ラインを構成する X 画素を $N \times M$ 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 $BLK \times N \times M$ 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 $BLK \times N \times M$ 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 $N \times M$ (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にそれぞれ供給し、各要素プロセッサ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 - に対し、隣接する要素プロセッサ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 2 9】

前記 N 個の要素プロセッサ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 - に対し、少なくとも直前に該要素プロセッサ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 $N \times M$ 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 $BLK \times N \times M$ 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。

【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 BND とした場合、前記誤差記憶工程で記憶する少なくとも $BLK \times N \times M$ 個の誤差データは、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 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。

10

20

30

40

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にそれぞれ供給し、各要素プロセッサ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にそれぞれ供給し、各要素プロセッサ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 $N \times M$ (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にそれぞれ供給し、各要素プロセッサ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。

30

【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 $N \times M$ (M は自然数) ライン分記憶する記憶工程と、

40

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にそれぞれ供給し、各要素プロセッサ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。

【 0 0 0 2 】

【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 - 3 1 5 1 2 6号公報に記載されている。

【 0 0 0 3 】

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。

【 0 0 0 4 】

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2 0 0 1 - 2 7 4 9 9 3号公報に記載されている。

【 0 0 0 5 】

【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。

【 0 0 0 6 】

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る。さらには、ラスタスキャンによる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。

【 0 0 0 7 】

【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にそれぞれ供給し、各要素プロセッサ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 0 9 】

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。

【 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を有する演算手段は、N個の要素プロセッサ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、各要素プロセッサは、隣接する要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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を有する演算手段は、N個の要素プロセッサ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、各要素プロセッサは、少なくとも直前に該要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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にそれぞれ供給し、各要素プロセッサ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、各要素プロセッサに対し、隣接する要素プロセッサ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、各要素プロセッサに対し、少なくとも直前に該要素プロセッサ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BLKとした場合、少なくともBLK×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BNDとした場合、前記誤差記憶工程で記憶する少なくともBLK×N個の誤差データは、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。

【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SIMD型プロセッサ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にそれぞれ供給し、各要素プロセッサ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を有する演算手段は、N個の要素プロセッサ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、各要素プロセッサは、隣接する要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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を有する演算手段は、N個の要素プロセッサ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、各要素プロセッサは、少なくとも直前に該要素プロセッサが算出した誤差値に基づいて各要素プロセッサ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】

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を有する演算手段は、1ラインを構成するX画素を $N \times M$ 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 $BLK \times N \times M$ 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 $BLK \times N \times M$ 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 $N \times M$ (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にそれぞれ供給し、各要素プロセッサ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。

【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へ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である。

50

【 0 0 4 0 】

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・コントローラー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。

【 0 0 4 1 】

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。

【 0 0 4 2 】

システム・コントローラー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

【 0 0 4 3 】

画像メモリ制御ユニット102によりおこなわれる処理は以下である。

【 0 0 4 4 】

システム・コントローラーとのインターフェース制御処理、メモリ部への書き込み・読み出し処理、メモリ・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

【 0 0 4 5 】

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。

【 0 0 4 6 】

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。

【 0 0 4 7 】

画像書込ユニット104によりおこなわれる処理は以下である。

【 0 0 4 8 】

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

【 0 0 4 9 】

（デ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。

【 0 0 5 0 】

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、読取ユニット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ー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

【 0 0 5 1 】

上記した構成のうち、画像処理プロセッサ204は、画像に基づいて作成されたデ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、読取ユニット201、PDL処理ユニット202、画像処理プロセッサ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204に関するもので、画像処理プロセッサ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、画像メモリ制御部206に接続されるメモリ・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、読取ユニット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、メモリ制御部206およびメモリ・モジュール207により画像メモリ制御ユニット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、読取ユニット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取ユニット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、読取ユニット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ヘ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、読取ユニット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】

読取ユニット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、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。

【0063】

次に、読取ユニット 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部 204 に出力する。画像処理プロセッサ部 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。

【0064】

上記動作例は、読取ユニット 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、処理済み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。

【0066】

読取ユニット 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部 204 に出力される。画像処理プロセッサ部 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 画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、読取ユニット 201 を 1 回だけ動作

10

20

30

40

50

させ、読取ユニット 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画像ユニット 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、画像読込、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 308 と、制御プロセッサ部 305、外部メモリーインターフェース 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、読取ユニット 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ーであり、書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、外部メモリーインターフェース 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、外部メモリーインターフェース 306 に出力される。尚、外部メモリーインターフェース 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ーである。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 308 及び外部メモリーインターフェース 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部 305 と接続されている。制御プロセッサ部 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ー、レジスタへのデータの入出力等の制御及び外部メモリーインターフェース部 306 を制御し、SIMD プロセッサ 308 の内部メモリーやレジスタ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、制御プロセッサ部 305 と SIMD プロセッサ 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部を示し、制御プロセッサ 401 お

10

20

30

40

50

よび、制御プロセッサ及びSIMDプロセッサ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410、2Kバイトの容量を有するメモリ411から構成される。ALU405、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ALU405、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ALU405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、ALU405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中の読取ユニット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、読取ユニット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）方向の読取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取動作が行なわれる。なお、本実施の形態では、読取ユニット２０１の読取解像度は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、読取ユニット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 10
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、読取ユニット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。

【００８１】

読取ユニット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、読取ユニット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 20
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、ＦＩＦＯメモリ３０１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 30
も連続的にＦＩＦＯメモリ３０１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。

【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画素ずつ、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、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 40
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。

【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 3 0 3 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、1 2 8 ライン分の入力画像データがワーキングメモリ 2 1 6 に取り込まれるまで待機する。ワーキングメモリ 2 1 6 に 1 2 8 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 A 4 サイズの原稿画像に相当する 4 9 9 2 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 3 0 0 にて用いられる変数としては、データ演算処理部の各 P E が並列に持つ変数と、並列に持たない変数がある。各 P E が並列に持つ変数とは、各 P E 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 P E 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 3 0 3 を構成する P E の個数と等しい 1 2 8 個が設けられることになる。各 P E が並列に持たない変数を用いた演算を各 P E が並列に行なう場合には、各 P E が共通の変数値を参照するのに対して、各 P E が並列に持つ変数を用いた演算を各 P E が並列に行なう場合には、変数名が同一だが、各 P E がそれぞれ異なる変数値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)、B N D (バンド)、B L K (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 E R R B [0] ~ E R R B [5 5] (つなぎ誤差データ 5 6 個)の初期化を行なう。ここで、B N D (バンド)とは、入力画像データの副走査方向の 1 2 8 ラインを 1 バンドとする変数、B L K (ブロック)とは、入力画像データの主走査方向の 1 2 8 画素を 1 ブロックとする変数であり、A 4 サイズの原稿画像データ；主走査画素数 = 7 1 6 8、副走査ライン数 = 4 9 9 2 に対しては、B N D は、0 ~ 3 8、B L K および i は、0 ~ 5 5 の値をとる。また、sep 型変数 E R R B [0] ~ E R R B [5 5] 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 P E それぞれにつき 5 6 個が設けられている。5 6 個とは上述した、主走査方向に配置される B L K 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 B L K 間にて各 P E 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 2 1 6 のラインアドレス N の 1 2 8 画素分のデータをデータ演算処理部 3 0 3 の各 P E が有するメモリ 4 1 1 に転送する。L % 1 2 8 は、L を 1 2 8 で割り算した余りの数値を示し、L を 1 2 8 で割り算した余りがメモリ 4 1 1 の転送先アドレスとなる。L を 1 2 8 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 2 1 6 よりメモリ 4 1 1 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 1 2 8 画素分の画像データが、各 P E が有するメモリ 4 1 1 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 P E N U M) を 1 2 8 で割り算した余りを代入する。この演算は各 P E にて並列実行されるものである。P E N U M は、プロセッサエレメント P E 0 に対して 0、P E 1 に対して 1、・・・、P E 1 2 7 に対して 1 2 7 が割り当てられたシステム定数であり、各 P E に対応する sep 型変数 K の値は、L の値に応じて図 1 0 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 P E において並列に、P E N U M = 0、かつ B L K = 0 か否かが判定され、P E N U M = 0、かつ B L K = 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。P E N U M = 0 は、プロセッサエレメント P E 0 に割り当てられたシステム定数であり、P E 0 以外のプロセッサエレメントの P E N U M が、

10

20

30

40

50

0' と等しくなることはなく、PE 1 ~ PE 127 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。

【0091】

Step_5 では、プロセッサエレメントPE 0 に対応したsep型変数ERh に0 が代入される。Step_6 では、PE 1 ~ PE 127 のプロセッサエレメントに対応したsep型変数ERh にsep型変数ERR' / 2 を代入する。ここで、変数ERh 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ERv は、副走査方向への誤差分配値を示す。図11 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。ERR 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、ERR の1 / 2 が、主走査方向への分配誤差ERh となり、副走査方向への分配誤差ERv は、ERR - ERR / 2 となる。また、step_6 に示した変数ERR' はデータ演算処理部303 中の各PE について、左に隣接するPE が有する変数ERR と同一変数を示し、PE 1 に対しては、PE 0 の変数ERR、PE 127 に対しては、PE 126 の変数ERR に相当する。さらに、PE 0 に対しては、PE 127 の変数ERR 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PE レジスタに変数ERR の値を設定することにより、全てのPE が同時に隣接するPE の変数ERR を参照することにより実行される。

10

【0092】

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ERh 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ERh の算出が行なわれる。

20

【0093】

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ERv にERRB[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、ERR - ERR / 2 が代入される。ここで、ERRB[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PE が有するメモリ411 に設けられている。ここで、BND = 0 かつK = 0 となるPE 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ERv を' 0 ' として、画像端部の2 値化処理を行うようにしてもよい。また、BND が' 0 ' でない場合には、後述するStep_17 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ERv に設定されることになる。

30

【0094】

Step_9 では、各PE が自身のプロセッサエレメントに属するsep型変数ERR 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ERv を算出する。

【0095】

次にStep_10 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 (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

【0096】

次に、Step_11 にて、D が127 よりも大きいかが判別され、大きい場合には、Step_12 にて、sep型変数OUT に255 を、大きくない場合には、Step_13 にてOUT に0 が代入される。Step_12、Step_13 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、255 に2 値化処理される。

【0097】

Step_14 では、sep型変数K が0 以上かつ127 以下か否かが判別され、Step_15 また

50

はStep_1 8 に分岐する。変数 K が 0 以上 1 2 7 以下の範囲でない P E が算出した 2 値化処理結果は、Step_1 5、Step_1 6、Step_1 7 が実行されないことにより破棄されることになる。

【 0 0 9 8 】

Step_1 5 では、前述したメモリ 4 1 1 の D A T A [K] の領域に 2 値化済みデータ O U T を記憶し、2 値化誤差 E R R に D - O U T を代入し、Step_1 6 に分岐する。

【 0 0 9 9 】

Step_1 6 では、K が 1 2 7 に等しいか否かが判別され、等しい場合にはStep_1 7 に、等しくない場合にはStep_1 8 に分岐する。Step_1 7 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 E R R B [i] に記憶する。さらに、i に 1 を加算した値を 5 6 で割り算した余りを i に設定する処理を行なう。

10

【 0 1 0 0 】

次に、Step_1 8 では、L に 1 を加算する。

【 0 1 0 1 】

次にフローはStep_1 9 に進み、L が 1 2 7 より大きいかが判別され、大きくない場合には Step_2 0 に進み N に 5 6 が加算され、Step_2 7 を介してStep_3 に分岐する。

【 0 1 0 2 】

ここまでの処理では、図 1 0 に示すとおり、sep型変数 K が 0 以上となるのは、P E 0 に対応した変数のみであるので、Step_1 5 では、P E 0 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 E R R に関しても、P E 0 に対応する変数 E R R にはStep_1 5 にて 2 値化誤差が記憶されている。次にStep_2 0 からStep_2 7 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 3 0 3 のメモリ 4 1 1 に転送されるデータは、図 9 の (B b) 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、P E 0 では、K = 1、P E 1 では、K = 0 となる。(図 1 0 参照)。

20

【 0 1 0 3 】

Step_4 の判定において、P E 1 に対するフローはStep_6 に分岐する。P E 1 の処理では、Step_6 にて直前のStep_1 5 が実行された際に P E 0 の E R R に記憶された 2 値化誤差、すなわち、P E 0 が入力データ (0 、 0) を 2 値化した際に生じた 2 値化誤差 E R R を E R R ' として参照し、入力データ (0 、 1) に対する誤差補正值 E R h を算出する。

30

【 0 1 0 4 】

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

【 0 1 0 5 】

次に、Step_7 では、P E 0 に対応する変数 K が ' 1 ' となっていることより、P E 0 に対するフローは、Step_9 に分岐する。Step_9 では、P E 0 に対応した sep 型変数 E R R を参照し、E R v を算出する。このときに P E 0 が使用する E R R の値は、前回のStep_1 5 の処理にて、P E 0 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 P E 0 が算出した E R v 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

【 0 1 0 6 】

また、P E 1 に対する変数 K は ' 0 ' となっていることより、P E 1 に対するフローは、Step_7 よりStep_8 に進む。ここで、P E 1 に対応する sep 型変数 i = 0、E R R B [i] = 0 により E R v には、初期設定値である ' 0 ' が設定されることになる。

【 0 1 0 7 】

以降フローが進みStep_1 4 の判定にて 0 以上 1 2 7 以下の変数 K を有するプロセッサエレメントは、P E 0、と P E 1 ということになり、Step_1 5 にて、P E 0、P E 1 が算出した 2 値化結果が記憶されている変数 O U T 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 PE_0 、 PE_1 の 2 個のプロセッサエレメント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_18 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エレメントの数も増加していくことになる。

【0109】

次に $L = 127$ となって、Step_27 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 $PE_0 \sim PE_{127}$ まで全ての PE に対して 0 以上 127 以下の値となり、全ての PE 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 PE 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_16 に至ると PE_0 のプロセッサエレメントに対する変数 K が 127 と等しいことにより、 PE_0 に対する処理は、Step_16 より Step_17 に分岐し、Step_17 の処理が実行される。Step_17 では、 PE_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 $ERRB[i]$ に設定する。ここでは、 i の値は、'0' となっていることより $ERRB[0]$ に 2 値化誤差の副走査方向への分配成分が設定され、 PE_0 に対応する sep 型変数 i には 1 が加算される。

【0111】

次に、Step_18 より Step_19 に至ると、Step_19 より Step_21 に進む。

【0112】

Step_21 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 PE 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。

【0113】

次にフローは Step_22 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_23 に進む。Step_23 にて BLK は $BLK = 1$ となり、Step_24 より、Step_26 に分岐する。Step_26 では、 N に $BLK + BND \times 56 \times 128$ の値が代入される。この場合は、 $N = 1$ となる。そして Step_26 より Step_27 を介して Step_3 にフローは分岐し、フローを繰り返す。

【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 PE の変数 K の値より、この場合に各 PE 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 PE_0 が担当する

10

20

30

40

50

画素データ (0、1 2 8) 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。

【 0 1 1 7 】

上記のような動作によりフローを繰り返す。

【 0 1 1 8 】

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) のライン上、別の図示では、図 1 2 中の C で示したライン上の 1 2 8 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 1 2 8 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。

【 0 1 1 9 】

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 1 2 8 ラインの先頭から 1 2 8 画素が設定されていることになる。

【 0 1 2 0 】

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 1 2 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 値化誤差に相当するものである。

【 0 1 2 1 】

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。

【 0 1 2 2 】

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。

【 0 1 2 3 】

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。

【 0 1 2 4 】

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、 $L = (2\ 7\ 9\ 6\ 7\ 8 + 1) = 2\ 7\ 9\ 6\ 7\ 9$ となっていることよりフローを終了する。

【0 1 2 5】

以上のようにして、A 4原稿画像1面分である7 1 6 8 × 4 9 9 2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2 1 6に記憶されることになる。

【0 1 2 6】

ワーキングメモリメモリ2 1 6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3 0 0の外部メモリインターフェース3 0 6を介して出力レジスタ3 0 4に入力される。次に、ワーキングメモリメモリ2 1 6より読み出された2値化済データは、出力レジスタ3 0 4からF I F Oメモリ3 0 7に書き込まれる。F I F O 3 0 7へのデータ書込みと読み出しの関係を図1 3に示す。

【0 1 2 7】

F I F Oメモリ3 0 7は、1ライン分(7 1 6 8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3 0 0より出力されるクロックにより出力レジスタ3 0 4より出力された画像データをF I F Oメモリ3 0 7にライトする。リードクロックは、作像エンジン2 0 5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、F I F Oメモリ3 0 7へのライトクロックが1 0 0 M H z、リードクロックは、2 0 M H z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。F I F Oメモリ3 0 7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1 3に示したように、F I F O 3 0 7への2値化済データの書込みは、1 2 8画素単位で行なわれる。また、読み出しは、作像ユニット2 0 5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。

【0 1 2 8】

F I F O 3 0 7から読み出された2値化済データは、作像ユニット2 0 5に入力され作像エンジン2 0 5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。

【0 1 2 9】

以上のようにして読取ユニット2 0 1により読みとられたA 4サイズ of 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。

【0 1 3 0】

本実施の形態では、読取ユニット2 0 1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5 6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1 4に示すように、主走査方向の先頭より、1 2 8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。

【0 1 3 1】

さらに、読取ユニット2 0 1より画像処理プロセッサ - 2 0 4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、入力画像データ読取ユニット2 0 1より読み取られた画像データをワーキングメモリ2 1 6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3 0 2より入力された画像データをデータ演算処理部3 0 3の内部メモリ4 1 1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1 2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2 0 5に出力させるようにする構成としてもよい。

【0 1 3 2】

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1 1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1 5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1 5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

10

20

30

40

50

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P E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E R R Bは各P E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S I M D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、S I M D型演算部が有すプロセッサ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、S I M D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、各プロセッサエレメントを並列動作させることが可能となり、S I M D型プロセッサ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F I F O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、F I F Oメモリ301の構成を示し、図18にF I F O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クタ、1701はセクタ1700より出力された偶数番目の画素データが入力されるF I F O1メモリ、1702はセクタ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F I F O2メモリ、1703はF I F O1メモリ1701、F I F O2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クタであり、セクタ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F I F O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 ERRO[0] ~ ERRO[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]、ERRO[0] ~ ERRO[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 P E 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 P E が有効な 2 値化処理を行い、これ以外の、K の値を有する P E 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 P E 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 P E 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 P E は、各 B N D の先頭ラインの偶数画素位置、K=1 の P E は、各 B N D の先頭ラインの奇数画素位置、K=254 の P E は、各 B N D の最終ラインの偶数画素位置、K=255 の P E は、各 B N D 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ERRE[0]、ERRE[1]、ERRO[0]、ERRO[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ERRE[0]、ERRE[1]、ERRO[0]、ERRO[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。ERRE[0]、ERRE[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、ERRE[0]には、直前に処理された偶数画素データの2値化誤差、ERRE[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、ERRO[0]、ERRO[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、ERRE'[0]、ERRE'[1]は、各PEに対して左に隣接したPEが有するERRE[0]、ERRE[1]に相当し、ERRE''[0]は、右に隣接したPEが有するERRE[0]、ERRE[1]に相当するものである。同様に、ERRO'[0]、ERRO'[1]は、各PEの左に隣接するPEが有するERRO[0]、ERRO[1]に相当し、ERRO''[0]、ERRO''[1]は、それぞれ右に隣接するPEが有するERRO[0]、ERRO[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

40

【0151】

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_1 0 8 1、Step_1 0 8 2で算出される ER_v 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_1 0 9 2における、 $ERRO'[1]/4$ を'0'に、さらに、Step_1 0 8 2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_1 1 0にてsep型変数DATA[K]に隣接画素が発生した2値化誤差 ER_h 、 ER_v 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_1 1 1、Step_1 1 2、Step_1 1 3にて2値化処理が行われ、Step_1 1 4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_1 1 8に分岐し、2値化結果は破棄される。Step_1 1 5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_1 1 5 0に分岐する。

【0154】

Step_1 1 5 0では、Lが奇数か偶数かが判別され、奇数の場合には、Step_1 1 5 1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_1 1 5 2に進み、 $ERRE[1]$ に $ERRE[0]$ を代入し、 $ERRE[0]$ 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_1 1 5 1、Step_1 1 5 2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_1 1 6にてKが254か否かが判定され、254に等しい場合にはStep_1 1 6 0に分岐し、2値化誤差 $ERRE[0]$ を $ERRBE[i]$ に代入し、Step_1 1 8に分岐する。Step_1 1 7では、Kが255か否かが判定され、255に等しい場合にはStep_1 1 7 0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_1 1 8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_1 1 6 0、Step_1 1 7 0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_1 1 8にて、Lに1が加算され、Step_1 1 9にて、Lが254よりも大きいかが判別され、大きくない場合にはStep_1 1 9 0に分岐する。Step_1 1 9 0では、Lが奇数か偶数かが判別され、偶数の場合にはStep_1 2 0へ、奇数の場合にはStep_1 1 9 1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_1 0 3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_1 2 7に分岐し、さらに、Step_1 0 3に進んでフローを繰り返す。

【0158】

ここで、Step_1 0 2における初期化以降の動作より、Step_1 0 3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 2 2 は、L の値に応じて内部メモリ 4 1 1 の内容を示した図面である。図 2 2 (a) は L = 0 の時の場合を示す。初めて、Step_1 0 3 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 1 2 8 画素のデータが、内部メモリ 4 1 1 のアドレス 0 に転送される。また、図 2 1 に示すように、L=0 の場合に 0 以上 2 5 5 以下の K の値を有数 P E は、P E 0 のみとなるので、L=0 の場合には P E 0 のみが有効な 2 値化処理を行うことになる。また、P E 0 に対しては、Step_1 0 5 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 E R h は ' 0 ' となり、さらに、Step_1 0 8 2 には図示していないが、前述したように先頭 B N D 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_1 0 8 2 にて E R v も ' 0 ' とされる。

10

【 0 1 5 9 】

Step_1 1 0 では、P E 0 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_1 1 1、Step_1 1 2、Step_1 1 3 にて 2 値化処理が行われる。さらに、Step_1 1 5 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_1 1 5 2 にて、2 値化誤差が、E R R E [0] に記憶される。

【 0 1 6 0 】

以上のようにして、図 2 2 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 4 1 1 のアドレス 0 に記憶されることになる。次に、Step_1 1 8 にて、L = 1 となり、フローを進むと、フローは、Step_1 1 9 1 に至る。Step_1 1 9 1 では、N に 1 が加算され、フローを進む。

20

【 0 1 6 1 】

このようにして、再度 Step_1 0 3 を実行した際の内部メモリ 4 1 1 の様子を図 2 2 の (b) に示す。N=1 となっていることより、図 1 9 のワーキングメモリのラインアドレス 1 の 1 2 8 画素データが内部メモリに転送される。図 2 1 より、この場合にも、有効な K の値を有する P E は、P E 0 のみであることより、P E 0 のみが有効な 2 値化処理を行うことになる。Step_1 0 3 よりフローが進むと、Step_1 0 4 より Step_1 0 6 0 をへて、Step_1 0 6 1 にフローは進む。ここでは、E R h に、E R R E [0] - 3 * (E R R E [0] / 4) が代入される。ここで、参照される E R R E [0] とは、P E 0 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 1 6 (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_1 1 5 にて、P E 0 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_1 1 5 1 にて、2 値化誤差が、E R R O [0] に記憶される。

30

【 0 1 6 2 】

Step_1 1 8 にて、L に 1 が加算され、L=2 となり、Step_1 1 9、Step_1 1 9 0 を経て Step_1 1 2 0 にフローが至る。ここでは、N に 5 5 が加算され、N=5 6 となり、フローが進む。次に Step_1 0 3 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=5 6 の 1 2 8 画素データが、内部メモリ 4 1 1 のアドレス 2 に転送される。さらに、図 2 1 に示すように、P E 1 の変数 K が 0 となり、有効な 2 値化処理を行う P E として、P E 1 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 P E の数が増加し、L=2 5 4 の場合には、P E 0 ~ P E 1 2 7 のすべての P E 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、P E 0 ~ P E 1 2 7 は、同時動作を行うプロセッサエレメントであることより、L=2 5 4 以降は、P E 0 ~ P E 1 2 7 すべての P E が同時に有効な 2 値化処理を行うように動作する。

40

【 0 1 6 3 】

ここで、図 2 3 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 P E 0 ~ P E 1 2 7 の下方向に示される原稿画像データは、対応する同一の P E 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、B N D と B L K の関係も示している。各 B N D は、原稿画像データの 1 2 8 ライン

50

に対応し、各 B L K は、主走査画素数 2 5 6 画素に対応したものとなる。

【 0 1 6 4 】

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。

【 0 1 6 5 】

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。

【 0 1 6 6 】

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

【 0 1 6 7 】

以下同様に、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 2 3 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 2 3 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=2 5 4 以降は、1 2 8 個の P E がすべて有効な 2 値化処理を行うことになる。

【 0 1 6 8 】

ここで、図 2 3 中の L=5 の時に処理される画像データに対する 2 値化誤差の分配に関して詳細に説明する。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 1 6 (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

【 0 1 6 9 】

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_1 0 6 1 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 2 3 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_1 0 6 1 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_1 1 5 2 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

【 0 1 7 0 】

さらに Step_1 0 9 1 では、図 2 3 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。

【 0 1 7 1 】

ここで、E R R E [1] とは、図 2 3 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_1 1 5 2 にて設定されたものである。また、E R R O [0] は、図 2 3 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_1 1 5 1 にて設定されたものであり、さらに、E R R E ' ' [0] は、図 2 3 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_1 1 5 1 にて設定されたものである。

40

【 0 1 7 2 】

上記の関係は、図 2 3 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 2 3 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。

【 0 1 7 3 】

次に図 2 3 中の番号 ' 6 ' を付した画素位置データに対する 2 値化誤差の分配に関して説明する。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。

【 0 1 7 4 】

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_1 0 6 2 にて、

$$ERh = ERRO' [0] - 3 * (ERRO' [0] / 4)$$
 として算出される。図 2 3 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 P E の左隣接 P E が、直前に行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_1 0 6 2 では、左隣接 P E 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_1 1 5 1 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。

【 0 1 7 5 】

さらにStep_1 0 9 2 では、図 2 3 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、

$$ERv = ERRO' [1] / 4 + ERRE [0] / 4 + ERRO [0] / 4$$

として算出する。

【 0 1 7 6 】

ここで、 $ERRO' [1]$ とは、図 2 3 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 P E 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_1 1 5 1 にて設定されたものであり、 $ERRE [0]$ は、図 2 3 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 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_1 1 5 2 にて設定されたものであり、さらに、 $ERRO' [0]$ は、図 2 3 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 P E 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_1 1 5 1 にて設定されたものである。

【 0 1 7 7 】

上記の関係は、図 2 3 中の番号 ' 6 ' を付した画素位置データに対する 2 値化誤差の分配に関して説明したが、図 2 3 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。

【 0 1 7 8 】

次に $L=2 5 4$ の場合の動作に関して説明する。 $L=2 5 4$ にて、Step_1 0 3 を実行すると、図 2 1 に示すように、P E 1 2 7 の変数 K が ' 0 ' となり、P E 0 ~ P E 1 2 7 までの 1 2 8 個の P E すべてが有効な 2 値化処理を実行することになる。図 2 2 (d)、および、図 2 3 に示すように、このとき P E 1 2 7 が 2 値化を実施する画素位置は、最初の主走査ラインの 2 5 5 画素目であり、これは、原稿画像データの (0、2 5 4) の位置のデータとなる。また、P E 0 が 2 値化を実施するのは、原稿画像データの (1 2 7、0) の位置データとなる。 $L=2 5 4$ にて、フローが進むと P E 0 の sep 型変数 K が 2 5 4 と等しいことにより、P E 0 に対する処理は、Step_1 1 6 より、Step_1 1 6 0 にフローは進み、Step_1 1 6 0 にて、 $ERRBE [0]$ に $ERRE [0]$ が代入される。ここで、 $ERRE [0]$ とは、P E 0 がこのときに処理した原稿画像データの (1 2 7、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 2 3 中の B 0 で示した、 $BND=0$ 、 $BLK=0$ における最終主走査ライン中の P E 0 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。

【 0 1 7 9 】

次にフローは、Step_1 1 8 にて、 $L=2 5 4 + 1$ とされて、Step_1 1 9 より、Step_1 2

1 にフローは至る。ここで、内部メモリ 4 1 1 のアドレス (L - 2 5 5) % 2 5 6 の 1 2 8 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=2 5 5 となっていることより、内部メモリ 4 1 1 のアドレス ' 0 ' の 1 2 8 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 2 2 の (d) に示すように、この場合には、内部メモリ 4 1 1 の内部アドレス ' 0 ' の 1 2 8 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、1 2 8 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_1 2 2 より、Step_1 1 9 0 に至り、L=2 5 5 により、Step_1 1 9 1 にて、 $N = N + 1 = 7 1 1 2 + 1 = 7 1 1 3$ となり、フローを繰り返す。この場合 Step_1 0 3 では、ワーキングメモリのラインアドレス 7 1 1 3 の 1 2 8 画素データが、内部メモリのアドレス ' 2 5 5 ' に転送され、P E 0 に対する変数 K は、' 2 5 5 ' となり、以降のフローが繰り返される。図 2 2 の (e) に L=2 5 5 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、P E 0 に対する処理は、Step_1 1 7 より、Step_1 1 7 0 にフローが至りここで、E R R B O [i] (このとき $i=0$) に E R R O [0] が代入される。ここで、E R R O [0] とは、P E 0 がこのときに処理した原稿画像データの (1 2 7 、 1) の 2 値化誤差値であり、これが、E R R B O [0] に代入されることになる。これは、図 2 3 中の B 1 で示した、B N D = 0 、B L K = 0 における最終主走査ライン中の P E 0 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、P E 0 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_1 1 8 にて、 $L = 2 5 5 + 1 = 2 5 6$ となり、Step_1 2 1 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_1 2 2 より、Step_1 2 3 にフローは至る。ここで、B L K に 1 が加算され、この場合には、B L K = 1 となる。次に Step_1 2 4 により、Step_1 2 6 にて、 $N = 1 \times 2 + 0 \times 2 8 \times 2 5 6 = 2$ となり、Step_1 2 7 を介して、フローは、Step_1 0 3 に至る。

【 0 1 8 0 】

Step_1 0 3 では、L=2 5 6 、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 1 2 8 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 2 2 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 、 2 5 6) ~ (0 、 5 1 2) までの偶数が素位置の 1 2 8 画素が記憶されていることになる。また、このときに、P E 0 に対する sep 型変数 K が K=0 となっていることより、P E 0 は原稿画像データの (0 、 2 5 6) 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、P E 0 が L=2 5 6 の時に 2 値化処理を行うのは、図 2 3 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、P E 0 が処理する原稿画像位置は、B N D = 0 、B L K = 1 の先頭主走査ラインの先頭画素となる。この場合に、P E 0 が処理する画素位置に対する 2 値化誤差の配分 E R h は、Step_1 0 6 2 により P E 1 2 7 が有する 2 値化誤差メモリ E R R O ' [0] より算出される。以降フローを繰り返し、L=2 5 7 の場合の内部メモリの内容を図 2 2 (g) に示す。

【 0 1 8 1 】

また、L = 7 1 6 7 の場合の動作に関して以下に説明する。L=7 1 6 7 の場合の内部メモリの内容を図 2 2 (h) に示す。また、図 2 3 よりこの時に各 P E が処理を行う画素位置は、B N D = 0 の B L K = 2 7 のブロックとなり、P E 0 は、最終主走査ラインの奇数画素目である (1 2 7 、 6 9 1 5) 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_1 0 3 より、Step_1 1 7 にフローが至ると、P E 0 の sep 型変数 K = 2 5 5 であることより、Step_1 1 7 0 にフローが進む。この場合の P E 0 に属する sep 型変数 i は変数 B L K と等しい数値である ' 2 7 ' となっていることより、2 値化誤差 E R R O [0] が E R R B O [2 7] に記憶され、 $i = (2 7 + 1) \% 2 8 = 0$ が設定される。さらにフローが進み Step_1 2 1 にフローが至ると、L = 7 1 6 8 により、 $(L - 2 5 5) \% 2 5 6 = 1$ 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 ' 5 4 ' に転送され、フローは、Step_1 2 2 より、L = 7 1 6 8

% 2 5 6 = 0 により、Step_1 2 3 に至る。ここで、 $B L K = (27 + 1) \% 28 = 0$ となり、Step_1 2 4 より、Step_1 2 5 に至り、 $B N D = B N D + 1 = 0 + 1 = 1$ となる。また、Step_1 2 6 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_1 2 7 より Step_1 0 3 に分岐する。Step_1 0 3 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 2 2 (i) に示す。

【0182】

図 2 2 (i)、および、図 2 3 に示すように、このとき P E 0 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 $E R h$ 、および、 $E R v$ の算出しに関して以下に説明する。

10

【0183】

Step_1 0 3 より、Step_1 0 4 に至ると P E 0 のフローは、 $P E N U M = 0$ 、 $L = 7168$ 、 $B L K = 1$ により、Step_1 0 4 に至り、 $E R h = 0$ が設定される。図 2 3 に示すように、このとき P E 0 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 $E R h$ は '0' に設定される。次に Step_1 0 7 にて、 K が 1 以下か否かが判別される。図 2 1 に示すように、この場合には、 $L = 7168$ であることより、P E 0 の sep 型変数 K は $K = 0$ となっており、P E 0 のフローは Step_1 0 8 2 に至る。

20

【0184】

Step_1 0 8 2 では、

$E R v = E R R B O'[i] / 4 + E R R B E[i] / 4 + E R R B O[i] / 4$ を算出する。

【0185】

ここで、P E 0 に関する sep 型変数 i は、Step_1 1 7 0 で $i = 0$ に設定されていることにより、

$E R v = E R R B O'[0] / 4 + E R R B E[0] / 4 + E R R B O[0] / 4$

が代入される。

【0186】

ここで、 $E R R B O'[0]$ とは、P E 0 の左隣接画素の P E として、P E 1 2 7 が有する sep 型変数 $E R R B O[0]$ に相当するが、 $B L K = 0$ の場合 P E 0 は、主走査の先頭画素を処理することより $E R R B O'[0] = 0$ が代入される。また、 $E R R B E[0]$ は、図 2 3 中の B 0 の位置の画素が発生した 2 値化誤差を示し、P E 0 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_1 1 6 0 にて、 $E R R B E[0]$ に設定されていたものである。さらに、 $E R R B O[0]$ は、図 2 3 中の B 1 の位置の画素が発生した 2 値化誤差を示し、P E 0 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_1 1 7 0 にて、 $E R R B O[0]$ 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_1 0 3、Step_1 0 4 より、Step_1 0 6 0 より、Step_1 0 6 1 に至る。ここで、 $E R h$ の算出が行われる。これは、上述したのと同様にして、各 P E 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 P E の sep 型変数 $E R R E[0]$ を参照して実施される。次に、フローは、Step_1 0 7 に至る。このとき、P E 0 の sep 型変数 K は、'1' となっていることより、P E 0 の処理は、Step_1 0 7 より、Step_1 0 8 に至り、 $L = 7169$ であることより、Step_1 0 8 1 に至る。Step_1 0 8 1 では、 $E R v = E R R B E[0] / 4 + E R R B O[0] / 4 + E R R B E'[0] / 4$ を算出する。

40

【0188】

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、セクタ2304に入力され、セクタ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、F I F O 1、F I F O 2 からの 2 値化済みデータの読み出しは、F I F O 1、F I F O 2 からのリード動作を画素毎に切り換えることにより、セクタ 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、F I F O 307 から読み出された 2 値化済みデータは、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 値化済み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

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 P E 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。

【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 P E が有する、BND 間の誤差分配用の sep 型変数 E R R B E [0] ~ E R R B E [27]、E R R B O [0] ~ E R R B O [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。

40

【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ており、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

【 0 2 0 1 】

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 inputs 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。

【 0 2 0 2 】

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

【 0 2 0 3 】

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。

【 0 2 0 4 】

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

【 0 2 0 5 】

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

【 0 2 0 6 】

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。

【 0 2 0 7 】

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

【 0 2 0 8 】

(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。

【 0 2 0 9 】

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PE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PE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PE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サイズの前稿画像データの2値化処理の順番と、各PE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PE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PE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可能となり、誤差拡散処理後の画像データのテクスチャーをより好ましい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)

40

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。

【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の画素位置へ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中の読取ユニット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】F I F Oメモリ 3 0 1 の構成を示した図である。

【図 18】F I F Oメモリ 3 0 1 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 P E 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 4 1 1 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】F I F Oメモリ 3 0 7 の構成を示した図である。

【図 25】F I F Oメモリ 3 0 7 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 2 1 6 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】

2 0 1 読取ユニット

2 0 2 P D L 処理ユニット

20

2 0 3 画像データ制御部

2 0 4 画像処理プロセッサ部

2 0 5 作像ユニット

2 0 6 メモリ制御部

2 0 7 メモリモジュール

2 0 8 制御用データバス

2 0 9 システムコントローラ

2 1 0 R O M

2 1 1 R A M

2 1 2 操作パネル

30

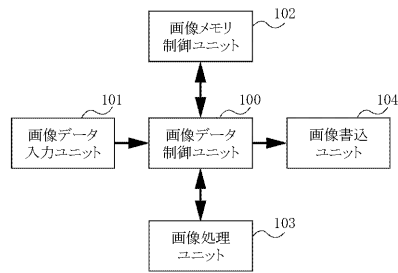
2 1 3 ネットワーク

2 1 4 ネットワーク制御部

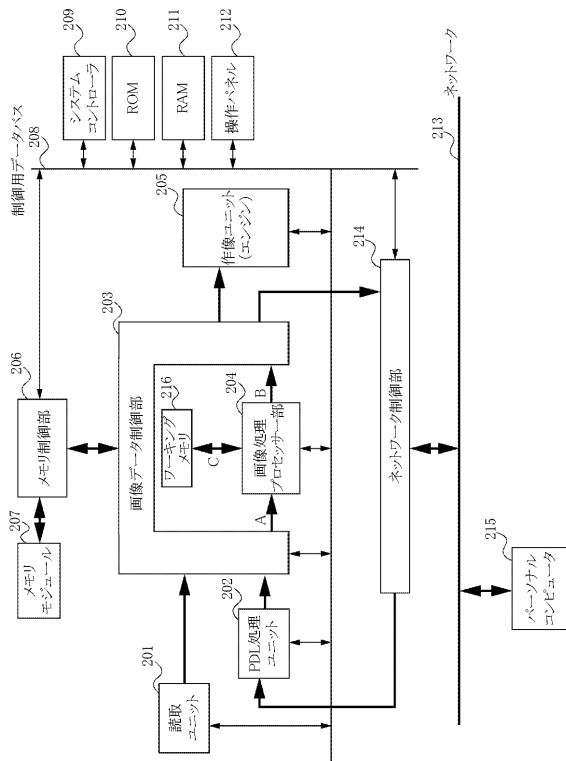
2 1 5 パーソナルコンピュータ

2 1 6 ワーキングメモリ

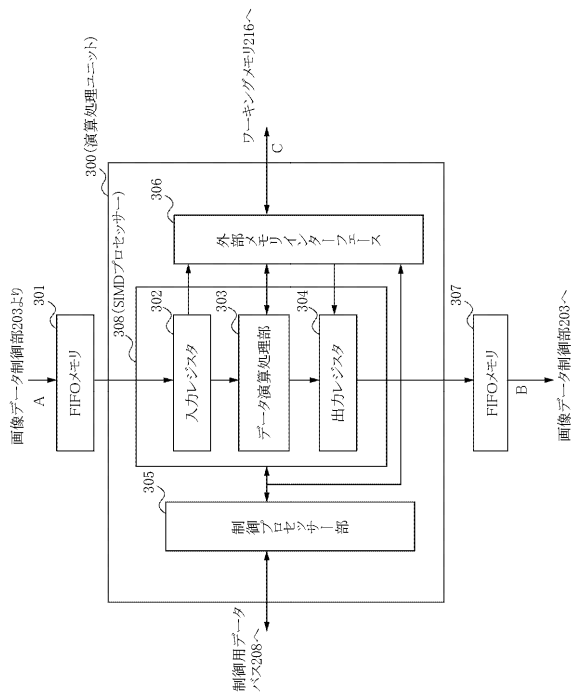
【 図 1 】



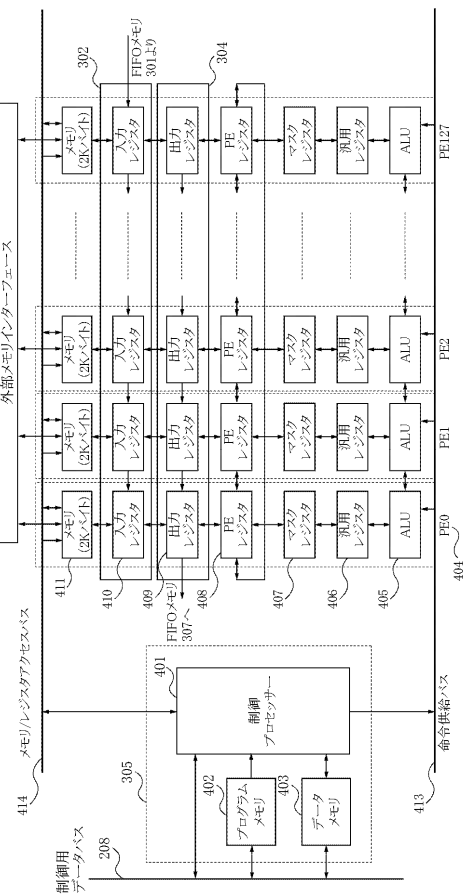
【 図 2 】



【 図 3 】



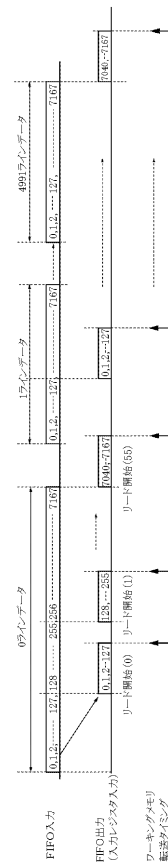
【 図 4 】



【図 5】

画像信号		主走査方向				
副走査方向	(0,0)	(0,1)	(0,2)	-----	(0,7166)	(0,7167)
	(1,0)	(1,1)	(1,2)		(1,7166)	(1,7167)
	(2,0)	(2,1)	(2,2)		(2,7166)	(2,7167)
	(3,0)	(3,1)	(3,2)		(3,7166)	(3,7167)
	⋮	⋮	⋮	⋮	⋮	⋮
	(4989,0)	(4989,1)	(4989,2)		(4989,7166)	(4989,7167)
	(4990,0)	(4989,1)	(4989,2)		(4990,7166)	(4990,7167)
	(4991,0)	(4989,1)	(4989,2)		(4991,7166)	(4991,7167)

【図 6】

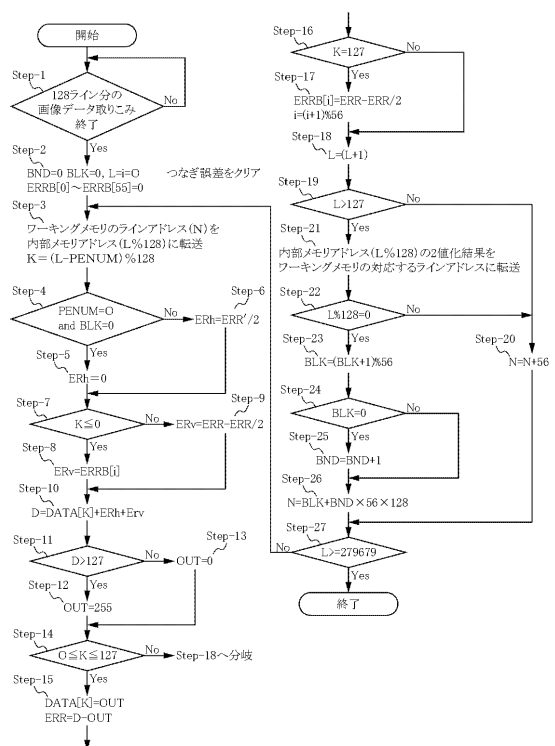


【図 7】

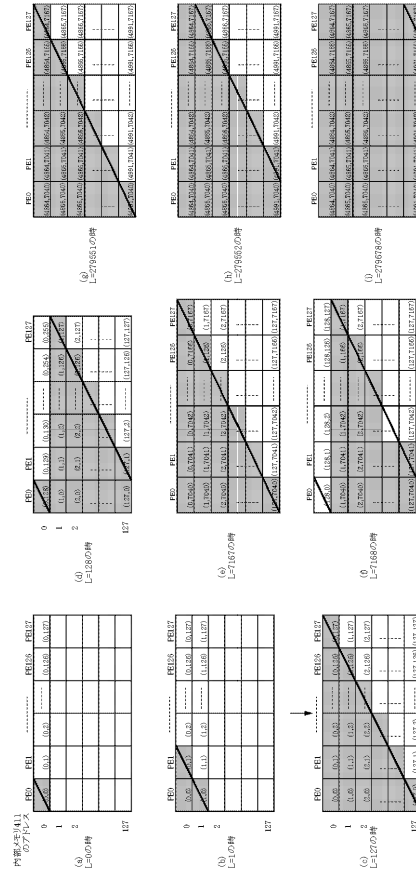
ワーキングメモリラインアドレス

0	(0,0)	(0,1)	(0,2)	...	(0,126)	(0,127)
1	(0,128)	(0,129)	(0,130)	...	(0,254)	(0,255)
2	(0,256)	(0,1)	(0,2)	...	(0,126)	(0,383)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
54	(0,6912)	(0,6913)	(0,6914)	...	(0,7038)	(0,7039)
55	(0,7040)	(0,7041)	(0,7042)	...	(0,7166)	(0,7167)
56	(1,0)	(1,1)	(1,2)	...	(1,126)	(1,127)
57	(1,128)	(1,129)	(1,130)	...	(1,254)	(1,255)
58	(1,256)	(1,257)	(1,258)	...	(1,382)	(1,383)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
110	(1,6912)	(1,6913)	(1,6914)	...	(1,7038)	(1,7039)
111	(1,7040)	(1,7041)	(1,7042)	...	(1,7166)	(1,7167)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7112	(127,0)	(127,1)	(127,2)	...	(127,126)	(127,127)
7113	(127,128)	(127,129)	(127,130)	...	(127,254)	(127,255)
7114	(127,256)	(127,257)	(127,258)	...	(127,382)	(127,383)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7166	(127,6912)	(127,6913)	(127,6914)	...	(127,7038)	(127,7039)
7167	(127,7040)	(127,7041)	(127,7042)	...	(127,7166)	(127,7167)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
279496	(4991,0)	(4991,1)	(4991,2)	...	(4991,126)	(4991,127)
279497	(4991,128)	(4991,129)	(4991,130)	...	(4991,254)	(4991,255)
279498	(4991,256)	(4991,257)	(4991,258)	...	(4991,382)	(4991,383)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
279550	(4991,6912)	(4991,6913)	(4991,6914)	...	(4991,7038)	(4991,7039)
279551	(4991,7040)	(4991,7041)	(4991,7042)	...	(4991,7166)	(4991,7167)

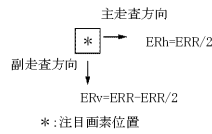
【図 8】



【図 9】



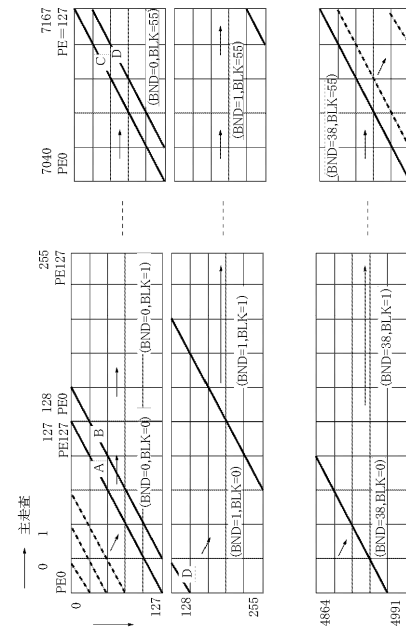
【図 11】



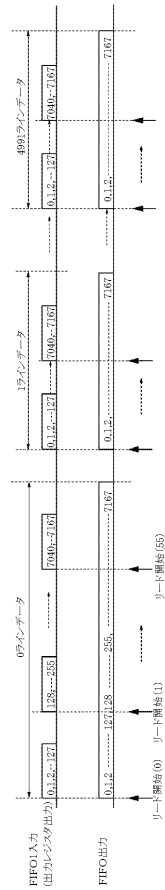
【図 10】

L	K	PE0	PE1	PE2	...	PE126	PE127
0	K=	0	-1	-2	...	-126	-127
1	K=	1	0	-1	...	-125	-126
2	K=	2	1	0	...	-124	-125
...	K=
127	K=	127	126	125	...	1	0
128	K=	0	127	126	...	2	1
129	K=	1	0	127	...	3	2
...	K=
511	K=	127	126	125	...	1	0
...	K=
7168	K=	0	127	126	...	2	1
279551	K=	127	126	125	...	1	0
279678	K=	126	125	124	...	0	127

【図 12】



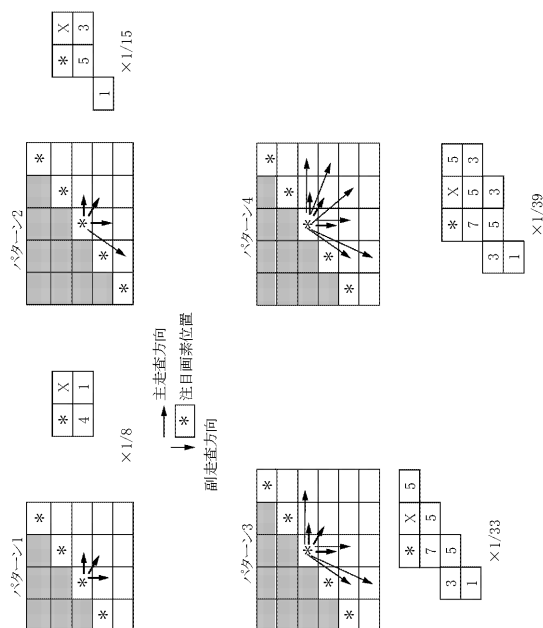
【図 13】



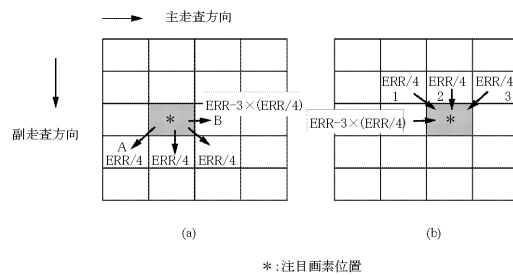
【図 14】

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)

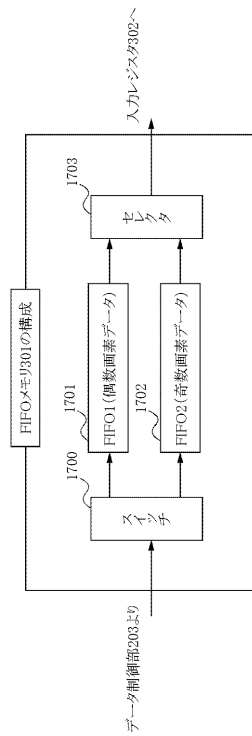
【図 15】



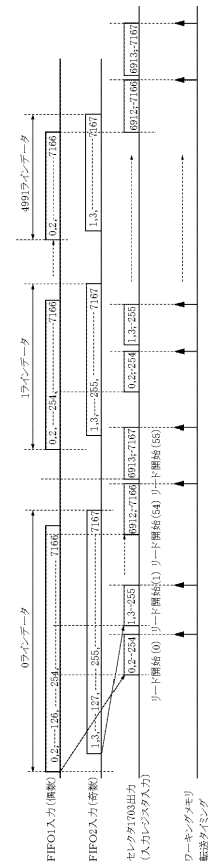
【図 16】



【 図 1 7 】



【 図 1 8 】



【 図 1 9 】

ワーキングメモリアドレス

0	(0,0)	(0,2)	(0,4)	...	(0,252)	(0,254)
1	(0,1)	(0,3)	(0,5)	...	(0,253)	(0,255)
2	(0,256)	(0,258)	(0,260)	...	(0,508)	(0,510)
...
54	(0,6912)	(0,6914)	(0,6916)	...	(0,7164)	(0,7166)
55	(0,6913)	(0,6915)	(0,6917)	...	(0,7165)	(0,7167)
56	(1,0)	(1,2)	(1,4)	...	(1,252)	(1,254)
57	(1,1)	(1,3)	(1,5)	...	(1,253)	(1,255)
58	(1,256)	(1,258)	(1,260)	...	(1,508)	(1,510)
...
110	(1,6912)	(1,6914)	(1,6916)	...	(1,7164)	(1,7166)
111	(1,6913)	(1,6915)	(1,6917)	...	(1,7165)	(1,7167)
...
7112	(127,0)	(127,2)	(127,4)	...	(127,252)	(127,254)
7113	(127,1)	(127,3)	(127,5)	...	(127,253)	(127,255)
7114	(127,256)	(127,258)	(127,260)	...	(127,508)	(127,510)
...
7166	(127,6912)	(127,6914)	(127,6916)	...	(127,7164)	(127,7166)
7167	(127,6913)	(127,6915)	(127,6917)	...	(127,7165)	(127,7167)
...
279496	(4991,0)	(4991,2)	(4991,4)	...	(4991,252)	(4991,254)
279497	(4991,1)	(4991,3)	(4991,5)	...	(4991,253)	(4991,255)
279498	(4991,256)	(4991,258)	(4991,260)	...	(4991,508)	(4991,510)
...
279550	(4991,6912)	(4991,6914)	(4991,6916)	...	(4991,7164)	(4991,7166)
279551	(4991,6913)	(4991,6915)	(4991,6917)	...	(4991,7165)	(4991,7167)

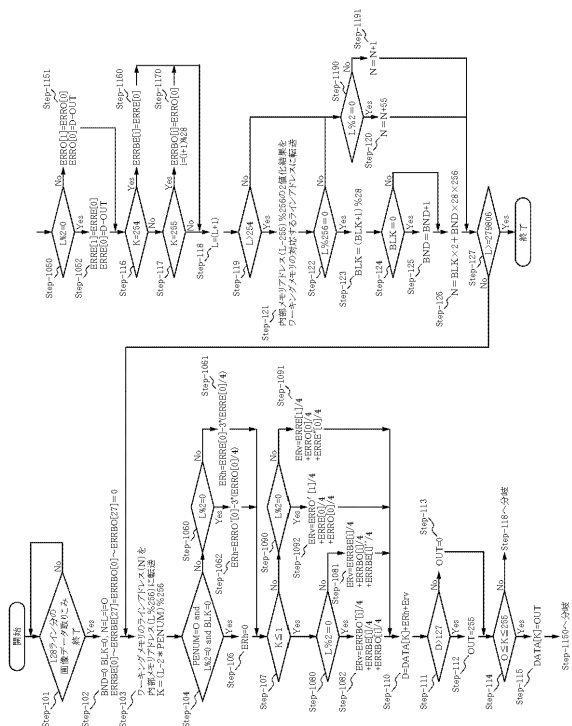
0ラインデュータ

1ラインデュータ

127ラインデュータ

4991ラインデュータ

【 図 2 0 】



【図 2 1】

	1	0	1	2	126	127
0	K=	0	-2	-4	PE126	PE127
1	K=	1	-1	-3	252	254
2	K=	2	0	-2	251	253
...	K=	250	252
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

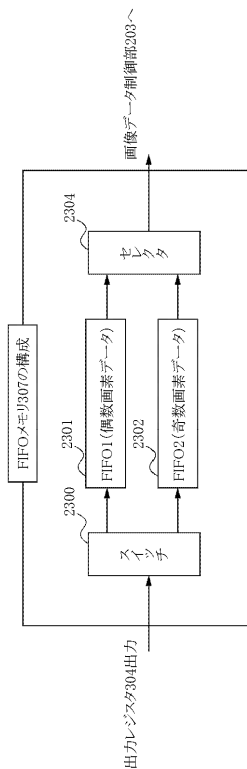
【図 2 3】

	PE0	PE1	PE2	PE126	PE127
0	0	1	2	3	4	5
1	6	7	8	9	10	11
2	12	13	14	15	16	17
3	18	19	20	21	22	23
...
126	252	253	254	255	256	257
127	258	259	260	261	262	263
...
254	252	253	254	255	256	257
255	258	259	260	261	262	263
...
279804	252	253	254	255	256	257
279805	258	259	260	261	262	263
...
279804	252	253	254	255	256	257
279805	258	259	260	261	262	263
...

【図 2 2】

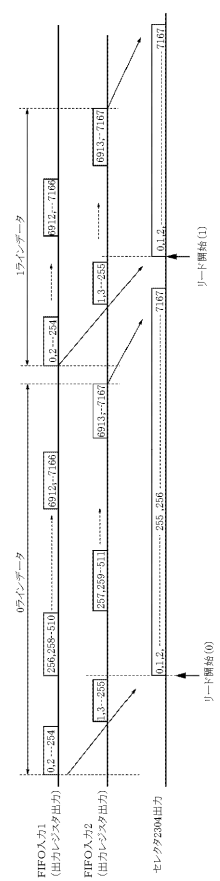
	PE0	PE1	PE2	PE126	PE127
0	0	1	2	3	4	5
1	6	7	8	9	10	11
2	12	13	14	15	16	17
...
254	252	253	254	255	256	257
255	258	259	260	261	262	263
...
279804	252	253	254	255	256	257
279805	258	259	260	261	262	263
...
279804	252	253	254	255	256	257
279805	258	259	260	261	262	263
...

【図 2 4】

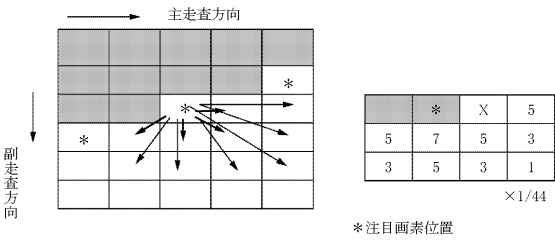


■ は値化済みの面素を示す。
* は割列毎に与えられる面素位置を示す。

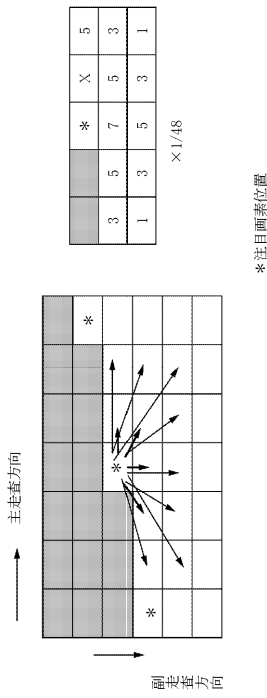
【図 2 5】



【図 2 6】



【図 2 7】



【図 2 8】

ワーキングメモリアドレス

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)
...
284487	(4991,0)	(4991,3)	(4991,6)	...	(4991,378)	(4991,381)
284488	(4991,1)	(4991,4)	(4991,7)	...	(4991,379)	(4991,382)
284489	(4991,2)	(4991,5)	(4991,8)	...	(4991,380)	(4991,383)
...
284542	(4991,6913)	(4991,6916)	(4991,6919)	...	(4991,7291)	(4991,7294)
284543	(4991,6914)	(4991,6917)	(4991,6920)	...	(4991,7292)	(4991,7295)
...

0ラインデータ

1ラインデータ

4991ラインデータ

【図 29】

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

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 聡

(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