

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3667025号

(P3667025)

(45) 発行日 平成17年7月6日(2005.7.6)

(24) 登録日 平成17年4月15日(2005.4.15)

(51) Int. Cl.⁷

F I

H 0 4 N 1/387

H 0 4 N 1/387 1 0 1

B 4 1 J 2/485

G 0 6 F 3/12 L

G 0 6 F 3/12

B 4 1 J 3/12 B

請求項の数 26 (全 24 頁)

(21) 出願番号	特願平9-46636	(73) 特許権者	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22) 出願日	平成9年2月28日(1997.2.28)	(74) 代理人	100076428 弁理士 大塚 康德
(65) 公開番号	特開平10-243215	(74) 代理人	100093908 弁理士 松本 研一
(43) 公開日	平成10年9月11日(1998.9.11)	(72) 発明者	木村 岳男 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
審査請求日	平成14年12月3日(2002.12.3)	審査官	白石 圭吾
		(56) 参考文献	特開平05-046152 (JP, A) 特開平05-183735 (JP, A) 最終頁に続く

(54) 【発明の名称】 解像度変換モジュールとそれが組み込まれたプリンタドライバ、及びその画像解像度変換方法

(57) 【特許請求の範囲】

【請求項1】

画像データの解像度を変換する解像度変換モジュールであって、
 変換前後の解像度を表わす情報を外部より受け取るステップと、
 前記情報から変換前の原画像の各画素の倍率を求めるステップと、
 変換後の画像サイズを基準に分割された処理領域の情報を外部より受け取るステップと、
 前記分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求めるステップと、
 求められた前記変換前の原画像上の領域に対応する情報を外部に渡すステップと、
 外部より、前記領域の原画像データを受け取るステップと、
 前記受け取った原画像データのサイズを前記各画素の倍率を基に変更するステップと、
 外部に対して、前記領域の解像度変換後のデータを渡すステップとを備えることを特徴とする解像度変換モジュール。

【請求項2】

画像処理プログラムに組み込まれて使用され画像データの解像度を変換する解像度変換モジュールであって、
 変換前後の解像度を表わす情報を前記画像処理プログラムから受け取るステップと、
 前記情報から変換前の原画像の各画素の倍率を求めるステップと、
 変換後の画像サイズを基準に分割された処理領域の情報を、前記画像処理プログラムから受け取るステップと、

10

20

前記分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求めるステップと、

前記画像処理プログラムに対し、前記求められた原画像上の領域の情報を渡すと共に原画像データの受渡し要求をし、該領域の原画像データを受け取るステップと、前記求められた原画像上の領域毎に、前記各画素の倍率を基に画像のサイズを変更するステップと、前記画像処理プログラムに対し、サイズ変更後のデータの引き取り要求をし、前記領域の解像度変換後のデータを渡すステップとを備えることを特徴とする解像度変換モジュール。

【請求項 3】

前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズであることを特徴とする請求項 1 又は 2 記載の解像度変換モジュール。 10

【請求項 4】

前記画像処理プログラムはプリンタドライバであって、前記変換後の解像度はプリンタの解像度、前記変換後の画像サイズを基準に分割された処理領域はバンド領域であることを特徴とする請求項 2 記載の解像度変換モジュール。

【請求項 5】

前記変換前の原画像の各画素の倍率は、複数画素の倍率が変換前後の解像度の倍率になるような整数倍或は整数分の 1 の組み合わせからなることを特徴とする請求項 1 又は 2 記載の解像度変換モジュール。

【請求項 6】

前記分割された処理領域の情報は領域の開始点の位置及び領域の大きさであることを特徴とする請求項 1 又は 2 記載の解像度変換モジュール。 20

【請求項 7】

前記原画像上の領域を求めるステップは、解像度変換処理に必要な画素数を使って開始点及び大きさを調整するステップと、原画像の端部を考慮して領域の開始点及び大きさを調整するステップとを含むことを特徴とする請求項 6 記載の解像度変換モジュール。

【請求項 8】

前記求められた原画像上の領域の情報は領域の開始点の位置及び領域の大きさであることを特徴とする請求項 1 又は 2 記載の解像度変換モジュール。 30

【請求項 9】

前記画像のサイズを変更するステップは、画像データを 1 ライン分ずつ受け取るステップと、受け取った画像データをサイズ変更に必要なライン分保持するステップと、必要でなくなったラインの画像データを破棄して新たなラインの画像データに更新するステップと、処理後のデータを 1 ライン分ずつ出力するステップとを含むことを特徴とする請求項 1 又は 2 記載の解像度変換モジュール。

【請求項 10】

画像データの解像度変換モジュールが組み込まれたプリンタドライバであって、前記解像度変換モジュールが、変換前後の解像度を表わす情報を前記ドライバから受け取るステップと、前記情報から変換前の原画像の各画素の倍率を求めて、これを記憶するステップと、変換後の画像サイズを基準に分割された処理領域の情報を、前記ドライバから受け取るステップと、前記分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求めるステップと、前記ドライバに対し、前記求められた原画像上の領域の情報を渡すと共に原画像データの受渡し要求をし、該領域の原画像データを受け取るステップと、前記求められた原画像上の領域毎に、前記各画素の倍率を基に画像のサイズを変更するステップと、前記ドライバに対し、サイズ変更後のデータの引き取り要求をし、前記領域の解像度変換 40 50

データを渡すステップとを備えることを特徴とするプリンタドライバ。

【請求項 1 1】

前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズであることを特徴とする請求項 1 0 記載のプリンタドライバ。

【請求項 1 2】

画像データの解像度変換を行う画像解像度変換方法であって、
解像度変換を行う解像度変換モジュールと画像データの解像度変換を必要とする画像処理プログラムとがある場合に、

前記画像処理プログラムから前記解像度変換モジュールへ、変換前後の解像度を表わす情報と、変換後の画像サイズを基準に分割された処理領域の情報とを渡し、

10

前記解像度変換モジュールは、前記変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求め、前記分割された処理領域からこれに対応する変換前の原画像上の領域を前記各画素の倍率を基に求めて、該領域の情報を前記画像処理プログラムに渡して、前記原画像上の領域の原画像データの受渡しを要求し、

前記画像処理プログラムから前記解像度変換モジュールへ、前記領域の原画像データが渡されると、前記解像度変換モジュールは、前記原画像上の領域毎に前記各画素の倍率を基に画像のサイズを変更して、サイズの変更された画像データを前記画像処理プログラムに返すことを特徴とする画像解像度変換方法。

【請求項 1 3】

前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズ

20

【請求項 1 4】

前記原画像上の領域を求める場合に、解像度変換処理に必要な画素数を使って開始点及び大きさを調整し、原画像の端部を考慮して領域の開始点及び大きさを調整することを特徴とする請求項 1 2 記載の画像解像度変換方法。

【請求項 1 5】

画像データの解像度を変換するプログラムを格納するコンピュータ可読メモリであって、
変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求める倍率作成モジュールと、

変換後の画像サイズを基準に分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求める対応領域計算モジュールと、

30

前記求められた領域の原画像データのサイズを前記各画素の倍率を基に変更するサイズ変更処理モジュールと、

外部より、前記変換前後の解像度を表わす情報、前記分割された処理領域の情報、及び前記求められた領域の原画像データを受け取り、外部に対して、前記求められた領域の情報と前記分割された領域の解像度変換後のデータとを渡すインターフェースモジュールとを含むことを特徴とするコンピュータ可読メモリ。

【請求項 1 6】

画像データの解像度を変換するプログラムを格納するコンピュータ可読メモリであって、
解像度の変換を行う解像度変換モジュールと該解像度変換モジュールを組み込むプリンタ

40

ドライバとを有し、
前記解像度変換モジュールは、

変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求め倍率作成モジュールと、

変換後の画像サイズを基準に分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求める対応領域計算モジュールと、

前記求められた領域の原画像データのサイズを前記各画素の倍率を基に変更するサイズ変更処理モジュールと、

前記プリンタドライバより、前記変換前後の解像度を表わす情報、前記分割された処理領域の情報、及び前記求められた領域の原画像データを受け取り、前記プリンタドライバに

50

対して、前記求められた領域の情報と前記分割された領域の解像度変換後のデータとを渡すインターフェースモジュールとを含むことを特徴とするコンピュータ可読メモリ。

【請求項 17】

前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズであることを特徴とする請求項 15 又は 16 記載のコンピュータ可読メモリ。

【請求項 18】

画像データの解像度変換を行う画像解像度変換方法であって、
変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求め、
変換後の画像サイズを基準に分割された処理領域の情報からこれに対応する変換前の原画像上の領域を前記各画素の倍率を基に求め、
前記原画像上の領域の原画像データ単位に、前記各画素の倍率を基に画像のサイズを変更することを特徴とする画像解像度変換方法。

10

【請求項 19】

前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズであることを特徴とする請求項 18 記載の画像解像度変換方法。

【請求項 20】

前記変換前の原画像の各画素の倍率は、複数画素の倍率が変換前後の解像度の倍率になるような整数倍或は整数分の 1 の組み合わせからなることを特徴とする請求項 18 記載の画像解像度変換方法。

【請求項 21】

前記原画像上の領域を求める場合に、解像度変換処理に必要な画素数を使って開始点及び大きさを調整し、原画像の端部を考慮して領域の開始点及び大きさを調整することを特徴とする請求項 18 記載の画像解像度変換方法。

20

【請求項 22】

画像データの解像度変換を伴う画像処理を行う画像処理装置であって、
前記解像度変換が、
変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求める倍率作成手段と、
変換後の画像サイズを基準に分割された処理領域の情報からこれに対応する変換前の原画像上の領域を前記各画素の倍率を基に求める対応領域計算手段と、
前記原画像上の領域の原画像データ単位に、前記各画素の倍率を基に画像のサイズを変更するサイズ変更手段とにより行われることを特徴とする画像処理装置。

30

【請求項 23】

前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズであることを特徴とする請求項 22 記載の画像処理装置。

【請求項 24】

前記変換前の原画像の各画素の倍率は、複数画素の倍率が変換前後の解像度の倍率になるような整数倍或は整数分の 1 の組み合わせからなることを特徴とする請求項 22 記載の画像処理装置。

【請求項 25】

前記原画像上の領域を求める場合に、解像度変換処理に必要な画素数を使って開始点及び大きさを調整し、原画像の端部を考慮して領域の開始点及び大きさを調整することを特徴とする請求項 22 記載の画像処理装置。

40

【請求項 26】

前記変換後の解像度はプリンタの解像度、前記変換後の画像サイズを基準に分割された処理領域はバンド領域であることを特徴とする請求項 22 記載の画像処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、画像を表現する画素数を増加したり減少したりして画像の解像度を変換するた

50

めの画像解像度変換方法に関するものである。特に、ホストコンピュータ上のプリンタドライバに組み込み可能な該画像解像度変換方法を実現する解像度変換モジュールやそれが組み込まれたプリンタドライバに関し、コンピュータ上の画像データの解像度をインクジェットプリンタ等の出力装置の解像度に合わせて変更すること等を目的とする。

【0002】

【従来の技術】

一般に、コンピュータシステムでは、入出力装置間、入出力装置とコンピュータ本体間、或はシステム間で、それぞれが目的に適合する解像度を使用するため異なる解像度で画像を処理することが多い。この中で、最も頻繁に発生するのは、コンピュータ本体内或いは表示の解像度とプリンタの解像度との相違である。

10

【0003】

そこで、プリンタが印刷するデータを作成してプリンタに描画命令を出力するプリンタドライバと呼ばれるコンピュータ上で動作するプログラムが、各OSにおいて標準化された文字や図形や画像を描画する命令を、各プリンタが解釈できる描画命令に変換する処理を行うが、この際、プリンタドライバは、アプリケーション上での処理解像度と実際に印刷を行うプリンタでの解像度との違いを意識する必要がある。例えば、アプリケーションが90dpiでデータを管理しており、360dpiの解像度のプリンタから出力する場合には、縦横ともに4倍の画素数からなるページを作成する必要がある。

【0004】

本発明が特に好適に適用されるところのインクジェットプリンタに代表されるシリアルプリンタでは、PDL(ページ記述言語)を持たず、プリンタドライバで生成されたビットマップデータを基本とする印刷命令を受け取って印刷を行うのが一般的である。したがって、これらのプリンタドライバにおいてはベクターで表現された文字や図形データは、プリンタの解像度に見合った画素数で直接メモリ上に描画することで出力解像度のビットマップデータに変換し、また元来ビットマップデータであった画像データは出力解像度に合わせて拡大や縮小操作によってそのサイズを変更することで出力解像度のビットマップデータに変換する。

20

【0005】

ビットマップデータを拡大したり縮小することによる解像度変換方法には各種あり、例えば拡大では、各画素を単純に拡大倍率分繰り返す0次補間、拡大によって生じた画素を元の画素からの距離の比率等によって埋める線形補間、周囲の画素の配置状態によって各種の補間方法を切り替えて最適な拡大を行う方法(特開平7-105359号)等がある。

30

【0006】

【発明が解決しようとする課題】

しかしながら、プリンタドライバにおける画像処理には多くの制約があり、実際には、各方式そのままでは非整数倍率を含む任意の倍率に対応できない、バンド間で出力結果に不連続が生じるような任意のライン数単位での処理に対応できない等の理由で、解像度変換処理のドライバへの組み込みができない或いは難しいという問題があった。この問題は、特にそれぞれの画素の処理に際し、周囲の画素を必要とするデジタルイメージフィルタ系の画像処理方法においては顕著である。しかしながら、一般にデジタルイメージフィルタ系の処理は、周囲の画素を使用しない0次補間方法のような単純な処理に比べて処理後の画質は良いとされている。

40

【0007】

本発明は、簡単な操作及び手順で非整数倍率を含む任意の倍率に対応できる画像解像度変換方法及びそれを実現する画像解像度変換モジュールを提供する。

特に、コンピュータ上でソフトウェアとして実現されるデジタルフィルタ系の拡大縮小方法であっても、統一されたインターフェースによってプリンタドライバに効率良く組み込まれる画像解像度変換方法及びそれを実現する画像解像度変換モジュールを提供する。

【0008】

更に、上記画像解像度変換モジュールが組み込まれたプリンタドライバを提供する。

50

【 0 0 0 9 】

【課題を解決するための手段】

そこで、上記課題を解決するために、本発明の解像度変換モジュールは、画像データの解像度を変換する解像度変換モジュールであって、変換前後の解像度を表わす情報を外部より受け取るステップと、前記情報から変換前の原画像の各画素の倍率を求めるステップと、変換後の画像サイズを基準に分割された処理領域の情報を外部より受け取るステップと、前記分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求めるステップと、求められた前記変換前の原画像上の領域に対応する情報を外部に渡すステップと、外部より、前記領域の原画像データを受け取るステップと、前記受け取った原画像データのサイズを前記各画素の倍率を基に変更するステップと、外部に
10 対して、前記領域の解像度変換後のデータを渡すステップとを備えることを特徴とする。なお、前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズである。

【 0 0 1 0 】

ここで、前記解像度変換モジュールは画像処理プログラムに組み込まれる。また、前記画像処理プログラムはプリンタドライバであって、前記変換後の解像度はプリンタの解像度、前記変換後の画像サイズを基準に分割された処理領域はバンド領域である。また、前記変換前の原画像の各画素の倍率は、複数年素の倍率が変換前後の解像度の倍率になるような整数倍或は整数分の1の組み合わせからなる。また、前記分割された処理領域の情報は領域の開始点の位置及び領域の大きさである。また、前記原画像上の領域を求めるステッ
20 プは、解像度変換処理に必要な画素数を使って開始点及び大きさを調整するステップと、原画像の端部を考慮して領域の開始点及び大きさを調整するステップとを含む。また、前記求められた原画像上の領域の情報は領域の開始点の位置及び領域の大きさである。また、前記画像のサイズを変更するステップは、画像データを1ライン分ずつ受け取るステップと、受け取った画像データをサイズ変更に必要なライン分保持するステップと、必要でなくなったラインの画像データを破棄して新たなラインの画像データに更新するステップと、処理後のデータを1ラインずつ出力するステップとを含む。

【 0 0 1 1 】

又、本発明のプリンタドライバは、画像データの解像度変換モジュールが組み込まれたプリンタドライバであって、前記解像度変換モジュールが、変換前後の解像度を表わす情報を前記ドライバから受け取るステップと、前記情報から変換前の原画像の各画素の倍率を求めて、これを記憶するステップと、変換後の画像サイズを基準に分割された処理領域の情報を、前記ドライバから受け取るステップと、前記分割された処理領域からこれに対応する変換前の原画像上の領域を、前記各画素の倍率を基に求めるステップと、前記ドライバに対し、前記求められた原画像上の領域の情報を渡すと共に原画像データの受渡し要求をし、該領域の原画像データを受け取るステップと、前記求められた原画像上の領域毎に、前記各画素の倍率を基に画像のサイズを変更するステップと、前記ドライバに対し、サイズ変更後のデータの引き取り要求をし、前記領域の解像度変換データを渡すステップとを備えることを特徴とする。なお、前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズである。
40

【 0 0 1 2 】

又、本発明の画像解像度変換方法は、画像データの解像度変換を行う画像解像度変換方法であって、解像度変換を行う解像度変換モジュールと画像データの解像度変換を必要とする画像処理プログラムとがある場合に、前記画像処理プログラムから前記解像度変換モジュールへ、変換前後の解像度を表わす情報と、変換後の画像サイズを基準に分割された処理領域の情報を渡し、前記解像度変換モジュールは、前記変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求め、前記分割された処理領域からこれに対応する変換前の原画像上の領域を前記各画素の倍率を基に求めて、該領域の情報を前記画像処理プログラムに渡して、前記原画像上の領域の原画像データの受渡しを要求し、前記画像処理プログラムから前記解像度変換モジュールへ、前記領域の原画像データが渡されると
50

、前記解像度変換モジュールは、前記原画像上の領域毎に前記各画素の倍率を基に画像のサイズを変更して、サイズの変更された画像データを前記画像処理プログラムに返すことを特徴とする。なお、前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズである。ここで、前記原画像上の領域を求める場合に、解像度変換処理に必要な画素数を使って開始点及び大きさを調整し、原画像の端部を考慮して領域の開始点及び大きさを調整する。

【0013】

又、本発明の画像解像度変換方法は、画像データの解像度変換を行う画像解像度変換方法であって、変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求め、変換後の画像サイズを基準に分割された処理領域の情報からこれに対応する変換前の原画像上の領域を前記各画素の倍率を基に求め、前記原画像上の領域の原画像データ単位に、前記各画素の倍率を基に画像のサイズを変更することを特徴とする。なお、前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズである。ここで、前記変換前の原画像の各画素の倍率は、複数画素の倍率が変換前後の解像度の倍率になるような整数倍或は整数分の1の組み合わせからなる。また、前記原画像上の領域を求める場合に、解像度変換処理に必要な画素数を使って開始点及び大きさを調整し、原画像の端部を考慮して領域の開始点及び大きさを調整する。

10

【0014】

又、本発明の画像処理装置は、画像データの解像度変換を伴う画像処理を行う画像処理装置であって、前記解像度変換が、変換前後の解像度を表わす情報から変換前の原画像の各画素の倍率を求める倍率作成手段と、変換後の画像サイズを基準に分割された処理領域の情報からこれに対応する変換前の原画像上の領域を前記各画素の倍率を基に求める対応領域計算手段と、前記原画像上の領域の原画像データ単位に、前記各画素の倍率を基に画像のサイズを変更するサイズ変更手段とにより行われることを特徴とする。なお、前記変換前後の解像度を表わす情報は、画素数で表された変換前後の画像データのサイズである。ここで、前記変換前の原画像の各画素の倍率は、複数画素の倍率が変換前後の解像度の倍率になるような整数倍或は整数分の1の組み合わせからなる。また、前記原画像上の領域を求める場合に、解像度変換処理に必要な画素数を使って開始点及び大きさを調整し、原画像の端部を考慮して領域の開始点及び大きさを調整する。また、前記変換後の解像度はプリンタの解像度、前記変換後の画像サイズを基準に分割された処理領域はバンド領域である。

20

30

【0015】

【発明の実施の形態】

以下、添付図面にしたがって、本発明による実施の形態を詳細に説明する。

(第1の実施の形態)

第1の実施の形態を図1から図15に基づき説明する。

<本実施の形態の画像解像度変換方法を実現するシステム構成例>

図1は、本発明の実施形態を示すシステム構成図である。尚、本実施の形態では、本発明の画像解像度変換方法をプリンタドライバに組み込んだ構成について説明するが、これに限定されることなく、本発明の画像解像度変換方法を他のドライバに組み込んで、独立して解像度変換モジュールとして使用しても、同様の作用効果を得られることは明らかである。

40

【0016】

図中、1は本実施の形態の画像解像度変換方法を具備するプリンタドライバを動作させるためのホストコンピュータであり、CPU、RAM、ROM、外部記憶装置、プリンタポート等を備えている。ホストコンピュータ1では、ROM内に格納されたプログラム、或は外部記憶装置からRAMにロードされたプログラムに従って、CPUにより演算・制御が行われて、描画命令の作成、ビットマップデータの作成、解像度の変換、印刷コマンドの作成・出力を行う。2はプリンタドライバによって制御する、例えばインクジェットプリンタ等の2値カラープリンタである。ホストコンピュータ1とプリンタ2とは通信ケー

50

ブル3で接続され、ホストコンピュータ1からプリンタ2に対してビットマップデータをベースとする印刷データが送信され、印刷が行われる。

【0017】

図2は、本発明の実施の形態であるところの解像度変換方法を組み込んだプリンタドライバ100を使用して印刷を行う場合のデータの流れ図である。

このプリンタドライバ100では、アプリケーションから画像102、図形103、文字104の各描画命令を受けると、各命令に対応した描画ルーチン105~107を使用してバンドバッファ上にビットマップデータ108をプリンタ2の解像度で作成する。その後、作成されたビットマップデータ108は印刷コマンド生成部109によってプリンタ2が印刷できる形式のデータ(印刷コマンド110)に変換され、通信ケーブル3を通じてプリンタ2へ送信され、印刷される。

10

【0018】

図3は、図2における画像描画部105のブロック図である。

画像描画部105は、画像描画命令102をアプリケーション101から受け取ると、ホストコンピュータ1のモニタ上で観察される画像の色とプリンタ2から出力される印刷物の色とができるだけ近づくように、カラーマッチング部200で画像データ内の各画素の色を変更し、次に解像度変換部201でカラーマッチングされた画像データをプリンタ2の持つ解像度に合わせるように、そのサイズを変更して画素数を調整する。

【0019】

例えば、アプリケーション101から渡された画像描画命令102による画像データが、縦横ともに100画素からなる正方形の画像でありその解像度が90dpiであったならば、360dpiの解像度を持つプリンタ2から同じ面積を持つ画像を出力するためには、縦横ともに400画素からなる正方形の画像に大きさを変更しなければならない。つまり、アプリケーション101から渡された画像データの各画素はそれぞれ16(4×4)倍の画素数になるように解像度変換部201によって処理される。

20

【0020】

この場合に画像の大きさを変更(拡大)する方法には様々なものがあるが、本実施の形態によれば、解像度変換方法のインターフェースを統一することができるようになる。

<実施の形態1の画像解像度変換方法の手順例>

図4は、本発明の実施の形態であるところの画像解像度変換方法の処理の流れを示す図面である。以下順に処理の流れを説明する。

30

【0021】

まず、ステップS300にて、処理に必要な作業メモリの確保などの初期化処理を行う。次に、ステップS301において解像度変換のためのサイズ変更情報を獲得する。具体的には、原画像の縦横の各画素数(DHとDW)、各画素のビット長、各画素の格納され方、及び解像度変換後の縦横の各画素数(SHとSW)等の情報を得る。そしてこれらの内、解像度変換前後の縦横の各画素数の情報から、ステップS302において各画素の倍率(拡大倍率又は縮小倍率)を保持するテーブルを作成する。

【0022】

尚、ステップS301において、本例のように解像度変換モジュールがプリンタドライバに組み込まれている場合には、サイズ変更情報をプリンタドライバより獲得することとなる。ここで、プリンタドライバがサイズ変更情報をユーザインタフェースを介してユーザの指定により得るか、プリンタからの情報伝送により自動的に得るかは問わない。一方、解像度変換モジュールを独立で使用する場合は、直接ユーザインタフェースを介してユーザの指定により設定されても良い。

40

【0023】

解像度変換例として、図5に示した画像500(縦4画素、横6画素)を図6に示したサイズの画像600(縦6画素、横10画素)に変換する場合を示す。この例では、画像500が縦横にそれぞれ6/4、10/6倍に拡大されることを示している。

(倍率テーブル例及びその作成手順例)

50

図7は、本例での倍率テーブルの内容を示したものであり、画像500の各列はそれぞれX方向の倍率テーブル701にしたがって2, 1, 2, 2, 1, 2倍され、各行はそれぞれY方向の倍率テーブル702にしたがって2, 1, 2, 1倍されることが示されている。そしてこれらのテーブルによって、画像500における画素501は画像600における領域601(縦2画素、横2画素)の、画素502は領域602(縦1画素、横2画素)の領域を作成する元(原画素)となることがわかる。このように縦横それぞれの倍率テーブルを用意することで、縦横独立して非整数倍を含む任意の倍率で画像の解像度変換処理を行うことができるようになる。

【0024】

図8は、倍率テーブル701, 702の作成方法の手順を示した図面である。図中、SH、SWはそれぞれ解像度変換前の画像の縦と横の画素数(例ではSH=4、SW=6)を示し、DH、DWはそれぞれ解像度変換後の画像の縦と横の画素数(例ではDH=6、DW=10)を示している。TY[i]はY方向の倍率テーブル702のi番目の要素を示し、TX[i]はX方向の倍率テーブル701のi番目の要素を示している。なお割算は整数演算であり結果は切り捨てられる(例えば、10/6=1)。ここで、ステップS801やS805に示す変数=a, b, cは、変数をaからbまでc単位で増加(cがマイナス値であれば減少)させながら、下位に接続されているステップを繰り返すことを意味している。以下、同様である。

【0025】

処理の流れを説明すると、まずステップS800で演算の誤差を格納する変数eを0で初期化する。次にステップS801では変数iを1ずつ増やしながらからSH-1まで、ステップS802とS803とを繰り返す。ステップS802では、Y方向の倍率テーブルの各要素の値を蓄積された計算誤差を考慮しながら求め、ステップS803では計算誤差の更新を行う。この結果、例えばTY[i](i=0~3)としてY方向の倍率テーブル702が作成される。

【0026】

同様にステップS804で演算の誤差を格納する変数eを再度0で初期化し、次にステップS805では変数iを1ずつ増やしながらからSW-1までステップS806とS807とを繰り返す。ステップS806ではX方向の倍率テーブルの各要素の値を蓄積された計算誤差を考慮しながら求め、ステップS807では計算誤差の更新を行う。この結果、例えばTX[i](i=0~5)としてX方向の倍率テーブル701が作成される。

【0027】

倍率テーブル701, 702は上記方法によって作成されたテーブルである。倍率テーブルの各要素はここに示したように整数値である。倍率テーブルを使う本実施の形態は、整数倍の組み合わせにより擬似的に任意の倍率の解像度変換ができるようになっている。このことは整数倍の画像サイズの変更しかできない解像度変換方法であっても、それを本実施の形態の方法に使用すれば擬似的に非整数倍の画像サイズ変更が可能になることを示している。尚、本実施の形態の方法によれば、倍率テーブルの各要素の値の合計は拡大の場合は処理後の縦横それぞれの画素数に一致する。本例では、X方向の倍率テーブル701の各要素の合計は10(2+1+2+2+1+2)となり、拡大後の画像600の横の画素数10と一致し、Y方向の倍率テーブル702の各要素の合計は6(2+1+2+1)となり、拡大後の画像600の縦の画素数6と一致する。

【0028】

(解像度変換処理例)

さて、ステップS302によって倍率テーブルが作成されたならば、次にステップS303において処理対象となる全ての領域の処理が終了したかどうかをチェックしながら、ステップS304からS312までを各領域に対して繰り返し適用する。ここで、領域の定義については以下に詳細に示す。

【0029】

(領域の分割例及び周辺画素の使用例)

10

20

30

40

50

ステップS304では、解像度変換処理を行う領域を決定することで、処理領域の分割を行う。本実施の形態の方法が適用されるプリンタドライバではメモリ使用量に制限があることが多く、この様な領域分割ができることが必要である。特にインクジェットプリンタに代表されるシリアルプリンタ用のドライバでは、バンド（対象処理領域を短冊状に区切ったもの）と呼ばれる単位で画像処理、コマンド生成処理が行われるのが一般であるが、本実施の形態ではより自由度の高い任意の矩形領域単位に処理ができるような構成になっている。なお、この領域分割はプリンタドライバという事情から解像度変換後の領域の分割を指定することで行われる。つまり、印刷用にどの部分のデータを作成するかによって分割が決まることを意味しており、現状では先に述べたようにバンド単位に分割が行われるのが一般的であるが、これに限定されない。

10

【0030】

図9は、拡大後の画像600の占める領域から領域901を切り出した場合と、この領域の処理に必要な原画像500中の領域902との関係を示した図である。

領域902の決定には、先に示した倍率テーブル701, 702を使用している。倍率テーブル701, 702は原画像500の各画素の拡大率を格納しているが、これを用いることで拡大後の画像600の各画素が原画像500のどの画素を拡大した結果であるかをも同時に知ることができる。したがって、拡大後の領域901から拡大前の領域902を求める際には、領域901に含まれる各画素の拡大元の画素の集合を求めれば良い。ただし、画素501と領域601の関係からも明らかのように拡大後の各画素は元の画素を拡大した領域の一部でしかないが、それが領域901に属しているのならば、その拡大元の画素は領域902に含めるようにしなければならない。

20

【0031】

図10は、画像600を4つの領域1001, 1002, 1003, 1004に分割して処理を行う場合の分割例を示している。

このとき、それぞれの領域に対応する原画像の領域は、図11に示した領域1101, 1102, 1103, 1104である。先に述べた理由から、ある原画素の拡大後の領域を跨いだ分割の場合（図10）、原画像中の4つの領域1101, 1102, 1103, 1104は一部が互いに重なりあった状態となる。例えば領域1101と領域1102とでは3つの画素（3, 0）、（3, 1）、（3, 2）が両方の領域内に存在することとなる。

30

【0032】

ところで、画像の解像度変換処理では変換後のある画素を作り出すために、変換元の画素の周辺の画素も必要とするものが少なくない。例えば線形補間法では、拡大後の画素の位置によって拡大前の隣接する画素間の割合を求めて新しい画素を作り出している。また、前出の特開平7-105359号における解像度変換方法でも、処理の際に周囲の画素を必要とする。ところが周辺画素が何画素必要であるかは各方法によって、さらには求める処理精度によっても変化する。本実施の形態では、解像度変換モジュールが変換後の領域に対応する原領域を決定する際に予めこの情報を含めておくことで、ドライバ側が各方法毎にデータを引き渡す方法を変更しなくても済むようにしている。

【0033】

例えば、倍率テーブル701, 702によって求められた領域1101は領域1001に対応する原領域であるが、ドライバから呼び出される拡大方法が各原画素の周囲1画素を必要とする場合には、領域1101に領域1111を足した領域をドライバに通知することで、必要なデータの全てをドライバから渡してもらえらることとなる。同様に領域1002, 1003, 1004に対してはそれぞれ1102+1112、1103+1113、1104+1114が対応することになる。そしてこの際、新たに必要な領域（1111, 1112, 1113, 1114）が原画像で賄えるならばこれを使用することで、領域毎に処理を分けた場合でも処理後の結果は領域を分割しなかった場合と全く同じ連続した画像が得られる。何故ならば、先に述べたように各領域の周辺部分は他の領域と重なり合っているため重複して処理される部分が多少発生するものの、各画素に対しては全く同じ

40

50

処理が施されることになるからである。

【 0 0 3 4 】

図 1 2 は、画像の領域分割に際して更に画像の周囲部分の処理（端部処理）を組み込んだ例を示した図である。

図において、1 2 0 0 は原画像を示し、指定された領域から周辺画素を含んでいない原領域 1 2 0 1 が倍率テーブル 7 0 1, 7 0 2 によって求められたとする。この時、原画像 1 2 0 0 のサイズは縦 $S H$ 画素、横 $S W$ 画素であり、領域 1 2 0 1 のサイズは縦 $R H$ 画素、横 $R W$ 画素であって、領域 1 2 0 1 の左上の画素 1 2 0 2 は画像 1 2 0 0 の画素 $(R X, R Y)$ に位置するものとする（図 1 2 の例では、 $S H = 1 0, S W = 1 0, R H = 5, R W = 8, R X = 0, R Y = 1$ ）。そして、拡大処理に原画素の周囲 B 画素（図 1 2 の例では、 $B = 3$ ）の情報が必要であるとすると、その一部は原画像 1 2 0 0 中には存在しない位置の画素となってしまう。この場合、本実施の形態では、原画像 1 2 0 0 中から得られる画素はできるだけこれを利用し、原画像から得られない画素は最も近い画素の値をコピーすることで仮想的な画像 1 2 0 6 を作成し、これを解像度変換処理に使用することで画像処理手順の簡略化及び画像周辺での画質劣化を防止する。

10

【 0 0 3 5 】

図 1 2 において領域 1 2 0 3 は原画像を利用して調整された領域であり、領域 1 2 0 4 は領域 1 2 0 3 の周辺の画素をコピーして作成された領域である。この時、ドライバには領域 1 2 0 3 の開始点 1 2 0 5 の座標 $(R X', R Y')$ とそのサイズの縦 $R H'$ 画素、横 $R W'$ 画素の情報を通知することで、原画像 1 2 0 0 の内で最大限利用できる領域を解像度変換処理モジュールはドライバから得ることができる（図 1 2 の例では、 $R X' = 0, R Y' = 0, R H' = 9, R W' = 1 0$ ）。また、同時に周辺でのコピーする画素数の情報を保存しておくことで画像 1 2 0 6 を作り出すことができることは容易に想像できる。図 1 2 では、 $C T, C B, C L, C R$ がそれぞれ領域 1 2 0 3 での上端、下端、左端、右端でのコピー画素数を示している（図 1 2 の例では、 $C T = 2, C B = 0, C L = 3, C R = 1$ ）。このとき、当然のことながら、 $B + R H + B = C T + R H' + C B$ 、及び、 $B + R W + B = C L + R W' + C R$ の 2 式が必ず成立しなければならない。

20

【 0 0 3 6 】

以上のステップ $S 3 0 5$ の対応領域の計算方法の流れを示したものが、図 1 3 である。ここでは、座標 $(D R X, D R Y)$ を左上の点とする縦 $R D H$ 画素、横 $D R W$ 画素からなる領域に対応する原領域を、倍率テーブル $T X$ と $T Y$ を使って計算する。また同時に、開始点 $(D R X, D R Y)$ が原画素 $(R X, R Y)$ を拡大した時の領域のどの位置に対応しているかを、 $O X$ と $O Y$ とに格納している。この $O X$ と $O Y$ の情報は、拡大方法によっては出力開始点の計算をする上で計算係数の初期値を決める際になどに利用される重要な値となる。

30

【 0 0 3 7 】

ステップ $S 1 3 0 1$ から $S 1 3 0 6$ は、領域 1 2 0 1 の左上の点の Y 座標を求める操作である。縦方向の倍率テーブル $T Y [i]$ の各要素を順に加算しながら（ステップ $S 1 3 0 4$; z に加算後、 i を 1 増加）、開始点の Y 座標値 $D R Y$ と比較することで（ステップ $S 1 3 0 3$ ）、開始点の元となる画素の Y 座標 $R Y$ をテーブルのインデックス i の値から求めている（ステップ $S 1 3 0 6$ ）。また途中の値を利用して $O Y$ の値を求めることができる（ステップ $S 1 3 0 5$ ）。

40

【 0 0 3 8 】

ステップ $S 1 3 0 7$ から $S 1 3 0 9$ は、領域 1 2 0 1 の縦の画素数を求める操作である。縦方向の倍率テーブル $T Y [i]$ の各要素を更に順に加算しながら（ステップ $S 1 3 0 4$; z に加算後、 i を 1 増加）、値 $D R Y + D R H$ と比較することで（ステップ $S 1 3 0 8$ ）、領域 1 2 0 1 の縦の画素数 $R H$ をテーブルのインデックス i と先に求めた $R Y$ の値から求めている（ステップ $S 1 3 0 9$ ）。

【 0 0 3 9 】

ステップ $S 1 3 1 0$ から $S 1 3 1 5$ は、領域 1 2 0 1 の左上の点の X 座標と求める操作で

50

ある。横方向の倍率テーブル $T X [i]$ の各要素を順に加算しながら（ステップ $S 1 3 1 3$; z に加算後、 i を 1 増加）、開始点の X 座標値 $D R X$ と比較することで（ステップ $S 1 3 1 2$ ）、開始点の元となる画素の X 座標 $R X$ をテーブルのインデックス i の値から求めている（ステップ $S 1 3 1 5$ ）。また途中の値を利用して $O X$ の値を求めることができる（ステップ $S 1 3 1 4$ ）。

【 0 0 4 0 】

ステップ $S 1 3 1 6$ から $S 1 3 1 8$ は、領域 $1 2 0 1$ の横の画素数を求める操作である。横方向の倍率テーブル $T X [i]$ の各要素を更に順に加算しながら（ステップ $S 1 3 1 7$; z に加算後、 i を 1 増加）、値 $D R X + D R W$ と比較することで（ステップ $S 1 3 1 6$ ）、領域 $1 2 0 1$ の横の画素数 $R W$ をテーブルのインデックス i と先に求めた $R X$ の値から求めている（ステップ $S 1 3 1 8$ ）。

10

【 0 0 4 1 】

領域 $1 2 0 1$ の開始点座標 ($D R X$, $D R Y$) と大きさ ($D R W$, $D R H$) が求めたならば、次に必要な周辺画素の数と原画像の大きさを考慮して、ドライバに要求する領域 $1 2 0 1 + 1 2 0 3$ の開始点とその大きさを求める。

ステップ $S 1 3 1 9$, $S 1 3 2 0$ は以下の操作のための準備であり、 $C T$ 、 $C B$ 、 $C L$ 、 $C R$ を 0 で初期化し、 $R X'$ 、 $R Y'$ 、 $R W'$ 、 $R H'$ をそれぞれに対応する値 $R X$ 、 $R Y$ 、 $R W$ 、 $R H$ で初期化しておく。

【 0 0 4 2 】

ステップ $S 1 3 2 1$ から $S 1 3 2 5$ は、領域 $1 2 0 1$ の上側の調整を行う操作である。先に求めた領域 $1 2 0 1$ の開始点の Y 座標 $R Y$ と周辺画素数 B を 1 ずつ減じた値（ステップ $S 1 3 2 1$ ）とを比較して（ステップ $S 1 3 2 2$ ）、 $R Y$ の値がこれに同じか、大きければ使用できる原画素が上方にあることを意味するので、領域 $1 2 0 1$ を上側に 1 画素分拡張する（ステップ $S 1 3 2 3$, $S 1 3 2 4$ ）。逆に $R Y$ の方が小さければ、領域 $1 2 0 1$ は原画像 $1 2 0 0$ の上端を越えるので、上端部のコピー画素数を記憶する $C T$ に 1 を加算する（ステップ $S 1 3 2 5$ ）。

20

【 0 0 4 3 】

ステップ $S 1 3 2 6$ から $S 1 3 3 0$ は、領域 $1 2 0 1$ の下側の調整を行う操作である。先に求めた値 $R Y'$ と $R H'$ とを加算した値 R （ステップ $S 1 3 2 6$ ）と、原画像 $1 2 0 0$ の縦の画素数 $S H$ から、周辺画素数 B を 1 ずつ減じた値（ステップ $S 1 3 2 7$ ）を差し引いた値とを比較して（ステップ $S 1 3 2 8$ ）、 R の値がこれに同じか、小さければ使用できる原画素が下方にあることを意味するので、領域 $1 2 0 1$ を下側に 1 画素分拡張する（ステップ $S 1 3 2 9$ ）。逆に R の方が大きければ領域 $1 2 0 1$ は原画像 $1 2 0 0$ の下端を越えているので、下端部のコピー画素数を記憶する $C B$ に 1 を加算する（ステップ $S 1 3 3 0$ ）。

30

【 0 0 4 4 】

ステップ $S 1 3 3 1$ から $S 1 3 3 5$ は、領域 $1 2 0 1$ の左側の調整を行う操作である。先に求めた領域 $1 2 0 1$ の開始点の X 座標 $R X$ と周辺画素数 B を 1 ずつ減じた値（ステップ $S 1 3 3 1$ ）値とを比較して（ステップ $S 1 3 3 2$ ）、 $R X$ の値がこれに同じか、大きければ使用できる原画素が左方にあることを意味するので、領域 $1 2 0 1$ を左側に 1 画素分拡張する（ステップ $S 1 3 3 3$, $S 1 3 3 4$ ）。逆に $R X$ の方が小さければ領域 $1 2 0 1$ は原画像 $1 2 0 0$ の左端を越えているので、左端部のコピー画素数を記憶する $C L$ に 1 を加算する（ステップ $S 1 3 3 5$ ）。

40

【 0 0 4 5 】

ステップ $S 1 3 3 6$ から $S 1 3 4 0$ は、領域 $1 2 0 1$ の右側の調整を行う操作である。先に求めた値 $R X'$ と $R Y'$ とを加算した値 R （ステップ $S 1 3 3 6$ ）と、原画像 $1 2 0 0$ の横の画素数 $S W$ から、周辺画素数 B を 1 ずつ減じた値（ステップ $S 1 3 3 7$ ）を差し引いた値とを比較して（ステップ $S 1 3 3 8$ ）、 R の値がこれに同じか、小さければ使用できる原画素が右方にあることを意味するので、領域 $1 2 0 1$ を右側に 1 画素分拡張する（ステップ $S 1 3 3 9$ ）。逆に R の方が大きければ領域 $1 2 0 1$ は原画像 $1 2 0 0$ の右端を

50

越えているので、右端部のコピー画素数を記憶するCRに1を加算する(ステップS1330)。

【0046】

以上、上記ステップS1301からS1340によって求めた各値をドライバ側と解像度変換処理モジュール側とで使用することで、画像1206をメモリ上に作り出すことができる。そして画像1206を処理することで画像周辺の歪みを抑えた、また各領域間で不連続が生じない出力を得ることが可能となる。

(分割領域の解像度変換例)

さて、分割された領域に対する原画像上の領域(開始点(RX', RY)、縦RH'画素、横RW'画素)がステップS305によって得られたならば、ステップS306からS312によって、この領域の解像度変換を行う。

10

【0047】

解像度変換は、ステップS308で得られる状態変数Statusの値によって制御される。Statusは解像度変換の終わりを示すEND、次のデータの引渡を要求するPASS_SRC、解像度変換処理モジュールからの出力の引き取りを指示するTAKE_DSTのいずれか、あるいはPASS_SRCとTAKE_DSTとの組合せを保持する。

【0048】

ステップS306では、StatusをPASS_SRCで初期化し、原領域の最初のラインを解像度変換処理モジュールに引き渡すよう指示する。そして、ステップS306の後は、Statusの値がENDになるまでステップS308からS312までの操作を繰り返す(ステップS307)。

20

このように状態変数Statusを導入するのは、拡大処理の場合には1ラインの入力から複数のラインの出力が得られること、周辺の画素を必要とする処理方法では最初の数ラインの入力からは出力が得られないこと、縮小処理の場合には複数ラインの入力で1ラインの出力しか得られないこと等、画像処理の違いによる事情に同時に対処するためである。状態変数を導入することで、ドライバはどんな処理方法を使っても解像度変換処理モジュールの返す状態にしたがって、入力の引き渡し、出力の引き取りをしさえすれば良い。そしてこのように1ラインずつの処理というインターフェースを実現することで、ドライバは任意の大きさの領域(任意のライン数単位の領域)の任意の倍率での拡大/縮小を、画像解像度変換処理モジュールを使って統一的に処理できるようになる。

30

【0049】

ところで、全ての解像度変換処理において1ラインずつの処理を実現するためには、解像度変換処理モジュール内に解像度変換処理に必要なバッファを用意する必要がある。幅がRW画素である原領域を周辺B画素を必要とする方法で解像度変換する場合には、少なくともRW+B+B画素分の幅のものがB+B+1ライン分必要となる。解像度変換処理モジュールではこれらのバッファに順次入力されたデータを蓄え、バッファが一杯になった段階で解像度変換処理を開始する。そして必要に応じてバッファをシフト(最も古いデータを捨てて次の入力を格納する)して、次のラインの処理を開始する。なお、データの取り込みの際には、先に求めたCL、CRの値を使ってバッファの両端を調整する。

【0050】

40

解像度変換処理モジュール内のバッファの様子を示したのが図15である。

図15において、(a)は空の状態、(b)は最初の1ライン目がバッファ1505に取り込まれた状態、(c)から(e)は2から4番目のラインがバッファシフトを行いながら次々とバッファ1505に取り込まれていった状態を示している。(f)に至り解像度変換処理モジュール内の全てのバッファは入力データによって満たされた状態となり、画素1506の処理が領域1507の画素を用いて行われる。各画素の処理は(g)から(i)に示したように右に1画素ずつずれて(1508, 1510, 1512)行き、それに伴って使用される領域も右に1画素ずつ移動して行く(1509, 1511, 1513)。そして、バッファ1503の処理が終わったならば、結果を出力し、このラインを使っての処理が終わったならば、(j)に示すように、バッファをシフトして新たなライン

50

を取り込んで、画素 1 5 1 4 の処理を領域 1 5 1 5 の画素を使って開始する。

【 0 0 5 1 】

図 1 4 にステップ S 3 0 8 における処理の流れを示す。

ステップ S 1 4 0 1 では指定領域の全ての処理が終わったかどうかを判定する。これは既に何ライン出力したかを調べることで実現できる。そして全ての処理が終了しているのならば、状態変数 Status に E N D をセットして終了する。この結果、ステップ S 3 0 7 によって原領域での処理の終了が検出され、ステップ S 3 0 3 によって次の領域の処理が開始される。

【 0 0 5 2 】

逆にステップ S 1 4 0 1 によって原領域での全ての処理が終わっていないと判定されたならば、状態変数 Status の値を調べて (ステップ S 1 4 0 3)、もしこの時の状態が P A S S _ S R C を含んでいれば、次のデータがドライバより渡されて来ているので、ステップ S 1 4 0 4 においてバッファのシフトを行い、ステップ S 1 4 0 5 において渡された次のラインをバッファに取り込む。そして最初のラインが渡された時に上端のコピーが必要であれば、C T に格納されたライン数分、バッファに格納されたラインデータのコピーを行う (ステップ S 1 4 0 6 ~ S 1 4 0 9)。

10

【 0 0 5 3 】

ラインデータの取り込が終わったならば、全てのバッファがラインデータで満たされていることを確認して (ステップ S 1 4 1 0)、1 ライン分の処理を行う (ステップ S 1 4 1 1)。この処理にはこれまで述べて来たように様々な方法を使うことができるが、拡大なのか縮小なのかそもそもステップ S 1 4 1 1 が実行されたのかにより出力ラインが作成されない場合があることに留意する必要がある。このためステップ S 1 4 1 2 において出力ラインが作成されたかを調べ、作成されていたならばステップ S 1 4 1 3 によって Status に T A K E _ D S T をセットし、そうでなければステップ S 1 4 1 4 によって Status に 0 (E N D、P A S S _ S R C、T A K E _ D S T のいずれでもまた組合せでもない値) をセットしておく。

20

【 0 0 5 4 】

そして、次の処理に次のラインデータの入力が必要であるかをステップ S 1 4 1 5 において判定し、もしそうであればステップ S 1 4 1 6 において次のデータが存在するかを判定する。もし現在のラインが原画像の下端であり次のデータが取得できない場合には、ステップ S 1 4 1 7 でバッファシフトを行い、ステップ S 1 4 1 8 で最後に入力されたラインデータをバッファにコピーする。また、現在のラインが画像の下端に達していなければ、ステップ S 1 4 1 3 または S 1 4 1 4 で設定した Status に P A S S _ S R C を追加して処理をステップ S 1 4 0 1 へ戻す。

30

【 0 0 5 5 】

こうして設定された状態変数 Status の値はステップ S 3 0 9 及び S 3 1 1 で調べられ、値に応じた対応を行った後にステップ S 3 0 7 へ戻る。

さて、ステップ S 3 0 8 において設定された Status がもし P A S S _ S R C を保持していたならば、次回に解像度変換処理モジュールを呼び出す際に引き渡す入力データを現在の次のラインに更新する (ステップ S 3 1 9)。また、ステップ S 3 0 8 において設定された Status がもし T A K E _ D S T を保持していたならば、解像度変換処理モジュールからの出力をあらかじめ用意した出力バッファに書き込む (ステップ S 3 1 2)。なお、ステップ S 3 1 0 と S 3 1 2 は、同グループ内では Status の値によりどちらか一方のみ実行されることもあり、両方とも実行されることもある。

40

【 0 0 5 6 】

以上、図 4 に示した画像解像度変換処理の流れを中心に、本発明の第 1 の実施の形態について述べた。ここで説明したように、本実施の形態によれば、従来の問題点が解決され、様々な種類の画像解像度変換処理方法をプリンタドライバに対して統一したインターフェイスで組み込むことが可能となり、解像度変換処理のドライバへの組み込みが可能となった。

50

【 0 0 5 7 】

(第 2 の実施の形態)

第 2 の実施の形態を図 1 6 から図 1 9 に基づき説明する。

第 1 の実施の形態では原画像を縦横ともに拡大する例を示したが、本発明の適用はこの場合に限定されるものではない。拡大と縮小の組み合わせにより倍率テーブルの作成手順のみ若干異なることになるが、基本的な処理手順は全く同じである。本実施の形態では縮小の場合の手順を示すが、倍率テーブルの作成部分を中心に記述する。例として挙げるのは、図 5 に示した画像 5 0 0 (縦 4 画素、横 6 画素) を図 1 6 に示したサイズの画像 1 6 0 0 (縦 3 画素、横 4 画素) に変換する場合である。この例では画像 5 0 0 が縦横にそれぞれ $3/4$ 、 $4/6$ 倍に縮小される。

10

【 0 0 5 8 】

図 1 7 は、この例での倍率テーブルの内容を示したものであり、倍率の逆数で表されている。

画像 5 0 0 の各列はそれぞれ X 方向の倍率テーブル 1 7 0 1 にしたがって $1/2$ 、 1 、 $1/2$ 、 1 倍され、各行はそれぞれ Y 方向の倍率テーブル 1 7 0 2 にしたがって 1 、 $1/2$ 、 1 倍されることが示されている。そしてこれらのテーブルによって、画像 5 0 0 における画素 5 0 1 は、画像 1 6 0 0 における画素 (1 , 1) の一部 1 6 0 1 に、画素 5 0 2 は画素 (3 , 1) の一部となることが分かる。一般に縮小の場合には、複数の原画素により変換後の 1 画素が生成されることとなる。この様な場合、変換後の各画素は、これを構成する複数の原画素の中から 1 画素を選ぶ (間引き) 方法や複数の原画素を混ぜ合わせる方法等によって作成されることが一般的であるが、本実施の形態では縮小の方法にも特に制限はなく拡大と同じインターフェイスで画像の縮小が可能である。

20

【 0 0 5 9 】

図 1 8 は、倍率テーブル 1 7 0 1、1 7 0 2 の作成方法を示した図面である。図中 S H、S W はそれぞれ解像度変換前の画像の縦と横の画素数 (例では S H = 4、S W = 6) を示し、D H、D W はそれぞれ解像度変換後の画像の縦と横の画素数 (本例では、D H = 3、D W = 4) を示している。また T Y [i] は Y 方向の倍率テーブル 1 7 0 2 の i 番目の要素を示し、T X [i] は X 方向の倍率テーブル 1 7 0 1 の i 番目の要素を示している。なお割算は整数演算であり結果は切り捨てられる (例えば、 $4/3 = 1$)。

【 0 0 6 0 】

処理の流れを説明すると、まずステップ 1 8 0 0 で演算の誤差を格納する変数 e を 0 で初期化する。次にステップ S 1 8 0 1 では変数 i を 1 ずつ増やしながらから D H - 1 までステップ S 1 8 0 2 と S 1 8 0 3 を繰り返す。ステップ S 1 8 0 2 では Y 方向の倍率テーブルの各要素の値を蓄積された計算誤差を考慮しながら求め、ステップ S 1 8 0 3 では計算誤差の更新を行う。この結果、ステップ S 1 8 0 0 から S 1 8 0 3 によって Y 方向の倍率テーブル 1 7 0 2 が作成される。

30

【 0 0 6 1 】

同様にステップ S 1 8 0 4 で演算の誤差を格納する変数 e を再度 0 で初期化し、次にステップ S 1 8 0 5 では変数 i を 1 ずつ増やしながらから D W - 1 までステップ S 1 8 0 6 と S 1 8 0 7 を繰り返す。ステップ S 1 8 0 6 では X 方向の倍率テーブルの各要素の値を蓄積された計算誤差を考慮しながら求め、ステップ S 1 8 0 7 では計算誤差の更新を行う。この結果、ステップ S 1 8 0 4 から S 1 8 0 7 によって X 方向の倍率テーブル 1 7 0 1 が作成される。

40

【 0 0 6 2 】

倍率テーブル 1 7 0 1、1 7 0 2 はこの方法によって作成されたテーブルである。倍率テーブルの各要素は個々に示したように整数値である。倍率テーブルを使う本実施の形態は整数倍の組み合わせにより任意の縮小の解像度変換ができるようになっている。このことは、整数分の 1 倍の画像サイズの変更しかできない方法であっても、非整数分の 1 倍の画像サイズ変更が可能になることを示している。なお、倍率テーブルの各要素の値の合計は縮小の場合は処理前の縦横それぞれの画素数に一致する。例えば、X 方向の倍率テーブル

50

1701の各要素の合計は $6(2+1+2+1)$ となり、縮小前の画像500の横の画素数6と一致し、Y方向の倍率テーブル1702の各要素の合計は $3(1+2+1)$ となり、縮小前の画像500の縦の画素数4と一致する。

【0063】

図19は、縮小の場合の対応領域の計算方法の手順である。ここでは、座標(DRX, DRY)を左上の点とする縦DRH画素、横DRW画素からなる領域に対応する原領域を倍率テーブルTXとTYを使って計算する。拡大時に求めていたOXとOYは縮小時にはあまり意味がないので、本例では0を格納している。ステップS1901からS1906は、領域1201の左上の点のY座標を求める操作である。縦方向の倍率テーブルTY[i]の各要素を順に加算しながら(ステップS1904)、倍率テーブルのインデックスiと開始点のY座標値DRYと比較することで(ステップS1903)、開始点の元となる画素のY座標RYをテーブルの各要素の加算値から求めている(ステップS1906)。

10

【0064】

ステップS1907からS1909は、領域1201の縦の画素数を求める操作である。縦方向の倍率テーブルTY[i]の各要素を更に順に加算しながら(ステップS1907)、倍率テーブルのインデックスiと値DRY+DRHと比較することで(ステップS1908)、領域1201の縦の画素数RHをテーブルの各要素の加算値と先に求めたRYの値から求めている(ステップS1909)。

【0065】

ステップS1910からS1915は、領域1201の左上の点のX座標を求める操作である。横方向の倍率テーブルTX[i]の各要素を順に加算しながら(ステップS1913)、倍率テーブルのインデックスiと開始点のX座標値DRXと比較することで(ステップS1912)、開始点の元となる画素のX座標RXをテーブルの各要素の加算値から求めている(ステップS1915)。

20

【0066】

ステップS1916からS1918は、領域1201の横の画素数を求める操作である。横方向の倍率テーブルTX[i]の各要素を更に順に加算しながら(ステップS1917)、倍率テーブルのインデックスiと値DRX+DRWと比較することで(ステップS1916)、領域1201の横の画素数RWをテーブルの各要素の加算値と先に求めたRXの値から求めている(ステップS1918)。

30

【0067】

領域1201の開始点座標と大きさが求まったならば、次に必要な周辺画素の数と原画像の大きさを考慮して、ドライバに要求する領域1201+1203の開始点とその大きさを求めるが、この手順は第1の実施の形態と全く同じであるため説明は省略する。そしてこれまで説明した手順により画像1206を作成できるパラメータが揃ってしまえば、どんな解像度変換処理でも利用できる。図14に関しては縮小の場合も拡大と同じに動作するが、縮小の場合はステップS1417に入り込まない限り、Statusには必ずPASS_SRCが含まれることになる。

【0068】

以上説明したように、縮小の場合も倍率テーブルの作成手順と変換後の領域から変換前の領域を作成する手順を若干手直しするだけで、拡大の時と全く同様に解像度変換処理を扱うことができる。

40

(第3の実施の形態)

第3の実施の形態を図20から図21に基づき説明する。

【0069】

第1の実施の形態では原画像を縦横ともに拡大する例を、第2の実施の形態では原画像を縦横ともに縮小する例を示したが、本発明の適用はこれらの場合に限定されるものではない。本例として挙げるのは、図5に示した画像500(縦4画素、横6画素)を図20に示したサイズの画像2000(縦6画素、横4画素)に変換する場合である。本例では画像500が縦には $6/4$ 倍に拡大され、横には $4/6$ 倍に縮小される。

50

【 0 0 7 0 】

図 2 1 は、本例での倍率テーブルの内容を示したものであり、X 方向は倍率の逆数、Y 方向は倍率が示されている。

画像 5 0 0 の各列はそれぞれ X 方向の倍率テーブル 2 0 0 1 にしたがって $1/2$, 1 , $1/2$, 1 倍され、各行はそれぞれ Y 方向の倍率テーブル 2 0 0 2 にしたがって 2 , 1 , 2 , 1 倍されることが示されている。そしてこれらのテーブルによって、画像 5 0 0 における画素 5 0 1 は画像 2 0 0 0 における領域 2 0 0 1 に、画素 5 0 2 は画素 2 0 0 2 となることが分かる。

【 0 0 7 1 】

このように拡大と縮小が混在する場合には、第 1 の実施の形態と第 2 の実施の形態とで示した倍率テーブルの作成手順と変換後の領域から変換前の領域を作成する手順を組み合わせることで、画像 1 2 0 6 を作成するための全パラメータを容易に得ることができ、その後は拡大と縮小の方法を順に適用することで目的の画像が得られることが容易に想像できる。

10

【 0 0 7 2 】

以上説明したように、拡大と縮小が混在する場合も倍率テーブルの作成手順と変換後の領域から変換前の領域を作成する手順を若干手直しするだけで、他の場合と全く同様に解像度変換処理を扱うことができる。

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

20

【 0 0 7 3 】

また、本発明の目的は、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ムコードを記録した記憶媒体を、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。

【 0 0 7 4 】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROM などを用いることができる。

30

また、コンピュータが読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 OS（オペレーティングシステム）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。

【 0 0 7 5 】

さらに、記憶媒体から読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ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わる CPU など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。

40

【 0 0 7 6 】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図 2 2 のメモリマップ例に示す各モジュールを記憶媒体に格納することになる。すなわち、少なくとも倍率テーブル作成モジュール、対応領域計算モジュール、サイズ変更処理モジュール、および（他のプログラムとの）インターフェースモジュールの各モジュールのプログラムコードを記憶媒体に格納すればよい。更に、これらモジュールを解像度変換モジュールとし

50

て組み込んだプリンタドライバを、記憶媒体に格納してもよい。

【0077】

【発明の効果】

以上説明したように、本発明によれば、簡単な操作及び手順で、非整数倍率を含む任意の倍率に対応できる画像解像度変換方法及びそれを実現する画像解像度変換モジュールが提供できる。

特に、コンピュータ上でソフトウェアとして実現されるデジタルフィルタ系の拡大縮小方法であっても、統一されたインターフェースによってプリンタドライバに効率良く組み込まれる画像解像度変換方法及びそれを実現する画像解像度変換モジュールが提供できる。

【0078】

更に、上記画像解像度変換モジュールが組み込まれたプリンタドライバを提供できる。具体的な適用によれば、どんな解像度変換の方法であっても統一されたインターフェイスでプリンタドライバへの組み込みを可能にすることにより、各処理方法それぞれにおいて非整数倍率を含む任意の倍率に対応できないという問題、バンド間で出力結果に不連続が生じるという問題、任意のライン数単位での処理に対応できないという問題があってもこれらを解消することができ、プリンタドライバ内でこれらを使用することができるようになる。

【0079】

また、全ての処理方法が任意の矩形領域を処理単位として利用できるようになるので、メモリ量の都合によってはバンドという処理単位を更に細かく分割して処理することもできるようになるという効果がある。また、同様の理由で印刷方向の変化(portraitとlandscape)によるバンド分割の方向の変化に柔軟に対応できるという効果もある。さらには、各処理方法によらずドライバからの呼び出し方法が統一されることになるのでドライバ側の処理が簡略化でき、したがってドライバ本体のプログラムサイズを小さくできる、ドライバ本体のコードに誤りが入りにくくなるなどの効果もある。

【図面の簡単な説明】

【図1】本発明の実施の形態のシステム構成例を示す図である。

【図2】本実施の形態を含む印刷処理部の構成を示すブロック図である。

【図3】画像データに関する描画処理の構成を示すブロック図である。

【図4】画像解像度変換処理の手順例を示す図である。

【図5】原画像の例を示す図である。

【図6】拡大時の変換後の画像の例を示す図である。

【図7】拡大時の倍率テーブルの例を示す図である。

【図8】拡大時の倍率テーブル作成の手順例を示す図である。

【図9】変換前後の領域の対応を示す図である。

【図10】領域分割の例を示す図である。

【図11】領域分割された原画像上の領域を示す図である。

【図12】原領域の調整を説明する図である。

【図13A】対応領域作成の手順例を示す図である。

【図13B】対応領域作成の手順例を示す図である。

【図14】サイズ変更処理の手順例を示す図である。

【図15】バッファの使用状態を説明する図である。

【図16】縮小時の変換後の画像の例を示す図である。

【図17】縮小時の倍率テーブルの例を示す図である。

【図18】縮小時の倍率テーブル作成の手順例を示す図である。

【図19A】対応領域作成の手順例を示す図である。

【図19B】対応領域作成の手順例を示す図である。

【図20】拡大と縮小が組み合わされた場合の変換後の画像の例を示す図である。

【図21】拡大と縮小が組み合わされた場合の倍率テーブルの例を示す図である。

【図22】記憶媒体上のメモリマップ例を示す図である。

10

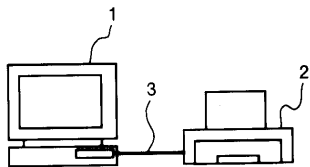
20

30

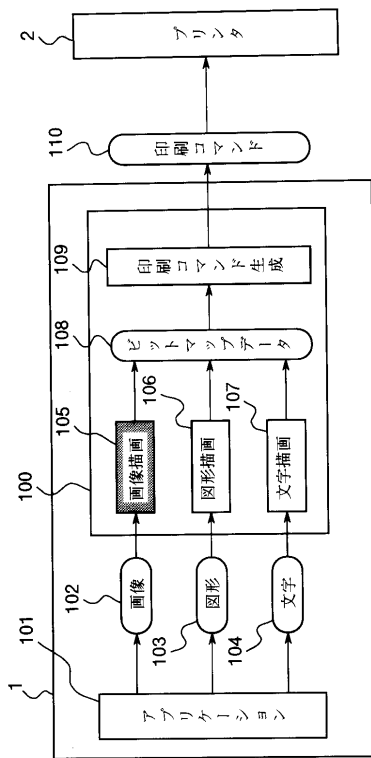
40

50

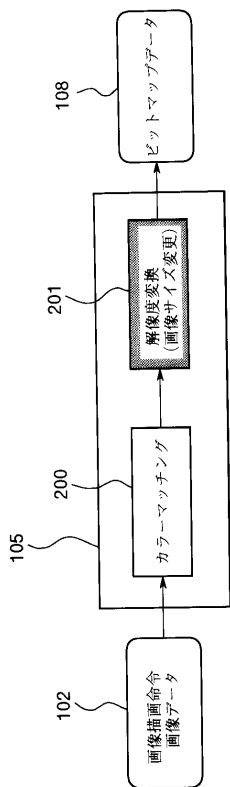
【 図 1 】



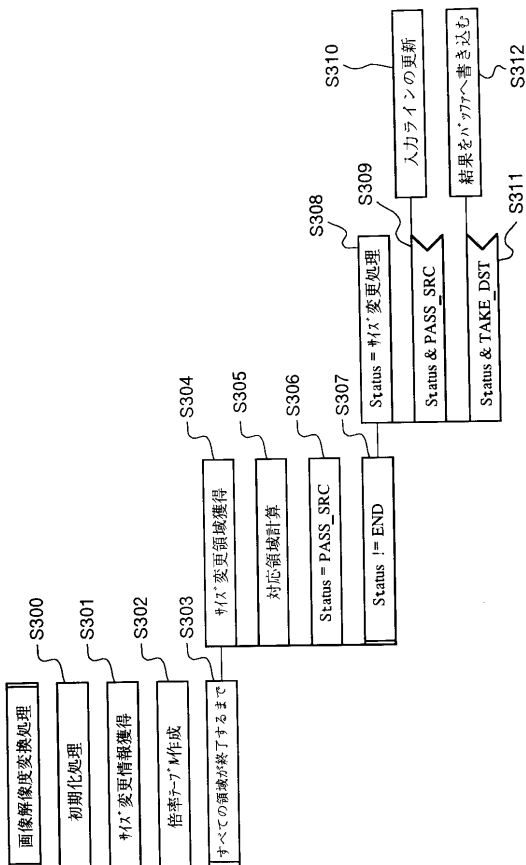
【 図 2 】



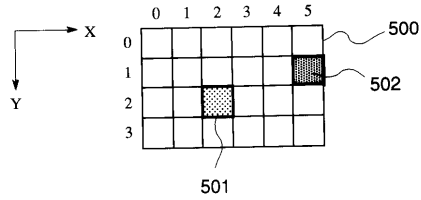
【 図 3 】



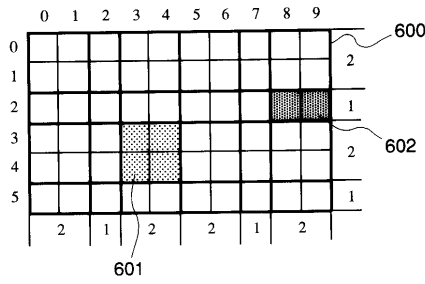
【 図 4 】



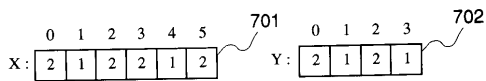
【 図 5 】



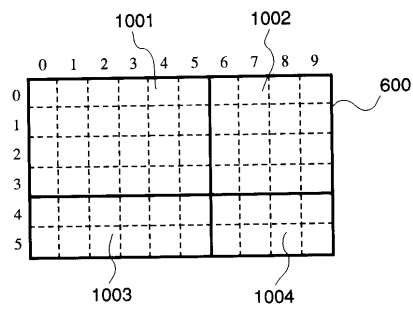
【 図 6 】



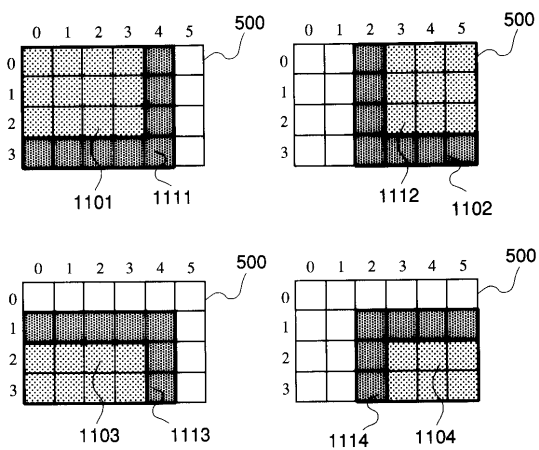
【 図 7 】



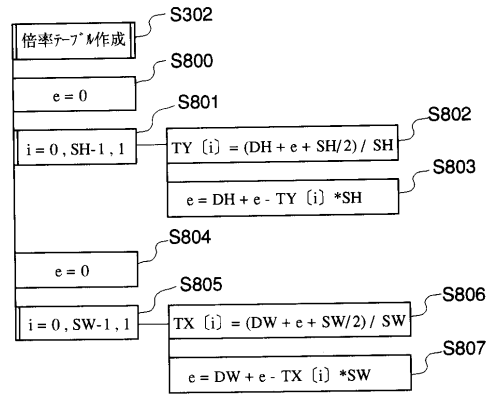
【 図 10 】



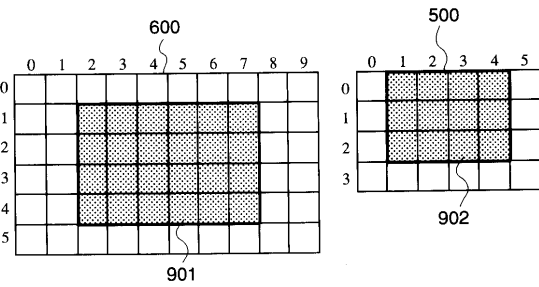
【 図 11 】



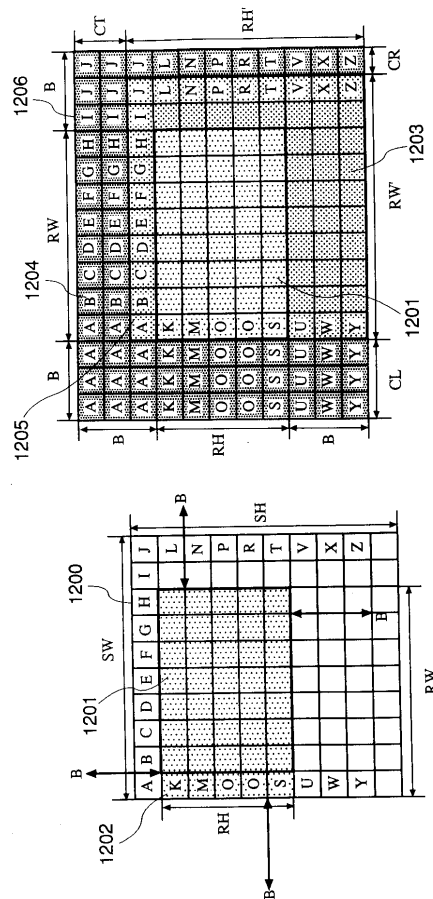
【 図 8 】



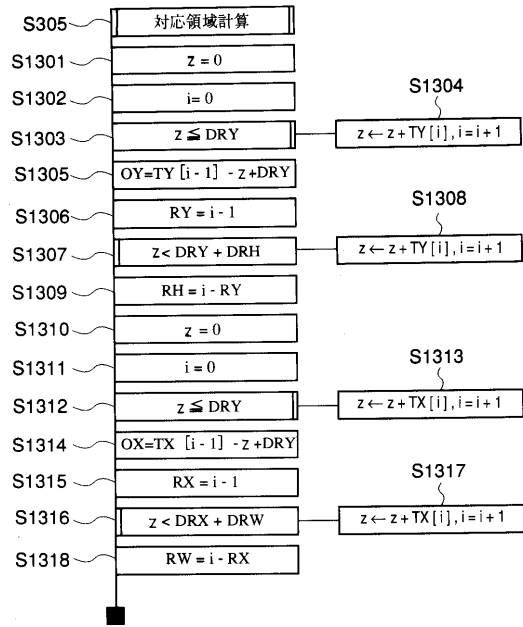
【 図 9 】



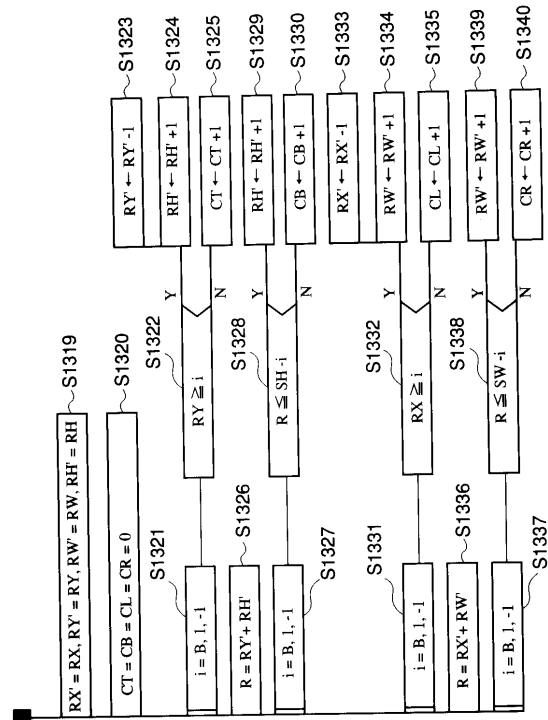
【 図 12 】



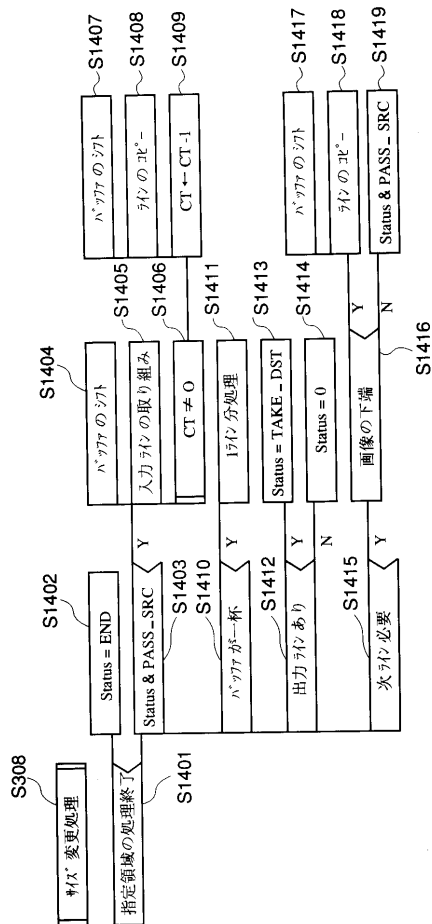
【図13A】



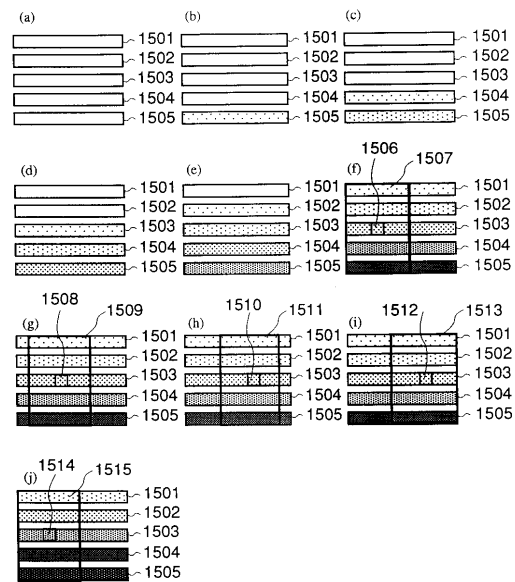
【図13B】



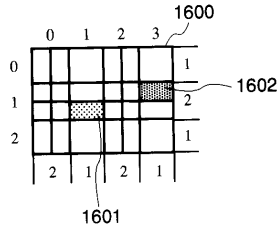
【図14】



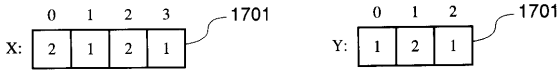
【図15】



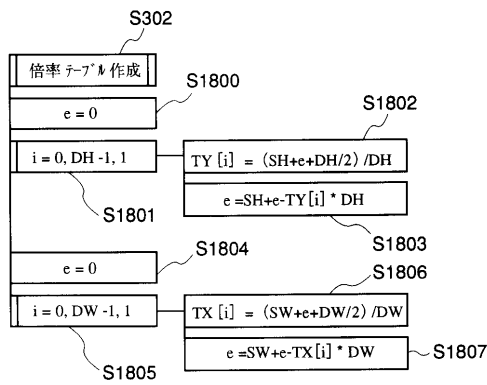
【 図 1 6 】



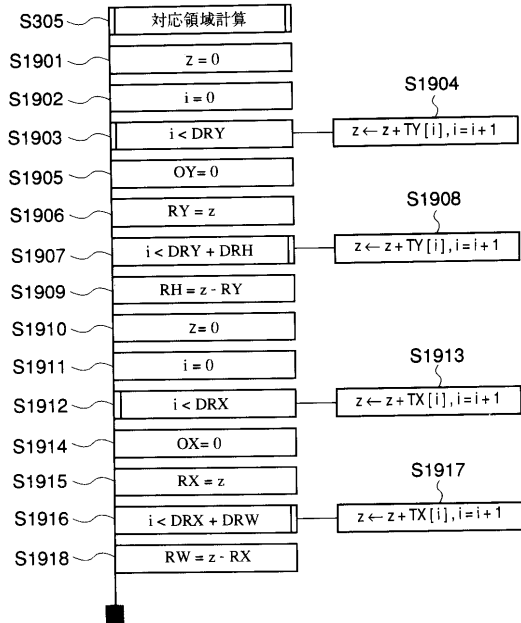
【 図 1 7 】



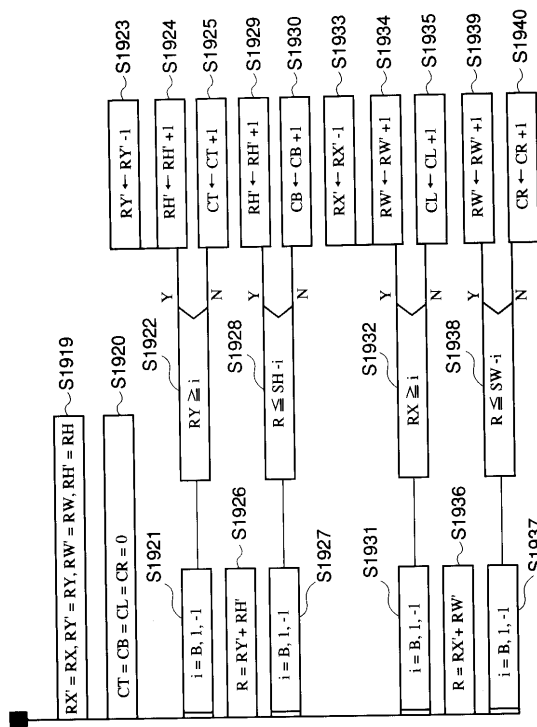
【 図 1 8 】



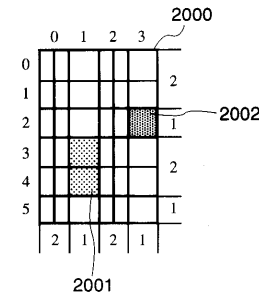
【 図 1 9 A 】



【 図 1 9 B 】



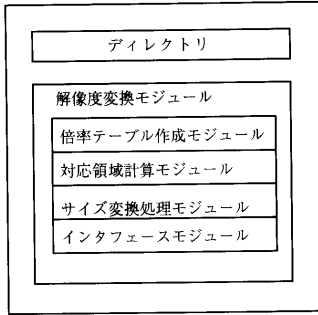
【 図 2 0 】



【 図 2 1 】



【 図 2 2 】



フロントページの続き

(58)調査した分野(Int.Cl.⁷, DB名)

H04N 1/38 - 1/393

G06T 3/40