

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3673536号

(P3673536)

(45) 発行日 平成17年7月20日(2005.7.20)

(24) 登録日 平成17年4月28日(2005.4.28)

(51) Int. Cl.<sup>7</sup>

F I

H O 4 N 1/60

H O 4 N 1/40

D

B 4 1 J 5/30

B 4 1 J 5/30

Z

H O 4 N 1/46

H O 4 N 1/46

Z

請求項の数 10 (全 27 頁)

(21) 出願番号	特願平6-277830	(73) 特許権者	000001007
(22) 出願日	平成6年11月11日(1994.11.11)		キヤノン株式会社
(65) 公開番号	特開平8-139953		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成8年5月31日(1996.5.31)	(74) 代理人	100090538
審査請求日	平成13年11月7日(2001.11.7)		弁理士 西山 恵三
前置審査		(74) 代理人	100096965
			弁理士 内尾 裕一
		(72) 発明者	清水 治夫
			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	金田 孝之
		(56) 参考文献	特開平04-067964(JP,A)
			特開平07-137353(JP,A)
			最終頁に続く

(54) 【発明の名称】 印刷装置及び印刷方法

(57) 【特許請求の範囲】

【請求項1】

外部装置から入力した描画コマンドに応じたレンダリング処理を実行し、印刷する印刷装置であって、

描画コマンドをハードウェアによってレンダリング処理する第一レンダリング手段と、

描画コマンドをソフトウェアによってレンダリング処理する第二レンダリング手段と、

前記描画コマンドを解析して中間情報を生成する生成手段と、

前記描画コマンドに対してハードウェアのレンダリング処理によるバンディング処理が可能であるか否かを判断する判断手段と、

前記判断手段でバンディング処理が不可能であると判断された場合に、階調または解像度を落とし、プリンタエンジンへ送出する前に1ページ分の描画コマンドに対してレンダリング処理を行うフルペイント処理を実行する実行手段を有し、

前記実行手段によって実行されるフルペイント処理は、前記描画コマンドから前記第一レンダリング手段によって処理することができない高度な所定の論理描画コマンドが検出された場合、該高度な所定の論理描画コマンドを前記第二レンダリング手段によってレンダリングし、

前記判断手段は、前記生成された中間情報のレンダリング時間を算出することにより、バンディング処理が可能であるか否かを判断することを特徴とする印刷装置。

【請求項2】

前記レンダリング処理は、前記生成された中間情報に基づくビットマップデータを保存

10

20

するためのメモリ領域に描画することを特徴とする請求項 1 に記載の印刷装置。

【請求項 3】

前記判断手段が前記描画コマンドに対して前記バンディング処理が行えると判断した場合、前記バンディング処理は Y M C K カラーオブジェクトをバンドラスタメモリにレンダリングし、

前記判断手段が前記描画コマンドに対して前記バンディング処理が行えないと判断した場合、前記フルペイント処理は 1 ページ分の描画コマンドに対してレンダリング処理を実行し、前記レンダリング処理によって R G B カラーオブジェクトを R G B ページバッファにレンダリングすることを特徴とする請求項 1 または 2 に記載の印刷装置。

【請求項 4】

前記フルペイント処理は、前記描画コマンドからハードウェアによってレンダリング処理することができない前記高度な所定の論理描画コマンドが検出された場合、前記高度な所定の論理描画コマンドを Y M C K カラーから R G B カラーへ逆変換した後に、R G B カラー空間で前記高度な所定の論理描画コマンドに基づいてレンダリングすることを特徴とする請求項 1 乃至 3 のいずれかに記載の印刷装置。

【請求項 5】

前記判断手段によって前記バンディング処理が行えると判断された場合、前記バンディング処理は、Y M C K カラーオブジェクトをバンドラスタメモリにレンダリングし、

前記判断手段によって前記バンディング処理が行えないと判断された場合、前記フルペイント処理は 1 ページ分の描画コマンドに対してレンダリング処理を実行し、該フルペイント処理は Y M C K カラーオブジェクト及び該 Y M C K カラーオブジェクトからレンダリングされたデータをそれぞれ Y M C K カラーから R G B カラーへ逆変換した後に、R G B カラー空間で前記高度な所定の論理描画コマンドに基づいてレンダリングし、その後、該演算結果を Y M C K カラーへ変換することを特徴とする請求項 1 乃至 4 のいずれかに記載の印刷装置。

【請求項 6】

外部装置から入力した描画コマンドに応じたレンダリング処理を実行し、印刷する印刷方法であって、

描画コマンドをハードウェアによってレンダリング処理する第一レンダリング工程と、

描画コマンドをソフトウェアによってレンダリング処理する第二レンダリング工程と、

前記描画コマンドを解析して中間情報を生成する生成工程と、

前記描画コマンドに対してハードウェアのレンダリング処理によるバンディング処理が可能であるか否かを判断する判断工程と、

前記判断工程でバンディング処理が不可能であると判断された場合に、階調または解像度を落とし、プリンタエンジンへ送出する前に 1 ページ分の描画コマンドに対してレンダリング処理を行うフルペイント処理を実行する実行工程を有し、

前記実行工程によって実行されるフルペイント処理は、前記描画コマンドから前記第一レンダリング工程によって処理することができない高度な所定の論理描画コマンドが検出された場合、該高度な所定の論理描画コマンドを前記第二レンダリング工程によってレンダリングし、

前記判断工程は、前記生成された中間情報のレンダリング時間を算出することにより、バンディング処理が可能である否かを判断することを特徴とする印刷方法。

【請求項 7】

前記レンダリング処理は、前記生成された中間情報に基づくビットマップデータを保存するためのメモリ領域に描画することを特徴とする請求項 6 に記載の印刷方法。

【請求項 8】

前記判断工程が前記描画コマンドに対して前記バンディング処理が行えると判断した場合、前記バンディング処理は Y M C K カラーオブジェクトをバンドラスタメモリにレンダリングし、

前記判断工程が前記描画コマンドに対して前記バンディング処理が行えないと判断した

10

20

30

40

50

場合、前記フルペイント処理は1ページ分の描画コマンドに対してレンダリング処理を実行し、前記レンダリング処理によってRGBカラーオブジェクトをRGBページバッファにレンダリングすることを特徴とする請求項6または7に記載の印刷方法。

【請求項9】

前記フルペイント処理は、前記描画コマンドからハードウェアによってレンダリング処理することができない前記高度な所定の論理描画コマンドが検出された場合、前記高度な所定の論理描画コマンドをYMKKカラーからRGBカラーへ逆変換した後に、RGBカラー空間で前記高度な所定の論理描画コマンドに基づいてレンダリングすることを特徴とする請求項6乃至8のいずれかに記載の印刷方法。

【請求項10】

前記判断工程によって前記バンディング処理が行えると判断された場合、前記バンディング処理は、YMKKカラーオブジェクトをバンドラスタメモリにレンダリングし、

前記判断工程によって前記バンディング処理が行えないと判断された場合、前記フルペイント処理は1ページ分の描画コマンドに対してレンダリング処理を実行し、該フルペイント処理はYMKKカラーオブジェクト及び該YMKKカラーオブジェクトからレンダリングされたデータをそれぞれYMKKカラーからRGBカラーへ逆変換した後に、RGBカラー空間で前記高度な所定の論理描画コマンドに基づいてレンダリングし、その後、該演算結果をYMKKカラーへ変換することを特徴とする請求項6乃至9のいずれかに記載の印刷方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、ホストコンピュータ等の情報処理装置から入力する入力情報を解析して中間情報を生成し、該中間情報に対してレンダリング処理（描画処理）を実行して高精細、高階調に印刷する印刷装置及び印刷方法に関する。

【0002】

【従来の技術】

最近の高機能ワークステーションやパーソナルコンピュータの出現により、フルカラーによる文字、図形、イメージデータのハンドリングが、容易に行なえる環境が整った。その結果、カラーを用いた文書、OHP（オーバヘッドプロジェクタ）、スライド、アート、デザイン等の広範な分野でカラー情報が利用されている。

【0003】

しかし、上記のカラー論理描画は色深さを持った情報を、それも3、4面分演算する必要があるため、非常にコストの高くつく処理であるという課題があった。

【0004】

また、カラーPDLプリンタにおいては製品が二極化され、すなわちインクジェットや熱転写型に代表される低価格カラープリンタにおいて、ページメモリは白黒プリンタと同様にYMKK4色それぞれに1bit分を保有する。カラーイメージや色文字、色指定による塗りつぶしでは基本的にディザ、あるいは色精度を追求する際には誤差拡散法等により、疑似的に解像度を犠牲にして色階調を再現する。

【0005】

一方、カラーLBP等に代表される高価格カラープリンタは、1ピクセルにおいてYMKK各色で複数の階調/濃度（例えば各色256階調）を表現可能である。この印刷装置においては、上記疑似階調処理を必要としないで、指定された色をそのまま記録装置内部で保持し、プリンタにエンジンに送出すればよい。しかし本記録装置においては多値YMKKカラーを格納するページバッファが必要となり、解像度300DPI、階調が各色8bit、最大紙サイズA4とすると、ページメモリとして1M×4色×8bit=32MBのメモリが必要であるため、非常にコストの高くつく処理であるという課題があった。

【0006】

そのため、後者の方が高精細な画像が印刷可能であるが、物理的な印刷装置+コントロー

10

20

30

40

50

ラの価格は前者をはるかにしのぐものになる。

【0007】

【発明が解決しようとする課題】

このように、ホスト計算機側におけるカラー情報を用いたアプリケーションが、広範な分野に渡り利用されている。しかしながらホスト側で作成したカラー情報を、印刷装置に記録する際には従来では、ホスト側のCPUパワー（処理機能）を利用し、ホスト側で文字、イメージ、図形を記録装置の解像度に合わせて、イメージに展開した後、カラープリンタに送るといふ、いわゆるダンププリンタまたはビデオプリンタと呼ばれる利用形態が一般的であった。この処理方式はプリンタ側の機構をシンプルにし、ホスト側で多くの処理を実行する点に特徴があるが、カラー情報を取り扱う場合はそのデータ量の多さから、通信に多くの時間をさかれ、スループットが非常に落ちる場合があるという問題がある。

10

【0008】

一方、白黒プリンタにおいては、ページ記述言語（Page Description Languageの略であり、以後PDLと略す。）方式という、ホスト側から文字、図形、イメージを言語として送り、プリンタでPDL言語を解釈し、各種情報をプリンタの解像度に対応したラストメモリ中にスキャン変換（走査変換）することにより、ページイメージを生成する方式が一般的である。本スキャン方式をカラープリンタにも適用したカラーPDLプリンタが最近普及し始めている。

【0009】

しかしながら、カラーPDLプリンタにおいては、従来の白黒PDLプリンタの言語アーキテクチャー（構造）を踏襲しており、白黒すなわち1bit情報に対する、すでにレンダリング（描写）された情報（デスティネーション）とこれからレンダリングする図形、イメージ、文字情報（ソース）との間で、SET、OR、XOR等の演算をビット毎に施すというアーキテクチャーであった。しかし、カラープリンタにおいては各カラープレーン、例えばRGB（レッド、グリーン、ブルー）毎に深さを持っており（例えば、1、2、4、8ビット）、従来SET、OR、XORの演算をビット毎に演算すると、所望の色が得られないという問題があり、そのためカラー論理描画という（Add、Sub、Max、Min、Blend）等のビット深さを考慮した論理演算機能が導入されている。

20

【0010】

本発明の目的は、上記の問題点を解決するためになされたもので、論理描画をある程度の精度（階調、解像度）を保持しながらも低コストで実現する印刷装置および印刷方法を提供することを目的とする。

30

【0011】

本発明の目的は、上記の問題を解決するためになされたもので、論理描画を効率的に実現する印刷装置および印刷方法を提供することを目的とする。

【0012】

【課題を解決するための手段】

上記目的を達成する本発明の印刷装置は以下に示す構成を備える。即ち、外部装置から入力した描画コマンドに応じたレンダリング処理を実行し、印刷する印刷装置であって、描画コマンドをハードウェアによってレンダリング処理する第一レンダリング手段と、描画コマンドをソフトウェアによってレンダリング処理する第二レンダリング手段と、前記描画コマンドを解析して中間情報を生成する生成手段と、前記描画コマンドに対してハードウェアのレンダリング処理によるバンディング処理が可能であるか否かを判断する判断手段と、前記判断手段でバンディング処理が不可能であると判断された場合に、階調または解像度を落とし、プリンタエンジンへ送出手前に1ページ分の描画コマンドに対してレンダリング処理を行うフルペイント処理を実行する実行手段を有し、前記実行手段によって実行されるフルペイント処理は、前記描画コマンドから前記第一レンダリング手段によって処理することができない高度な所定の論理描画コマンドが検出された場合、該高度な所定の論理描画コマンドを前記第二レンダリング手段によってレンダリングし、前記判断手段は、前記生成された中間情報のレンダリング時間を算出することにより、バンディ

40

50

ング処理が可能である否かを判断することを特徴とする。

【0024】

【実施例】

以下、図面を参照して本発明の実施例を詳細に説明する。

【0025】

図1は本発明の一実施例のカラー印刷装置1000の画像処理系の基本構成を示す。この図を用いて本発明実施例における処理の大まかな流れを説明する。

【0026】

(全体構成)

図1において、1は、カラーアプリケーションとしてカラー情報を作成し、このカラー情報に対応するカラー、データをPDL形式に変換して、変換したPDLデータをカラー印刷装置1000の記録装置コントローラ14に送出するホスト計算機(ワークステーション)である。ここで、ホスト計算機1と記録装置14間にPDLデータが流れる。このPDLデータの通信状態はシリアル、ネットワーク、バス接続等何であっても問題はないが、パフォーマンス的には高速通信路であることが望ましい。記録装置コントローラ14へ送られたカラーPDLデータは入力バッファ(データ入力用バッファ2)に一時格納され、プログラムROM6内のPDLコマンド解析プログラムによって、入力データがスキャンされる。3は文字のビットパターンまたはアウトライン情報、および文字ベースラインや文字メトリック情報を格納するフォントROMであり、文字の印字に際して利用される。4のパネルI/O P(入出力プロセッサ)は、プリンタ本体に装着されるパネルにおけるスイッチ入力の検知やLCD(液晶ディスプレイ)への表示を司る、I/Oプロセッサおよびファームウェア(パネルI/Oプロセッサ)であり、低価格のCPUが利用される。拡張I/F(インタフェース)5は、プリンタの拡張モジュール(フォントROM、プログラムROM、RAM、ハードディスク)とのインターフェース回路である。

【0027】

プログラムROM6は本発明に係る図2、図6、図8、図10、図11、図15、図17に示すような処理手段(ソフトウェア)を格納するメモリであり、CPU12がこのソフトウェアに従って本カラーPDLデータを読み込み処理を実行する。7はソフトウェアのための管理領域のRAMであり、入力されたカラーPDLデータを解析して中間データ形式(ページオブジェクト)に変換したデータや、グローバル情報等が本管理用RAM7に格納される。

【0028】

色変換ハードウェア8は、通常ワークステーション(WS)で利用されているモニタの表色系のレッド、グリーン、ブルーのRGB(加法混色)からプリンタのインク処理で用いるイエロー、マゼンタ、シアン、ブラックのYMKK(減法混色)への変換を行なうハードウェアである。本色変換処理は色精度を追求すると、非線形なログ変換・ $3 \times 3$ のマトリックス演算等で、演算パワーを大変要するものである。ハード的にはテーブル・ルックアップ処理により高速化を図っている。この色変換パラメータは最初プリンタエンジンにとって最適なものに調節されているが、ホスト側から色変換方式を変更する要求があれば、テーブルの値を変更することにより、色変換アルゴリズムをユーザ定義のそれに変えるのは可能である。また、処理時間を犠牲にすれば、CPU12によってソフトウェアによる演算も可能である。

【0029】

ハードレンダラ(ハードウェアレンダリング回路)9は、カラーレンダリング処理をASIC(特定用途向けIC)ハードウェアで実行することにより、カラープリンタ(例えば、レーザビームプリンタ)13のビデオ転送に同期して実時間でレンダリング処理を行い、少ないメモリ容量でのバンディング処理を実現するものである。ページ(バンド)バッファ10は、PDL言語によって展開されるイメージを格納する領域であり、上述のバンディング処理を行なうための最低2バンドのメモリ(ページ幅\*256または512位のバンド高さ\*プレーン数として3(RGB)または(YMKK)\*ビット深さ)か、また

10

20

30

40

50

はバンディング処理を出来ない際に、LBP（レーザビームプリンタ）のようにプリンタエンジンに同期してイメージを転送する必要のある装置では、解像度かつ/または色階調を落したフルカラービットマップメモリを確保する必要がある。しかし、インクジェットプリンタのように記録ヘッドの移動をコントローラ側が制御可能な機構の場合には、上記2バンドのメモリが最低限あればよい。

#### 【0030】

ディザパターン15はバンディングにより、少ないメモリ容量で高速に印字している際には必要ないが、デグレード処理（あるバンド中にリアルタイムでイメージ展開できないほど印字データが存在するか、またはページオブジェクトが管理用RAM7から溢れてページの途中で強制印字を行なう）において、少ないビット深さで色精度を再現するために利用する。

10

#### 【0031】

プリンタインタフェース（I/F）11はカラープリンタ（カラー記録装置）13、例えばカラーLBPとの間で、ページバッファ10の内容をプリンタ側の水平・垂直同期信号に同期して、ビデオ情報を転送する。あるいは、カラーインクジェットプリンタにおけるヘッド制御および複数ラインのヘッドサイズに合わせたビデオ情報の転送を行なう。さらに、本プリンタインタフェース11ではカラープリンタ13との間でプリンタへのコマンド送信やプリンタからのステータス受信を行なう。CPU（中央演算処理装置）12は記録装置コントローラ14内部の処理を制御する演算装置である。カラープリンタ13はコントローラ14から送出されるビデオ信号を記録媒体にカラー印刷する。カラープリンタ13としては電子写真式によるカラーLBPでもインクジェット方式のプリンタであってもよい。

20

#### 【0032】

次に、図2に示すインタプリタ処理を示すフローチャートと図3に示す中間データフォーマットと図4に示すレンダリングモデルと図5に示す各種マスク情報に関する説明図を用いて、本実施例のインタプリタ処理（データ・フィルタリング・ダスク）を以下説明する。

#### 【0033】

図1中の矢印は、各種描画情報に関するホスト計算機1からプリンタエンジン13までの処理の流れを示す。なお、図2のフローチャートに示すプログラムは、プログラムROM6に格納されており、CPU12によって実行される。

30

#### 【0034】

まず、ステップ101において、CPU12は、割り込み処理等により入力バッファ2にカラーPDLデータをとり込み、次にステップ102では入力されたPDLコマンドを言語仕様に応じてインタプリット（解釈）する。インタプリットした結果、ステップ103において入力データが描画コマンド、例えば文字、直線、イメージ描画である際には、ステップ104において図3に示す中間データフォーマットの中間データを作成して、管理用RAMに記憶する。この中間データは、後述するレンダラ処理が実行されて、ページバッファ10にビットマップラリとして記憶される。

#### 【0035】

図3は、PDLデータをインタプリタした結果作成される中間データを管理用RAM7に記憶する際のフォーマットであり、これらの情報は、1ページ分格納された後排紙命令（Form Feed）により、R、G、Bプレーン毎に後述するレンダリング処理を実行し、カラー画像が生成される。

40

#### 【0036】

図において、20には例えば後述するソフトレンダリングの実行を示すフラグ（0）或いは後述するハードレンダリングの実行を示すフラグ（1）が格納される。

#### 【0037】

21は、後述するマスク情報151が格納され、22は後述するバックグラウンド情報152が格納され、23は後述する論理描画方式153が格納される。そして、24は次の中

50

間データへ移行するためのネクストポイント情報154が格納される。

【0038】

以後の説明のため、本実施例におけるレンダリングモデルを図4を用いて簡単に説明する。本モデルは、各種描画データの幾何的な情報、すなわちどの部分が描画対象かというON、OFFの1bitで表現可能なマスク情報151、そしてマスクをどのような色で塗るかというカラー多値のバックグラウンド情報152、及び論理描画方式153(SET、OR、XOR、BLEND、ADDe t c)の三要素により構成される。任意形状でのクリップを行なう際には形状データ(マスク情報)にまずクリップを施し、クリップ後残った領域のみをマスクとする。その結果レンダリングされたイメージ154に示す。このカラー多値バックグラウンド情報152は、どのようにハンドリングするか決定される。

10

【0039】

本実施例においてサポートするマスク情報151として、ランレングス(X方向の一つのスキャンライン)、エッジが交差しない凸多角形、ビットマップイメージ、ビットマップフォントからなる。これらの情報からわかるように、これらのマスク情報は高速なハードウェア・レンダリングに適した構造とし、例えば図5(A)の五角形は本ステップ104において、図5(B)に示すような交差しない5個の三角形に分割する(この例では塗りつぶしは、even-oddルールを適用)。又図5(C)に示す、ラインの接続処理部においては本モジュールにおいてDDAアルゴリズムを適用して管理用RAM7内の作業領域にラインの接続情報(round、miter、triangle)を考慮して展開した後、最終的な外部形状を、Yスキャンライン毎にmin x、max xをペア情報

20

【0040】

最終的に生成される各マスクオブジェクトは、フルページメモリよりも少ないメモリ容量でのレンダリング、すなわちバンディングを行なうためのページメモリを複数のバンド(高さが2の冪乗が望ましく、512ドット位が最適である。)に分割し、各マスクオブジェクトをバンド毎にソーティングし、各バンド内で図5(D)に示すリンクリストを構成する。この際バンドに跨るマスク情報(例えば図5(B)に示す多角形)に関しては、各バンドで多角形情報を共有化する。各バンドに分割したマスクに対しステップ105において、レンダリング時に必要となるデータのデコード時間とレンダリング時間を、各バンドにおいてページオブジェクトを作成する毎に加算する。この情報をそれぞれバンドi毎

30

に保持し、pred\_\_decode(i)、pred\_\_render(i)とする。デコード時間は、作成されたオブジェクトのほぼデータ量に比例する。しかし、バンド3における三角形1、4(図5(B)参照)のデコード時間は、前のバンド2の開始点からのバンド3における多角形の開始点のオフセットを求める時間が余分に必要となる。

【0041】

レンダリング時間は、バンド内のマスク面積×バックグラウンドの色の深さ×色プレーン数×論理描画の種類による演算ファクタにより計算される。

【0042】

図2に戻り、入力されたデータが描画コマンドでない場合には、ステップ106でデータが各種属性(バックグラウンド、論理描画)設定コマンドかどうかを判定する。もしYESであれば、ステップ107において対応する処理を実行するが、これらはそれぞれハード(又はソフト)レンダラが読み込み可能なデータ形式(ページオブジェクト)に変換するものである。

40

【0043】

又本実施例のように、カラー論理描画の機能はハードウェアでサポートされないので、そのような属性設定コマンド情報を検知すると、フルペイントフラグ(full-p-lag)をステップ107でセットする。その結果として、強制的に印刷の解像度かつ/または階調を落として、後述するフルペイントモードでのレンダリングを行なう。

【0044】

また、同様に、Flood Fill等の命令(点指定塗りつぶし)はバンディングによ

50

るレンダリングを実行できないため、そのような情報を検知すると、フルペイントフラグ (full-p-lag) をセットする。結果として、強制的に印刷の解像度かつ/又は階調を落して、後述するフルペイント (デグレード) モードでのレンダリングを行なう。

【0045】

バックグラウンド情報152は、マスクに対してどのようにカラー・濃淡をつけるかを示す。バックグラウンドの種類として、イメージとして繰り返しを行わずにマスクに張りつくバックグラウンドパターンと、パターンを縦・横方向に繰り返してマスクに張り付けるタイルパターンが指定可能である。本実施例においてはカラー印刷装置を想定しているため、イメージ、タイルはカラー情報として指定可能である。

【0046】

ステップ108は、例えばデバッグ処理等の目的で状態をダンプ処理する。

【0047】

ステップ109では上記インタプリタ120の処理が1ページ分のPDLコマンド解析を終了したかを判定し、終了していれば後述するレンダラタスクに処理が移行するが、そうでなければステップ102において、次のコマンド解析を繰り返す。

【0048】

ここまでは基本的にPDLからページオブジェクトへの、データ・フィルタリング・タスクであり、これ以降の処理はページバッファ10への描画を行なうレンダリング・タスクである。この両者のタスクは、特にレンダリング・タスクにおいて実時間処理が要求されるため、リアルタイムOS上で別タスクとして実装され、かつ後者のレンダリング・タスクは前者よりもプライオリティを高く設定されて動作する。

【0049】

次に、図6に示すフローチャートを用いて、レンダリング処理 (レンダリングタスク) を説明する。

【0050】

なお、図6のフローチャートに示すプログラムは、プログラムROM6に格納されており、CPU12によって実行される。

【0051】

図2のステップ109で1ページ分のPDLコマンド解析を終了したとCPU12が判断すると、本プログラムを起動し、ステップ110において、ページオブジェクトをレンダリングする前処理として、バンドレンダリング (バンディング) 処理が可能か否かを判定する。このバンディング処理が不可能な場合を以下に列挙する。

【0052】

・上述したFlood Fill命令等がページ中に存在する。

【0053】

・大量のイメージ入力により管理用RAM7の情報があふれた (メモリデグレード)。

【0054】

・カラープリンタ13が電子写真式LBP、LED (発光ダイオード) プリンタの様に、一度紙を給紙して記録を開始すると、バンディング処理はプリンタ13へのビデオ信号転送とバンドへのレンダリングとを並行処理する必要があり、そのため図2のステップ105で計算されたバンド毎のレンダリング時間pred-decode(i)、pre-render(i)に関し、どれかのバンドが、所定の閾値をオーバーする (タイムデグレード)。

【0055】

上記の条件に適合すると、バンディングを実行できないため、ステップ112で解像度または階調を落としてページバッファ10のメモリ中に、フルペイントメモリを確保し、後述するレンダリング (カラー論理描画) を実行する。一方、インクジェットプリンタ等の記録ヘッドの移動をコントローラ側で制御出来る形態の装置においては、レンダリング時間に (上記3番目の条件) については上述の限りではなく、レンダリングスピードが低下すると、ヘッドの移動を遅らせることにより、バンディング処理が可能である。

10

20

30

40

50

## 【 0 0 5 6 】

なお、ステップ 1 1 0 で、上記の条件に適合しないと判断すると、後述するバンドレンダリングを 1 1 1 で実行する。

## 【 0 0 5 7 】

次に、上記ステップ 1 1 1 のバンドレンダリング処理を図 7 にバンドレンダリングの概念を示す図を図 8 のバンドレンダリング処理のフローチャートを用いて以下、説明する。

## 【 0 0 5 8 】

ステップ S 9 0 1 で C P U 1 2 は、P D L 解析タスク 1 2 0 により管理用 R A M 7 に作成された中間データのマスク情報から Y 座標におけるスキャンライン情報 ( x m i n 、 x m a x ) を抽出し、カレントのバックグラウンド情報、論理描画モードを参照して対応するバックグラウンド情報をバンド・ラスタメモリ 1 0 に書き込む。

10

## 【 0 0 5 9 】

そして、同じバンドのすべてのマスクに対応すべく図 3 に示すネクストポインタ情報 1 5 4 をたどり、リンクが終了するまでレンダリングを実行する ( ステップ 9 0 2 ) 。

## 【 0 0 6 0 】

図 7 では、各バンドのデータは連続したメモリ空間に存在するようになっているが、実際には各中間データがどこのメモリにあってもよく、バンド内の情報はリスト構造となっている。また、レンダリングを行なっているページ番号に対して、ページ解析しているページ番号は同じ必要がなく、一般的に、ページ解析中のページ レンダリング中のページ、の関係が成り立つ。

20

## 【 0 0 6 1 】

ハードレンダラ 9 はマスク情報 1 5 1、バックグラウンド情報 1 5 2、論理描画の方法 1 5 3 に従いバンド番号 i のページオブジェクトに対しレンダリングを行なうと共に、並行してプリンタエンジン 1 3 から送られてくる水平同期信号にあわせ、プリンタ I / F 1 1 を通じて既にレンダリング済みのバンド番号 i - 1 のバンドラスタ情報 ( ビットマップデータ ) をプリンタエンジン 1 3 にビデオ信号として送出する ( ステップ 9 0 3 ) 。

## 【 0 0 6 2 】

この処理をすべてのバンドに対して実行することにより、印字が実行される ( ステップ 9 0 4 ) 。

## 【 0 0 6 3 】

本実施例では、バンドラスタ情報として 2 バンド持ち、レンダリング中 ( 第 N + 1 バンド ) とエンジンへ転送中 ( 第 N バンド ) のバンドを所定時間内に切替え、リアルタイムのレンダリング処理を実現している。バンド高さを大きくできれば、局所的なオブジェクトの集中に対しても、時間デグレードの面で発生頻度を低下させることが可能である。

30

## 【 0 0 6 4 】

なお、本システムでは、カラープリンタを想定しているため、ページバッファ 1 0 には 4 面、すなわち Y M C K のプレーンが存在し、各色情報をプレーン毎にレンダリングする。

## 【 0 0 6 5 】

ここでハードレンダラでサポートできる論理描画としては、ソース・パターン ( S )、ディステーション・パターン ( D ) とすると、以下の三種類である。これらは、パターン S と D 間で両方の情報を入力し、両者間で演算してパターン D に設定するような、演算パワーが必要な処理はサポートされない。これは、カラーの 4 つのプレーンを参照する必要があり、更に各プレーンが 4 から 8 ビットの際データの演算量が非常に大きくなる点に起因する。従って、P D L データが以下の三種類に該当する場合は、図 3 のステートフラグを 1 にし、演算パワーが必要な P D L データの場合には図 3 のステートフラグを 0 ( ソフトレンダリング実行 ) にする。

40

## 【 0 0 6 6 】

- ・上書き ( D = S )
- ・透過、D に描画しない ( D = D )
- ・白 ( D = 0 )

50

## 【 0 0 6 7 】

また通常は、高度なカラー論理描画データはホスト計算機から頻繁に送出されないと仮定し、本ハードでサポート可能なデータ処理をなるべく高速化するため、またはプリンタの色モデルはY M C Kであることから、バックグラウンドの持つカラー情報をY M C Kとする。図2のステップ107においてバックグラウンド情報を解析してデータを管理用RAMに格納する際にホストから送られてくるRGBデータに対して、色変換ハードウェア8を用いてY M C Kカラーに変換し、バックグラウンド情報として保持しておく。色変換の際にハードウェアでなくソフトウェアで実現する形態も考えられるが、処理の高速化のためにはハードを用いるのが望ましい。

## 【 0 0 6 8 】

このようにしてハードウェアはマスク情報、バックグラウンド情報、論理描画の方法に従いバンドの番号*i*のページオブジェクトに対しレンダリングを行うと共に、並行してプリンタ13から送られてくる水平同期信号にあわせ、プリンタインタフェース11を通じて既にレンダリング済みのバンド番号*i*-1のバンド情報をプリンタ13にカラービデオ信号(Y M C K)として送出する。本バンディング処理は、上述の3個のカラー論理描画からなるページデータは、十分に高速印字が可能である。また、現在一般的に広く利用されているページ記述言語Pは、このシンプルなカラー描画論理に準拠しているので、本バンディング処理により多くのデータは高速にレンダリング可能である。

## 【 0 0 6 9 】

次に、上記ステップ112のカラー論理描画処理(フルペイントレンダリング処理)をカラー情報の流れを示す図9および図10のフローチャートを用いて説明する。本論理描画の際には、ホスト計算機1から入力されたカラーバックグラウンド情報(RGBデータ)401は、レンダリングハードまたはソフトウェアが取り扱えるページオブジェクト形式(RGB obj)405に変換する。

## 【 0 0 7 0 】

なお、図10のフローチャートに示すプログラムは、プログラムROM6に格納され、CPU12によって実行される。

## 【 0 0 7 1 】

まず、以下に示す高度な論理描画を実現するために、LBP等の実時間レンダリングを要求される印刷装置では、バンドレンダリングではなく、解像度かつ/または階調を落したフルページバッファ10上へのレンダリングとなる。また、ハードレンダラ9は処理の簡素化および高速化を要求されるため、レンダリング時にランレングスや凸多角形情報のリアルタイム解像度変換は実行できない。そこで、以下に示す処理が必要であるが、インクジェットプリンタ等においては、この限りではない。

## 【 0 0 7 2 】

レンダリングの前処理として、例えば600DPI(ドット/インチ)から300DPIに解像度を落とす際に、ランレングスは2ライン分をまとめて1つのランレングスとし、かつ凸多角形は頂点情報の再計算を実行する。これをページバッファ中のすべてのマスク情報に対して、インタプリタタスク201によって実行する。ランレングスは、例えば600DPIにおける、2つのライン*i*、*i*+1のX座標の開始・終了点をそれぞれ*x<sub>l</sub>*(*i*)、*x<sub>r</sub>*(*i*)、*x<sub>l</sub>*(*i*+1)、*x<sub>r</sub>*(*i*+1)とすると、新規300DPIでの一つの開始、終了点は以下ようになる。

## 【 0 0 7 3 】

$$\text{new-}x_l(i) = \min \cdot 1/2(x_l(i), x_l(i+1)),$$

$$\text{new-}x_r(i) = \max \cdot 1/2(x_r(i), x_r(i+1))$$

## 【 0 0 7 4 】

イメージに関しては、ページオブジェクトのイメージ情報自身は変化せずに、*x*、*y*方向へのスケージングファクタをそれぞれ1/2とする。

## 【 0 0 7 5 】

一方、ページバッファを階調を落としても、レンダラが容易に対応できるように、1、2

10

20

30

40

50

、4、8ビットレンダリングをサポートするので、インタプリタの前処理の負荷は大きくない。

【0076】

本処理以降のフルペイントレンダリング（カラー論理描画）に関する部分の処理手順を、図10のフローチャートに示す。ステップ501においてインタプリタで変換されたオブジェクト（中間データ）を入力し、入力されたオブジェクトが描画コマンドか否かをステップ502で判断する。描画コマンドでなければステップ505において、バックグラウンド情報や（論理）描画モードをカレント情報を保持するグローバル変数に代入する。次に後述のステップ507へ進む。

【0077】

もし、入力されたオブジェクトが描画コマンドであれば、ステップ503において現在の論理描画モードとステートフラグをチェックする。もし、描画モードが上書き、透過等のハードウェアによる高速レンダリングが可能な処理である場合（ステートフラグが1の場合）、ステップ506でハードレンダリングを起動する。これは既に説明済みのバンディングによる、高速処理のハードレンダラと同等の処理となる。唯一の相違は、高速ハンディング処理の場合はYMCKカラーでレンダリングし、本フルページモードはRGBカラーでレンダリングするため、あらかじめ異なるディザマトリックスをロードしておく必要がある点である。

【0078】

一方、高度な論理描画が指定された際（ステートフラグが0の場合）には、ステップ504でソフトウェアによるレンダリングを実行する。これは図9の406で示すようにフルページのページバッファ（D、ディスティネーション）407と現在のバックグラウンド・オブジェクト（S、ソース）405のカラー情報をフェッチ（取り込む）して、論理演算を施した後、フルページバッファ407に結果を格納する。ここで代表的な論理演算処理として、以下に示すようなものを各RGB成分毎に、ビット深さも考慮して演算する。ソフトウェアによるレンダリングにおいて、ソースイメージを作成するのは、ハードレンダラ9と同じアルゴリズムを適用して、下述の論理描画を適用するためであり、ページバッファアクセス・ライブラリを論理描画の方式毎に作成する。ソフトレンダリングとハードレンダリングを混在させるため、ハードレンダリングが終了すると、CPU12に割り込みが発生し、次のページオブジェクトをCPU12が続けてフェッチする仕掛けとなっている。

【0079】

- ・加算、 $D = S + D$
- ・減算、 $D = D - S$
- ・ブレンドで 値はユーザが指定、 $D = \quad \times S + (1 - \quad) \times D$
- ・最大値、 $D = \text{Max}(S, D)$
- ・最小値、 $D = \text{Min}(S, D)$

【0080】

この高度な論理描画は、ホスト計算機1で、一般的にCRT（陰極線管）ディスプレイで利用されるRGBデータ上で演算される。そのため、ホスト計算機と同じ色再現を行うには、プリンタ内部においてもRGBカラーモデル上で実現する必要がある。そのため、論理描画のフルペイントモードにおいては、ページバッファ407はRGBカラーモデルでなければならない。又、ページオブジェクト情報405も、RGBカラーモデルの必要がある。

【0081】

すでに述べたように高速バンディング処理においては、ページオブジェクト情報におけるバックグラウンド情報をYMCKカラーモデルで保持するので、本論理描画を行うために、ページオブジェクトとしてRGBカラーも持つ必要がある。その結果、ステップ507で、1ページ分のデータのレンダリング処理を終了すると、ステップ508においてページバッファ10のすべての情報をRGBからYMCKにすべて変換した後、YMCKのビデ

10

20

30

40

50

オデータをステップ509においてプリンタインタフェース11を通じてプリンタ13に送出する。

【0082】

ここで、上記ステップ504のソフトレンダリング処理を図11のフローチャートを用いて詳細に説明する。なお、図11のフローチャートに示すプログラムは、プログラムROM6に格納されており、CPU12によって実行される。図10のステップ503でスタートフラグが0であるとCPU12が判断すると、図3の21で示されるマスクポイントによって示されるアドレスからマスク情報、22で示されるアドレスからBG情報、23の論理描画情報を獲得する(ステップ851)。

【0083】

本データは基本的に、PDL解析部においてマスクデータが入力されるたびに作成され、論理描画やBGデータはカレント状態を保持し、マスク情報を結合される。

【0084】

ステップ852では、獲得したマスク情報によりハンド中に描画する最左上端X、Y座標を計算し、X、Y座標に対応するBGの多値情報(例えば32×32のbit-pixel深さのタイルの場合には)、

$tile\_address = tile\_top\_address(R, or\ G, or\ B) + (Y \bmod 32) \times tile\_width(byte\ boundary) + (X \bmod 32) \times bit\_pixel(tile) / 8$ 、及びページバッファ310における、デスティネーションアドレスを獲得し(ステップ853)、

$band\_address = band\_top\_address + Y \times band\_width(byte\ boundary) + x \times bit\_pixel(band) / 8$

によりデータの格納されているメモリアドレスを計算し、指定された論理描画にしたがって、ソース、BG、デスティネーション情報に対するレンダリング(論理描画)を実行する(ステップ854)。

【0085】

この処理をXのスキャンライン方向に増加させるため、マスクは次の情報の再計算、BG、デスティネーション情報を上記式に従って更新し(ステップ852、853)、そのスキャンが終了したかどうかを判定し、終了すればステップ856で次のスキャンライン(Y=Y+1)において、マスクが終了したかを判定し、終了すれば本処理を終了し図10のステップ507を実行する。もし終了しなければ、ステップ857でYを増加させ、その際のX座標としての再左端座標を計算し、同様な処理を繰り返すべくステップ852の処理へ移行する。

【0086】

次に、上記ステップ506のハードレンダリング処理を図12に示すハードウェア構成図を用いて詳細に説明する。なお、図12に示すハードウェア構成図は、図1のハードレンダラ9を詳細に示す。

【0087】

図12はハードウェアレンダリングを実行する実現例であり、構成を簡単に紹介すると、801のマイクロ実行解析部がマイクロコード(807)を読みだし、各ブロックに解析し生成された図3の中間情報に応じて必要なBG、マスク、情報等を切り出して、必要な情報を供給し、各ブロック802、803、805に並列に処理を行うべく起動をかける。

【0088】

802は入力されたマスク情報を、FIFOを経由して受けとり、マスク情報をRL、トラベ、ビットマップなどの情報に応じてデコード部に送り解析し、注目するX、Y座標を生成し、BG生成ブロック回路803やバンドメモリブロック805に渡して、それぞれ対応する情報を、レンダリング・ブロック806に送り出すよう、トリガをかける。

【0089】

BGパターン生成回路ブロック803は、上述のソフトレンダリングで説明したアルゴリ

10

20

30

40

50

ズムにより、指定された X、Y 位置に対する B G パターンを生成する。

【0090】

レンダリング回路ブロック 806 では、マスク、B G、ディスティネーションパターンを収集し、論理描画モードに応じてレンダリングを実行し、レンダリング結果（ビットマップデータ）をバンドメモリ 805 に格納した後、ページバッファ 10 に格納する。ここでは 3 つのパターンの最も処理の遅いプロセスに依存して、処理速度が決定される。

【0091】

こうして、マスク生成、レンダリング処理が 1 つのマスクの全領域をカバーにするまで、続行される。

【0092】

ハードレンダリングは、1 つのデータのレンダリングを終了すると、図 3 の 24 に示すネクストポイント情報を入力し、レンダリングを試みる。なお、データがハードレンダリング不可能である、すなわちステートフラグがソフトレンダリング（0）と判断されれば、CPU に割込信号及び現在の処理できなかった中間データの先頭アドレスを指定し、ソフトレンダラ（プログラム）の起動を要求する。

【0093】

最後に、図 13 を用いて、論理描画の結果を示す。この原図はカラー表示であるが、カラー表示を図面として添付できないので、便宜上ハッチングにより色の違いを表わした。図中 R はレッド（Red）、G はグリーン（Green）、B はブルー（Blue）、Y はイエロー（Yellow）、M はマゼンタ（Magenta）、C はシアン（Cyan）、W はホワイト（White）である。ここでは  $D = S + D$  のビット深さを考慮した論理演算を施した例であり、CRT の加法混色の原理を示すものである。図中の、 $Cyan = Green + Blue$ 、 $Yellow = Red + Green$ 、 $Magenta = Red + Blue$ 、 $White = Red + Green + Blue$  の加算により生成される。

【0094】

本発明の第 2 の実施例の構成を図 14 に示す。図 9 に示す上記実施例においては、バンディングによる高速レンダリング時には、Y M C K のページオブジェクトおよび Y M C K ページバッファ、高度な論理描画の際には、R G B のページオブジェクトおよび R G B ページオブジェクトを保有するという例を示した。この例では、R G B および Y M C K 両者のカラーモードのページオブジェクトを保有する必要があり、図 1 の管理用 R A M 7 の領域を多く消費する。そこで、図 14 に示す本実施例においては、高速バンディング処理の時も論理描画の際においても、ページオブジェクトおよびページオブジェクトを Y M C K で保有することにより、記録領域の減少を図った例を示す。

【0095】

図 14 における上半分の高速バンディング処理については、図 9 に示すものと同じなので説明を省略し、下半分の論理描画のパス（経路）について説明する。カラーの高度な論理描画によるレンダリングの際にはまず、カラー変換部 605 でソース（S）に対応するページオブジェクトのバックグラウンドデータ 603 を Y M C K から R G B に逆変換する。それと同様に、すでにレンダリングされたページオブジェクト（D）604 のうちでソースと論理描画する部分を同様にカラー変換部 608 で逆変換する。

【0096】

ここで問題となるのは、色変換処理 602 の色変換処理で R G B から Y M C K に変換する際に簡単なログ変換を施してれば、逆変換も容易に計算し得るが、U C R（下色除去処理）、マスキング処理を施している場合は、完全に Y M C K から精度を落とさずに R G B に逆変換するのは不可能である。また、このような変換を実時間で行うには、余りにも計算量が大きくなる。

【0097】

そこで、本第 2 実施例の処理では、各ビット深さのパターン 2、4、8 毎に Y M C K から R G B への近似変換テーブルをシュミレーションによりあらかじめ求めておき、この近似変換テーブルを用いて得られた R G B データを逆変換データとして用いる。近似変換テ

10

20

30

40

50

ブルにおいて2ビットの際はテーブルは256バイト、4ビットの際にはテーブルは65、536バイトであり、これらは実現可能なメモリサイズである。しかし、8ビットの際には余りにもデータ量が大きいので、4ビットテーブルを用いて色精度が犠牲にするか、あるいは計算時間の増大を覚悟でソフトウェアにより計算するかを、ユーザが選択できるようにするとよい。

**【0098】**

そして上述のようにして得られたRGBモデルにおけるSとDの情報を基に、論理演算606を実行したのち、得られたRGBデータを色変換ハードウェア8を用いてYMKKデータに変換し、ページバッファ10に格納する。格納したデータを最終的にプリンタインタフェース11を通じて、カラープリンタエンジン13にビデオ信号として転送する。

10

**【0099】**

以上説明した第1実施例によれば、このカラー印刷装置1000は、カラーページ記述情報を入力し、この入力情報を解析して記録のための中間情報を作成し、この中間情報に対してハードウェアによる高速レンダリング可能であって、ハードウェアによってサポートし切れない高度なカラー論理描画機能を、ソフトウェアによるレンダリングに切替える。これにより、色階調かつ/または解像度を低下させながらもある程度色品位を保持し、ハードウェアによる高速処理機能を生かし、少ないメモリ容量でカラー論理描画機能を実現し、通常の高速論理描画と両立することができる。

**【0100】**

また、以上説明した第2実施例によれば、高速バンディング処理の時も論理描画の際においても、ページバッファおよびページオブジェクトをYMKKで保有することにより、記録領域の減少を図ることができる。

20

**【0101】**

上記実施例では、図6に示すレンダリング処理と、バンディング可能な場合は、バンドレンダリングを実行し、バンディング不可能な場合にはフルペイントのカラー論理描画をソフトレンダリングとハードレンダリングを自動的に切換えて実行することで達成した。

**【0102】**

以下、第3実施例では、バンディング可能な場合には最高の色階調バンドレンダリングを実行し、バンディング不可能な場合には、解像度かつ/又は階調を落としたデクレードレンダリングを実行する。

30

**【0103】**

次に図15に示すフローチャートを用いて、第3実施例によるレンダリング処理(レンダリングタスク)を説明する。

**【0104】**

なお、図15のフローチャートに示すプログラムは、プログラムROM6に格納されており、CPU12によって実行される。

**【0105】**

図2のステップ109で1ページ分のPDLコマンド解析を終了したとCPU12が判断すると、本プログラムを起動し、ステップ110において、ページオブジェクトをレンダリングする前処理として、バンドレンダリング(バンディング)処理が可能かどうかを判定する。

40

**【0106】**

そして、ステップ110での判定で、バンディング処理が可能な場合には、第1実施例で説明したバンドレンダリングを実行し、ステップ110での判定で、バンディング処理が不可能な場合には、本第3実施例におけるデグレードレンダリングをステップ200で実行する。

**【0107】**

なお、ステップ110の判定処理、ステップ111のバンドレンダリング処理は第1実施例と同様なので説明を省略するが、例えば、Flood Fill等の命令(点指定塗りつぶし)はバンディングによるレンダリングを実行できないため、そのような情報をステ

50

ステップ110で検知すると、フルペイントフラグ (full-p-lag) をセットして、ステップ200において強制的に印刷の解像度かつ/又は階調を落して、フルペイント (デグレード) モードでのレンダリングを行う。

【0108】

また、ステップ111でのバンディング処理時にはページバッファに、  
 $256 (Y \text{ サイズ}) \times 4 (色) \times 8 (色 \text{ 深さ}) \times 2 \times 4800 (A4、400DPI \text{ の幅}) / 8 = 10MB$

弱を持つことで実現できる。

【0109】

またバンディング時にはページバッファに各色8bitを持つから、ホストから来るPDL情報は一般的に1、2、4、8bitであるため、対応するYCMK1、2、4、8bit情報をページオブジェクトとして格納し、レンダリング時にルックアップテーブルを介してのビット拡張処理となる。本処理は後述するディザや誤差拡散手法に比べて演算コストは非常に少ない。

【0110】

ステップ110でバンドレンダリングが不可能と判定されると、ステップ200でデグレード処理を実行するが、本第3実施例の中心となるカラー情報の流れを示す図16及び図17のフローチャートを用いて本第3実施例を説明する。

【0111】

デグレード処理はリアルタイムにレンダリング出来ないため、解像度かつ/又は階調を落したフルビットマップ (ビット深さ2or4bit) に対応するバッファ10上へのレンダリングとなる。レンダリングハード9は処理の簡素化及び高速化を要求されるため、レンダリング時にランレンクスや多角形情報のリアルタイム解像度変換は実行できない。そこで以下に示す処理がレンダリング以前に実行しておく必要があるが、BJプリンタ等においてはこの限りではない。

【0112】

レンダリングの前処理として、例えば600DPIから300DPIに解像度を落とす際、ランレンクスは2ライン分をまとめて1つのランレンクスとし、凸多角形は頂点情報の再計算を実行する。これをページバッファ中のすべてのマスク情報に対して、インタプリタ・タスク201にて実行する。ランレンクスは、例えば600DPIにおける、二つのラインi、i+1のX座標の開始・終了点をそれぞれ $x_1(i)$ 、 $x_r(i)$ 、 $x_1(i+1)$ 、 $x_r(i+1)$ とすると、新規300DPIでの一つの開始・終了点は以下のようになる。

【0113】

$$\text{new } x_1(i) = 1/2 \times \min(x_1(i), x_1(i+1), \text{new } x_r(i)) \\ = 1/2 \times \max(x_r(i), x_r(i+1))$$

【0114】

すなわち、解像度変換406は、イメージに関しては、ページオブジェクトのイメージ情報自身は変化せずに、x、y方向へのスケールングファクタをそれぞれ1/2倍する。

【0115】

一方ページバッファの階調を落としても、レンダラは1、2、4、8ビットレンダリングをハードまたはソフトでサポートするため、バックグランド情報はマスクとは異なり、特にCPUパワーの要求される処理は必要ではない。

【0116】

次に図15のステップ200のデグレードレンダリング処理を図17のフローチャートを用いて説明する。この図17のフローチャートに示すプログラムは、プログラムROM6に格納され、CPU12により実行される。ステップ701においてCPU12はインタプリタ・タスクにより解像度変換されたマスク及びバックグランド情報を入力し、入力されたオブジェクトが描画コマンドかどうかをステップ702で判断する。描画コマンドでなければステップ705において、バックグランド情報 (論理) 描画モードをカレント情

10

20

30

40

50

報を保持するグローバル変数に代入する。

【0117】

一方描画コマンドと判断された場合には、ステップ703において、マスク、バックグラウンド、論理描画情報を収集して、ハードウェアによるレンダリングを実行する（ステップ704）。この際にはページバッファは2または4bitであるため、PDLよりバックグラウンドとして8bit相当のイメージが入力されると、2または4bitへの変換のためのディザ、誤差拡散等の処理を実行する必要がある。

【0118】

なお、図10のフローチャートで説明した様に、図17のステップ704でステートフラグが0の場合には、前述したソフトレンダリングを実行し、ステートフラグが1の場合には、前述したハードレンダリングを実行する。従って、デグレードレンダリング（解像度かつ/又は階調を落としてレンダリング）する場合に、ソフトレンダリングとハードレンダリングを自動的に切替えることができる。

10

【0119】

多値ディザ法の原理を図18を用いて説明するが、ページメモリの階調より入力データの色階調が高い場合に本処理が必要となる。

【0120】

ディザ処理を説明するために、まず単純多値化の原理を多値として8bit（256レベル）入力を2bit（4値）化する、を例としてアルゴリズムを示す。

【0121】

注目画素の入力値が64未満だと0（00）、64以上128未満だと85（01）を、128以上192未満だと170（10）を、255以下だと255（11）を出力する。これは図18（A）に示すものであり、入力が属しているAREA内部で、そのAREA内の閾値（64、128、192）を利用し、出力がAREAの両端となるような2値化処理を行う。図中の太い縦線が領域の区切りを示し、下に8bitレベルおよび2bitレベル（ ）で括っているの出力値を示す。細い縦線が領域内での閾値8bitレベルを示す。

20

【0122】

この2値化処理を多値ディザに応用する例を図18（B）と（C）を参照して説明する。図18（B）で示される注目画素データと注目画素に対応するディザマトリックス図18（C）の値からその領域に適した閾値を計算し、注目画素のデータをこの閾値で2値化する。ここでディザマトリックスは、4×4のパターンとしてページバッファ上で同じパターンを繰り返す。ディザマトリックスの最大値は255 / (ビットレベル - 1) となる。入力データは拡大、縮小処理があるとすでにページメモリの解像度に変換されている。

30

【0123】

実際のディザ・アルゴリズムを以下に説明する。

【0124】

1. 入力データにおける注目画素を読みとり、どのAREAに属するかを判断する。注目画素は180であり、AREA2に属している。

【0125】

2. 対応するディザマトリックスの値を読み込み、このAREAに合致する閾値に変更する。 $threshold = 74 + 85 \times 2 = 244$

40

【0126】

3. 注目画素データが閾値以上であればこのAREAの最大値、閾値未満であればAREAの最小値を出力値とする。注目画素（180） < 閾値（244）なので、AREAの最小値（170）を出力する。

【0127】

4. 次の画素を処理する。

【0128】

この処理はハードウェア的にはルックアップテーブルにより、高速変換処理が可能である

50

。このテーブルは入力レベルが0から255のおのおのについて、4\*4のディザマトリックスの各位置においてディザ変換した2bit出力値をあらかじめ格納しておくことにより実現できる。

【0129】

この際のテーブルサイズは各YMKK毎に256\*4\*4\*2bit=1024byte分必要であり、2bitずつを図19(A)に示すポイントにより示されるディザテーブル図19(B)よりアクセスする。

【0130】

ステップ706において、1ページ分のマスクデータのレンダリング処理を終了すると、エンジンIFを通じてプリンタに各YMKKプレーン毎に水平・垂直同期信号に合わせて送出する(ステップ707)。

10

【0131】

なお、デグレート処理時にディザ処理を行わずに、バンディング処理と同様に入力されたデータをそのまま、あるいはビットを切り捨てることによりレンダリングすることも可能である。この結果特にソフトウェア処理する際には、印字品位は低下するがディザ処理よりも印字処理のスピードアップが図られ、ユーザにとってはドラフトモードとして位置付けることが可能となる。

【0132】

これは図16においてデグレート時には解像度変換406の処理は実行するが、ディザテーブル15の参照は行わない形態である。

20

【0133】

また、デグレート時の色精度を出すための疑似階調処理として、ディザ処理を例にとり説明したが、他の実現例として例えば誤差拡散方法や平均濃度保存法の処理も適用可能である。

【0134】

また、本実施例ではデグレート時に階調が落ちるという前提の元で実施例を説明したが、ページメモリをユーザが拡張RAMとして増設が可能であり、256階調分のフルメモリが獲得可能であれば、デグレート時にはディザ処理による疑似階調変換処理は必要ではなくなる。

【0135】

以上説明した様に、第3実施例によれば、カラーPDL情報の印刷を大量のフルビットマップを持たずに、多値表現可能なカラープリンタにおいて、バンディング処理可能な場合には最高の色階調によってバンディングレンダリング処理を実行し、またバンディング処理が不可能な場合にはディザ、誤差拡散を施すモード(色再現重視モード)或いは色再現を重視せずにスピード重視によるモードでのデグレートレンダリング処理を実行する。この両者のモードをユーザが指定することにより、ユーザの意図に合致したカラー印刷を行うことが可能になる。

30

【0136】

以上説明した様に本実施例によれば多値の色再現が可能なカラープリンタにおいてフル多値ビットマップメモリを用いずに、バンディング処理を行い高速にカラーイメージ展開を行い、メモリの使用量を削減することができる。

40

【0137】

なお、イメージ展開をリアルタイムに実現できないほどホスト装置側から印字データを送出されたり、ハードアクセラレータによる印字が不可能な場合にはデグレート処理として階調性を落し(8bitから2bit)フルビットマップを持つと、IM\*4(色)\*2(bit)=8MBのメモリ容量となる。

【0138】

ここで両者の処理方式による色再現性を比較すると、

【0139】

【外1】

50

バンディング方式⇒ $2^8 = 4 \cdot 2 \times 10^9$

デグレート方式⇒ $2^2 = 256$

色が可能である。これを比較すればバンディング方式ではユーザが指定された色により十分な色階調は出せるが、デグレート方式では256階調しか表現できないため、十分な階調を表現できない可能性が大きく、階調を重視する際にはデザや誤差拡散手法等の疑似階調処理を実施する。

【0140】

図20は上記実施例が適用できるカラープリンタエンジン13の一例としてのカラーインクジェットプリンタ装置1000の外観を示す。尚、図1におけるカラープリンタエンジン13以外の制御部は図20には図示していない。図20において、駆動モータ5013の正逆回転に連動して駆動力伝達ギヤ5011、5009を介して回転するリードスクリュー5005に螺旋溝5004に対してキャリッジHCが係合する。キャリッジHCはその螺旋溝5004と係合するピン（不図示）を有し、それにより、矢印a、b方向に往復移動させられる。このキャリッジHCには、インクジェットカートリッジIJCが搭載されている。5002は紙押え板であり、キャリッジHCの移動方向に沿って紙をプラテン5000に対して押圧する。5007、5008は fotocopy からなり、キャリッジHCのレバー5006のこの領域での存在を確認して、モータ5013の回転方向の切換えを行うためのホームポジション検知手段である。5016はインクジェットカートリッジIJCの記録ヘッドの全面をキャップ（覆う）するキャップ部材であり、5015はこのキャップ部材5016内を吸引する吸引手段であり、これらによりキャップ内開口5023を介して記録ヘッドの吸引回復を行う。5017はクリーニングブレードであり、5015はこのクリーニングブレードを前後方向に移動可能とする部材であり、本体支持板5018にこれらが支持されている。クリーニングブレードとしては、この形態だけではなく周知のクリーニングブレードが適用できるのは言うまでもない。また、5012は、吸引回復の吸引を開始するためのレバーであり、キャリッジHDの係合するカム5020の移動にともなって移動し、駆動モータ5013からの駆動力がクラッチ切換えの公知の伝達手段で移動制御される。これらのキャッピング、クリーニング、吸引回復は、キャリッジHCがホームポジション側の領域に来た時にリードスクリュー5005の作用によって、それらの対応位置で所望の処理が行えるように構成されているが、これら以外のタイミングで作動するようにしてもよい。

【0141】

図21は本実施例を適用可能なプリンタの構成を示す断面図であり、例えばレーザービームプリンタ(LBP)の場合を示す。

【0142】

図において、1500はLBP本体であり、外部に接続されているホストコンピュータから供給される印刷情報（文字コード等）やフォーム情報あるいはマクロ命令等を入力して記憶するとともに、それらの情報に従って対応する文字パターンやフォームパターン等を作成し、記録媒体である記録紙等に像を形成する。1501は操作のためのスイッチおよびLED表示器等が配されている操作パネル、1000はLBP本体1500全体の制御およびホストコンピュータから供給される文字情報等を解析するプリンタ制御ユニットである。このプリンタ制御ユニット1000は、主に文字情報を対応する文字パターンのビデオ信号に変換してレーザードライバ1502に出力する。レーザードライバ1502は半導体レーザー1503を駆動するための回路であり、入力されたビデオ信号に応じて半導体レーザー1503から発射されるレーザー光1504をオン・オフ切換えする。レーザー光1504は回転多面鏡1505で左右方向に振らされて静電ドラム1506上を走査露光する。これにより、静電ドラム1506上には文字パターンの静電潜像が形成されることになる。この潜像は、静電ドラム1506周囲に配設された現像ユニット1507により現像さ

10

20

30

40

50

れた後、記録紙に転写される。この記録紙にはカートシートを用い、カットシート記録紙はLBP1500に装着した用紙カセット1508に収納され、給紙ローラ1509および搬送ローラ1510と搬送ローラ1511とにより、装置内に取り込まれて、静電ドラム1506に供給される。また、LBP本体1500には、図示しないカードスロットを少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なる制御カード(エミュレーションカード)を接続できるように構成されている。

#### 【0143】

なお、本実施例を適用するプリンタは、レーザビームプリンタおよびインクジェットプリンタに限られるものではなく、他のプリント方式のプリンタでも良いことは言うまでもない。

#### 【0144】

以上説明したように、本実施例は、バンディング処理を行う場合は、当然低コスト化を実現することができ、さらにバンディング処理ができない場合にも解像度または階調を落としてフルペイント処理を行うので低コスト化を実現することができます。さらに本願発明は、従来では高度な所定の論理描画コマンドを処理しなければならない場合、ホスト側の処理機能がレンダリング処理を行うことしかできませんでしたが、本願では印刷装置におけるソフトウェアが高度な所定の論理描画コマンドに対してレンダリング処理を行えるようになりました。

さらに、カラー印刷を行う場合、第一レンダリング手段によって第一種(YMCK)の色成分でレンダリングできない高度な所定の論理描画コマンドが検出された場合、擬似階調処理によって階調数を落として第二レンダリング手段を行うため低コスト化が図れます。また、擬似階調処理によって階調数を落として第二レンダリング手段を行うため印刷品質の精度をある程度保ちながらレンダリングできるという効果が生じます。

#### 【発明の効果】

以上説明したように、本発明は、バンディング処理を行うことで低コスト化を実現するだけでなく、バンディング処理ができない場合にも解像度または階調を落としてフルペイント処理を行うので低コスト化を実現することができます。また、高度な所定の論理描画コマンドが検出された場合、印刷装置におけるソフトウェアが高度な所定の論理描画コマンドに対してレンダリング処理を行えるようになりました。

#### 【0145】

また、カラー印刷を行う場合、低コスト化が実現できると共に、描画コマンドから検出されたバンディング処理できない高度な所定の論理描画の階調数を擬似階調処理によって落としてレンダリング処理を行うので印刷品質の精度をある程度保てるという効果が生じます。

#### 【図面の簡単な説明】

【図1】本発明の一実施例のカラー印刷装置1000の基本構成を示すブロック構成図である。

【図2】本発明の一実施例のインタプリタ処理の概要を示すフローチャートである。

【図3】中間データのフォーマットを示す図である。

【図4】本発明の一実施例のイメージングモデルを示す構成図である。

【図5】本発明の一実施例における各種のマスク情報に関する説明図である。

【図6】本発明の一実施例におけるレンダリング処理全体の概要を示すフローチャートである。

【図7】本発明の一実施例におけるバンドレンダリングの概念を示す図である。

【図8】本発明の一実施例におけるバンドレンダリング処理の概要を示すフローチャートである。

【図9】本発明の一実施例の論理描画におけるカラー処理に関するブロック図である。

【図10】本発明の一実施例におけるフルペイントレンダリング処理の概要を示すフローチャートである。

【図11】本発明の一実施例におけるソフトレンダリング処理の概要を示すフローチャー

10

20

30

40

50

トである。

【図12】本発明の一実施例におけるハードレンダリング処理を行うハードウェア構成を示す図である。

【図13】本発明の一実施例のカラー論理描画の結果を示す図である。

【図14】本発明の一実施例における、論理描画時のカラー処理に関するブロック構成図である。

【図15】本発明の一実施例における、レンダリング処理全体の概要を示すフローチャートである。

【図16】本発明の一実施例におけるバンディングレンダリング処理及びデグレードレンダリング処理に関するブロック構成図である。

10

【図17】本発明の一実施例におけるデグレードレンダリング処理の概要を示すフローチャートである。

【図18】本発明の一実施例におけるディザ処理に関する説明図である。

【図19】本発明の一実施例におけるディザ処理を行うハードウェア構成を示す図である。

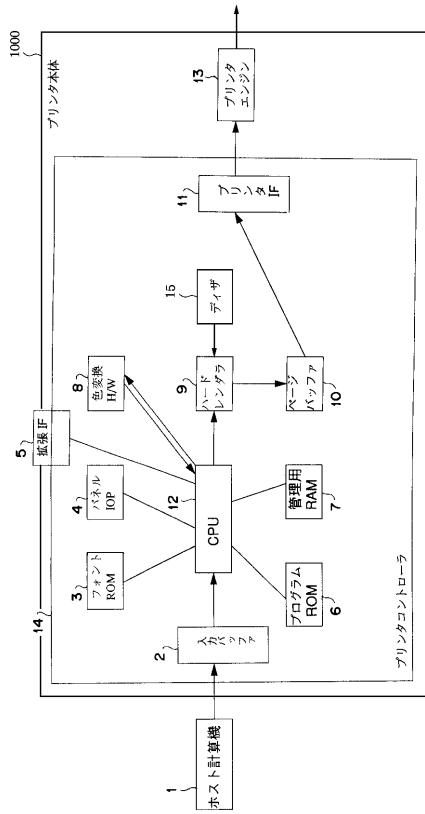
【図20】本発明を適用可能なインクジェットプリンタの外観図である。

【図21】本発明を適用可能なレーザービームプリンタの断面図である。

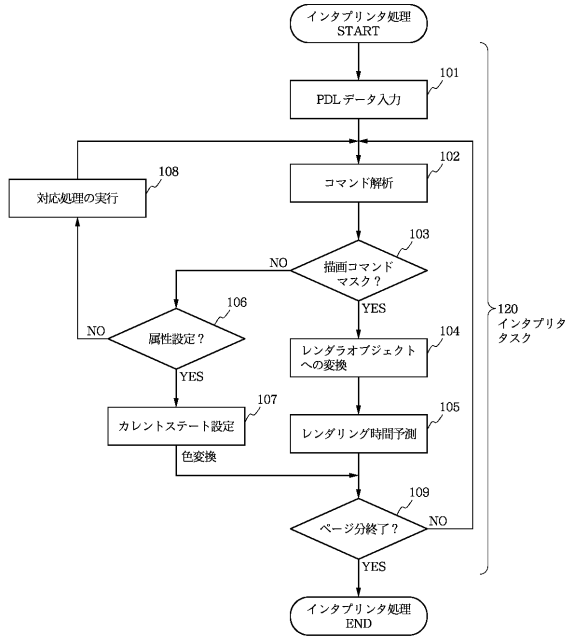
#### 【符号の説明】

- |      |              |    |
|------|--------------|----|
| 1    | ホスト計算機       |    |
| 2    | データ入力用バッファ   | 20 |
| 3    | フォントROM      |    |
| 4    | パネルI/Oプロセッサ  |    |
| 5    | 拡張I/F        |    |
| 6    | プログラムROM     |    |
| 7    | 管理用RAM       |    |
| 8    | 色変換ハードウェア    |    |
| 9    | ハードウェアレンダラ   |    |
| 10   | ページ(バンド)バッファ |    |
| 11   | プリンタインタフェース  |    |
| 12   | CPU          | 30 |
| 13   | カラープリンタエンジン  |    |
| 14   | コントローラ       |    |
| 15   | ディザパターン      |    |
| 1000 | プリンタ本体       |    |

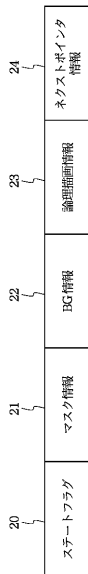
【 図 1 】



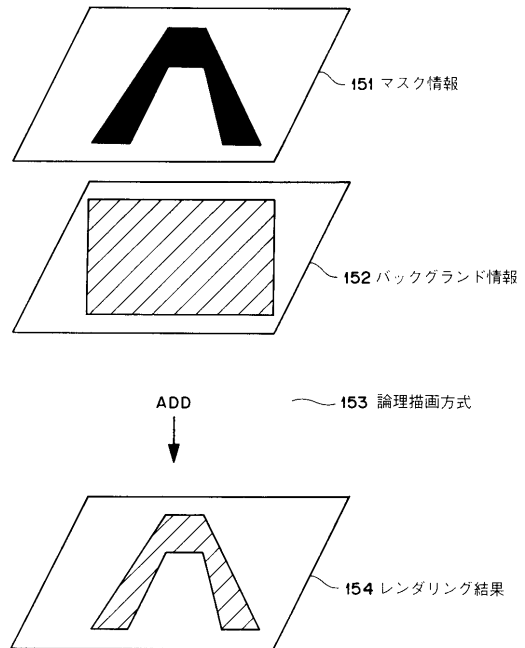
【 図 2 】



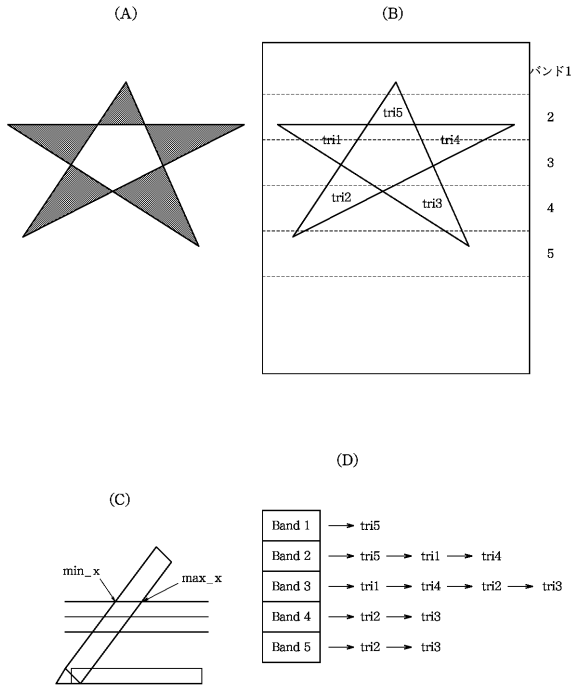
【 図 3 】



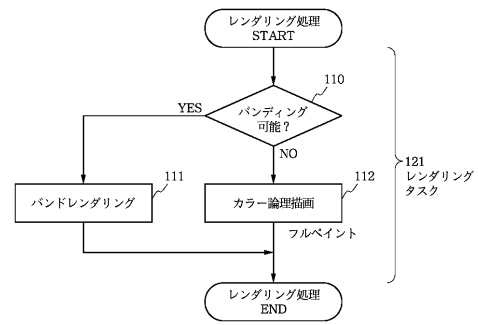
【 図 4 】



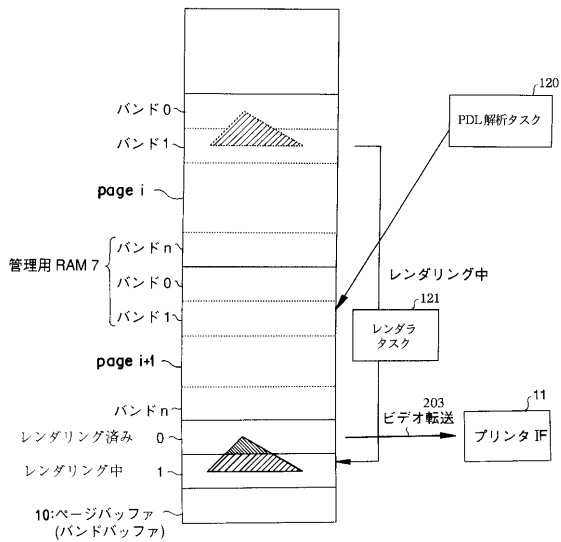
【 図 5 】



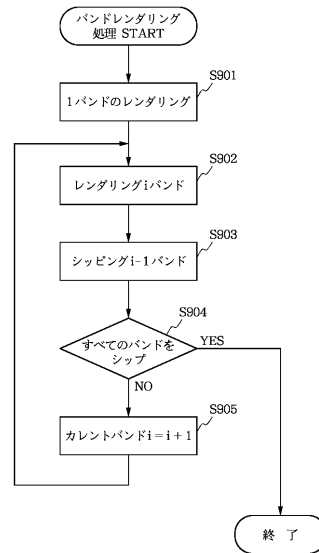
【 図 6 】



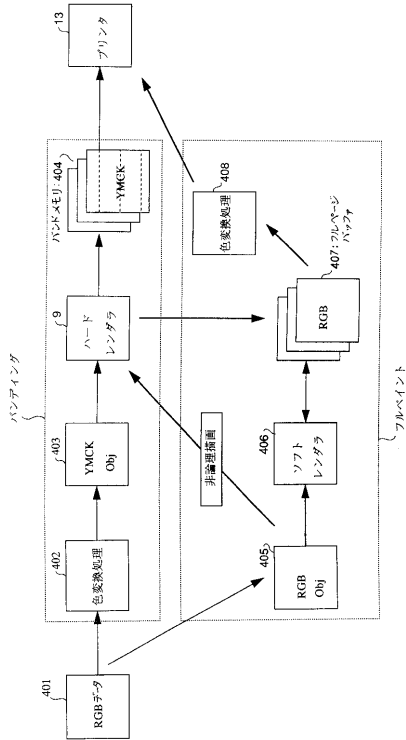
【 図 7 】



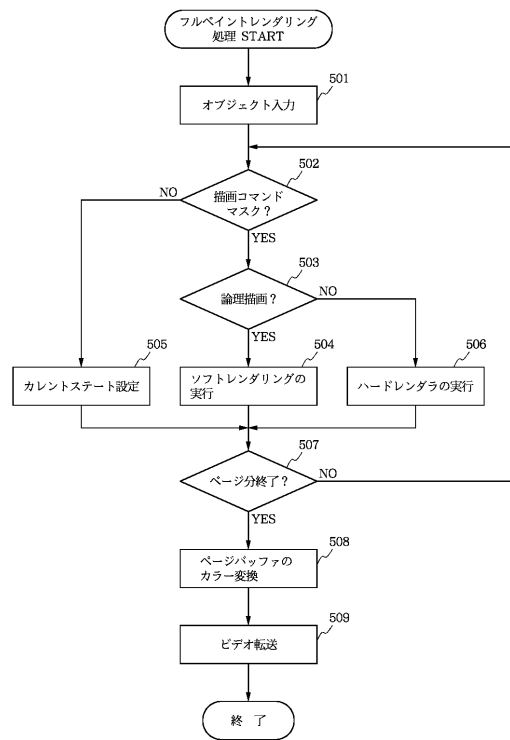
【 図 8 】



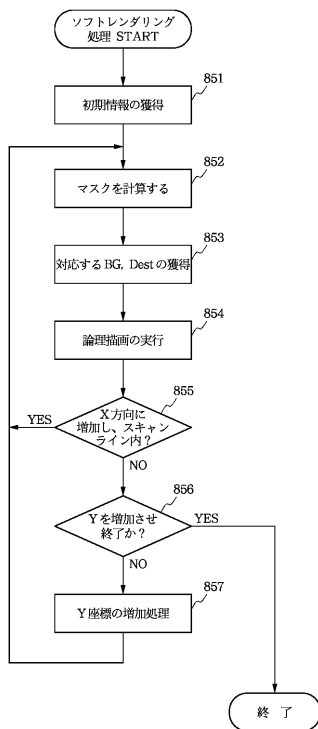
【 図 9 】



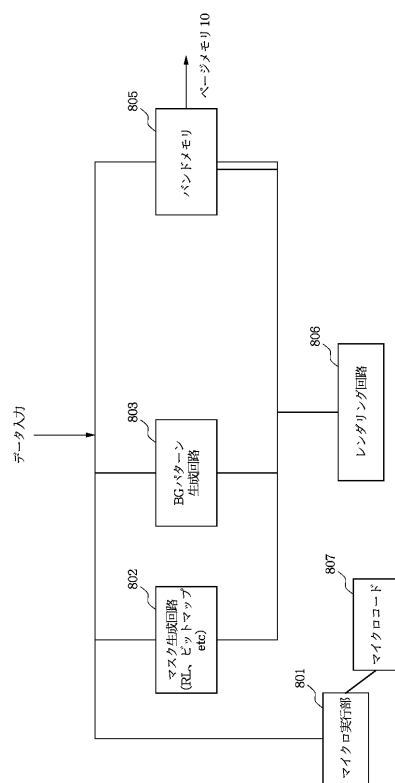
【 図 10 】



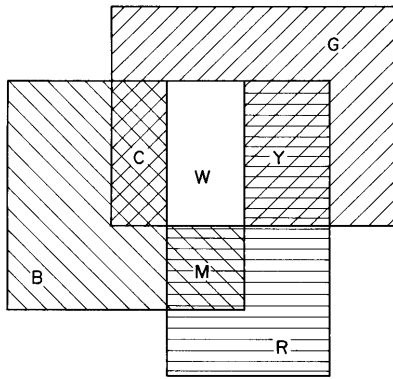
【 図 11 】



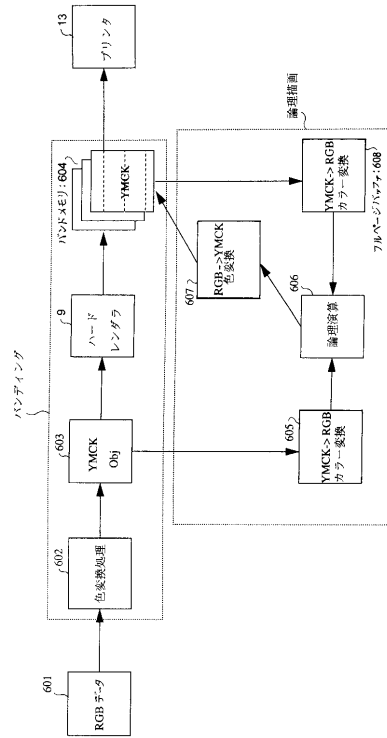
【 図 12 】



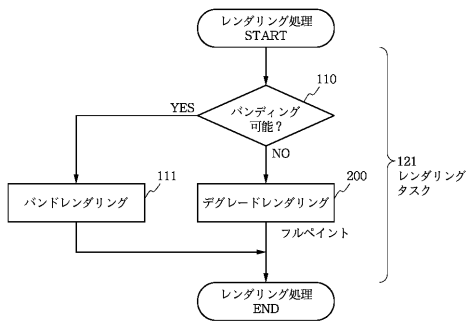
【 図 1 3 】



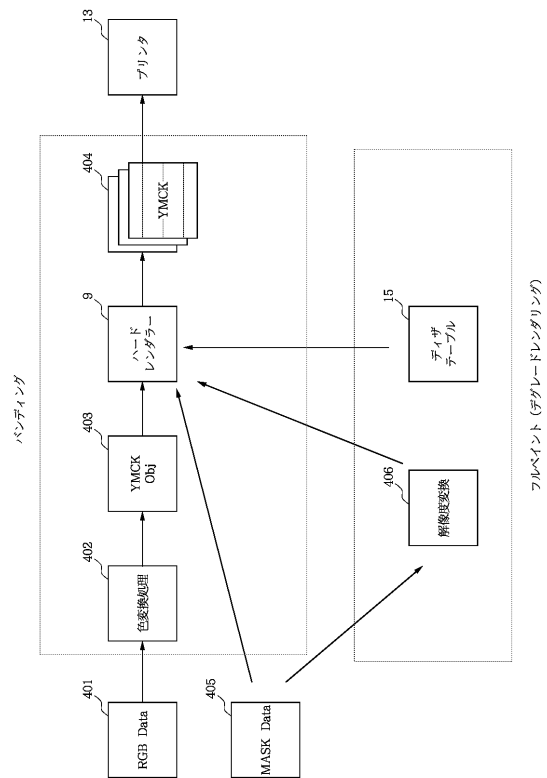
【 図 1 4 】



【 図 1 5 】

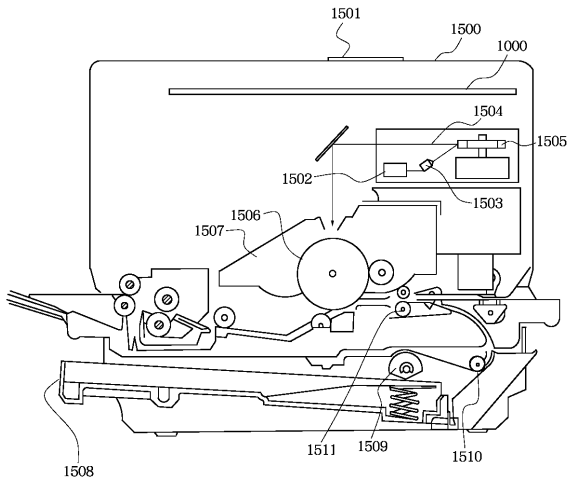


【 図 1 6 】





【 図 2 1 】



フロントページの続き

(58)調査した分野(Int.Cl.<sup>7</sup>, DB名)

H04N 1/46-1/64