

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第5317641号  
(P5317641)

(45) 発行日 平成25年10月16日(2013.10.16)

(24) 登録日 平成25年7月19日(2013.7.19)

(51) Int.Cl.

B 41 J 5/30 (2006.01)  
G 06 F 3/12 (2006.01)

F 1

B 41 J 5/30  
G 06 F 3/12Z  
Z

請求項の数 22 (全 30 頁)

(21) 出願番号 特願2008-293938 (P2008-293938)  
 (22) 出願日 平成20年11月17日 (2008.11.17)  
 (65) 公開番号 特開2010-120196 (P2010-120196A)  
 (43) 公開日 平成22年6月3日 (2010.6.3)  
 審査請求日 平成23年11月15日 (2011.11.15)

(73) 特許権者 000001007  
 キヤノン株式会社  
 東京都大田区下丸子3丁目30番2号  
 (74) 代理人 100076428  
 弁理士 大塚 康徳  
 (74) 代理人 100112508  
 弁理士 高柳 司郎  
 (74) 代理人 100115071  
 弁理士 大塚 康弘  
 (74) 代理人 100116894  
 弁理士 木村 秀二  
 (74) 代理人 100130409  
 弁理士 下山 治  
 (74) 代理人 100134175  
 弁理士 永川 行光

最終頁に続く

(54) 【発明の名称】 画像処理方法及び画像処理装置ならびに該画像処理方法を実行するためのプログラム

## (57) 【特許請求の範囲】

## 【請求項 1】

ページ記述言語 (PDL) で記述された PDL データからディスプレイリストを生成し、該ディスプレイリストをレンダリングしてビットマップデータを生成する画像処理方法であって、

決定手段が、PDL データからディスプレイリストを生成する PDL 処理の開始に先だって、前記 PDL 処理とディスプレイリストをレンダリングしてビットマップデータを生成する RIP 処理の負荷を均等とするために、前記 PDL 処理で重なり除去済みのディスプレイリストを生成するか、あるいは重なり除去を行っていないディスプレイリストを生成するかを決定する決定工程と、

前記決定工程で重なり除去を行っていないディスプレイリストの生成が決定された場合は、第 1 のビットマップデータ展開手段が、前記 PDL 処理で重なり除去を行っていないディスプレイリストを生成し、前記 RIP 処理で前記重なり除去を行っていないディスプレイリストを基にビットマップデータを生成する第 1 のビットマップデータ展開工程と、

前記決定工程で重なり除去済みのディスプレイリストの生成が決定された場合は、第 2 のビットマップデータ展開手段が、前記 PDL 処理で重なり除去済みのディスプレイリストを生成し、前記 RIP 処理で前記重なり除去済みのディスプレイリストを基にビットマップデータを生成する第 2 のビットマップデータ展開工程とを有することを特徴とする画像処理方法。

## 【請求項 2】

前記重なり除去は、オブジェクトの交差点間に含まれる塗りつぶし部分が持つレベル情報を基に、データを最上位から最下位に向かってソートして、上位レベルにより隠される不要なデータを除去するレベル処理と、ソートが終わった塗りつぶし部分が持つ重ね合わせ情報を基に複数の塗りつぶし部分を1つの塗りつぶし部分にするコンポジット処理とを含むことを特徴とする請求項1に記載の画像処理方法。

【請求項3】

前記決定工程では、前記PDL処理への負荷の予測と前記RIP処理への負荷の予測とに基づいて、前記RIP処理への負荷の予測が大きければ前記PDL処理で前記重なり除去済みのディスプレイリストの生成を決定し、前記PDL処理への負荷の予測が大きければ前記重なり除去を行っていないディスプレイリストの生成を決定することを特徴とする請求項1に記載の画像処理方法。

10

【請求項4】

前記決定工程では、重なり除去済みの部分と重なり除去を行っていない部分を前記PDL処理及びRIP処理への負荷に反比例する割合で含むディスプレイリストの生成を決定することを特徴とする請求項1に記載の画像処理方法。

【請求項5】

第1の取得手段が、前記PDL処理とRIP処理の順番待ちをしているデータの数と実行可能なCPUの個数を取得する第1の取得工程を更に有し、

前記PDL処理及びRIP処理への負荷は、各処理の順番待ちをしているデータの数と各処理を実行可能なCPUの個数に基づいて予測されることを特徴とする請求項3または4に記載の画像処理方法。

20

【請求項6】

第2の取得手段が、前記PDL処理とRIP処理の1頁平均の処理能力が取得可能な場合に、1頁平均の処理能力を取得する第2の取得工程を更に有し、

前記PDL処理及びRIP処理への負荷は、前記PDL処理とRIP処理の1頁当たりの平均処理能力と前記各処理の順番待ちしているデータの個数と前記各処理を処理可能なCPUの個数とに基づいて予測されることを特徴とする請求項5に記載の画像処理方法。

【請求項7】

前記PDL処理とRIP処理を行うデータがたまるキューのサイズが異なる場合に、第3の取得手段が、PDL処理の待ちのデータがためられるキューとRIP処理の待ちデータがためられるキューのサイズを取得する第3の取得工程を更に有し、

30

前記PDL処理及びRIP処理への負荷は、前記PDL処理のキューとRIP処理のキューのサイズと、前記各処理の順番待ちしているデータの個数と、前記各処理を処理可能なCPUの個数とに基づいて予測されることを特徴とする請求項5に記載の画像処理方法。

【請求項8】

前記決定工程では、前記RIP処理により生成されたビットマップデータをスプールするスプールメモリが使用可能であるか否かに基づいて、前記スプールメモリが使用可能で無ければ前記PDL処理で前記重なり除去済みのディスプレイリストの生成を決定し、前記スプールメモリが使用可能であれば前記重なり除去を行っていないディスプレイリストの生成を決定することを特徴とする請求項1に記載の画像処理方法。

40

【請求項9】

前記スプールメモリが他の処理に使われている場合に、前記スプールメモリが使用可能で無いと判断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10】

請求項1乃至9のいずれか1項に記載の画像処理方法の各工程をコンピュータに実行させるためのプログラム。

【請求項11】

請求項10に記載のプログラムを記憶したコンピュータで読み取り可能な記憶媒体。

【請求項12】

50

ページ記述言語（PDL）で記述されたPDLデータからディスプレイリストを生成し、該ディスプレイリストをレンダリングしてビットマップデータを生成する画像処理装置であって、

前記PDLデータからディスプレイリストを生成するPDL処理で重なり除去を行っていないディスプレイリストを生成し、前記RIP処理で前記重なり除去を行っていないディスプレイリストを基にビットマップデータを生成する第1のビットマップデータ展開手段と、

前記PDL処理で重なり除去済みのディスプレイリストを生成し、前記ディスプレイリストをレンダリングしてビットマップデータを生成するRIP処理で前記重なり除去済みのディスプレイリストを基にビットマップデータを生成する第2のビットマップデータ展開手段と、

前記PDLデータへの前記PDL処理の開始に先だって、前記PDL処理とRIP処理の負荷を均等とするために前記PDL処理で前記重なり除去済みのディスプレイリストと前記重なり除去を行っていないディスプレイリストの生成を決定する決定手段とを有することを特徴とする画像処理装置。

#### 【請求項13】

前記重なり除去は、オブジェクトの交差点間に含まれる塗りつぶし部分が持つレベル情報を基に、データを最上位から最下位に向かってソートして、上位レベルにより隠される不要なデータを除去するレベル処理と、ソートが終わった塗りつぶし部分が持つ重ね合わせ情報を基に複数の塗りつぶし部分を1つの塗りつぶし部分にするコンポジット処理とを含むことを特徴とする請求項12に記載の画像処理装置。

#### 【請求項14】

前記決定手段は、前記PDL処理への負荷の予測と前記RIP処理への負荷の予測とに基づいて、前記RIP処理への負荷の予測が大きければ前記PDL処理で前記重なり除去済みのディスプレイリストの生成を決定し、前記PDL処理への負荷の予測が大きければ前記重なり除去を行っていないディスプレイリストの生成を決定することを特徴とする請求項12に記載の画像処理装置。

#### 【請求項15】

前記決定手段は、重なり除去済みの部分と重なり除去を行っていない部分を前記PDL処理及びRIP処理への負荷に反比例する割合で含むディスプレイリストの生成を決定することを特徴とする請求項12に記載の画像処理装置。

#### 【請求項16】

前記PDL処理とRIP処理の順番待ちをしているデータの数と実行可能なCPUの個数を取得する第1の取得手段を有し、

前記PDL処理及びRIP処理への負荷は、各処理の順番待ちをしているデータの数と各処理を実行可能なCPUの個数に基づいて予測されることを特徴とする請求項14または15に記載の画像処理装置。

#### 【請求項17】

前記PDL処理とRIP処理の1頁平均の処理能力が取得可能な場合に、1頁平均の処理能力を取得する第2の取得手段を有し、

前記PDL処理及びRIP処理への負荷は、前記PDL処理とRIP処理の1頁当たりの平均処理能力と前記各処理の順番待ちしているデータの個数と前記各処理を処理可能なCPUの個数とに基づいて予測されることを特徴とする請求項16に記載の画像処理装置。

#### 【請求項18】

前記PDL処理とRIP処理を行うデータがたまるキューのサイズが異なる場合に、PDL処理の待ちのデータがためられるキューとRIP処理の待ちデータがためられるキューのサイズを取得する第3の取得手段を有し、

前記PDL処理及びRIP処理への負荷は、前記PDL処理のキューとRIP処理のキューのサイズと、前記各処理の順番待ちしているデータの個数と、前記各処理を処理可能

10

20

30

40

50

な C P U の個数とに基づいて予測されることを特徴とする請求項 1 6 に記載の画像処理装置。

【請求項 1 9】

前記 R I P 处理により生成されたビットマップデータをスプールするスプールメモリを有し、

前記決定手段は、前記スプールメモリが使用可能であるか否かに基づいて、前記スプールメモリが使用可能で無ければ前記 P D L 处理で前記重なり除去済みのディスプレイリストの生成を決定し、前記スプールメモリが使用可能であれば前記重なり除去を行っていないディスプレイリストの生成を決定することを特徴とする請求項 1 2 に記載の画像処理装置。

10

【請求項 2 0】

前記スプールメモリが他の処理に使われている場合に、前記スプールメモリが使用可能で無いと判断することを特徴とする請求項 1 9 に記載の画像処理装置。

【請求項 2 1】

請求項 1 2 乃至 2 0 のいずれか 1 項に記載の画像処理装置を有し、前記 R I P 处理により生成されたビットマップデータに従って画像を形成することを特徴とする画像形成装置。

【請求項 2 2】

ページ記述言語 ( P D L ) で記述された P D L データからディスプレイリストを生成する P D L 处理部と、

20

前記 P D L 处理部により生成されたディスプレイリストをレンダリングしてビットマップデータを生成する R I P 处理部と、

前記 P D L 处理部の能力と前記 R I P 处理部の能力と前記 R I P 处理部の処理待ち状態とに応じて、前記 P D L 处理部に、重なり除去済みのディスプレイリストを生成させるか、前記重なり除去が行われていないディスプレイリストを生成させるか、を決定し、重なり除去が行われていないディスプレイリストが前記 P D L 处理部によって生成された場合、前記 R I P 处理部に、該重なり除去が行われていないディスプレイリストを基にビットマップデータを生成させ、重なり除去済みのディスプレイリストが前記 P D L 处理部によって生成された場合、前記 R I P 处理に、該重なり除去済みのディスプレイリストを基にビットマップデータを生成させる制御手段と

30

を有することを特徴とする画像処理装置。

【発明の詳細な説明】

【技術分野】

【 0 0 0 1 】

本発明は、画像処理方法及び画像処理装置ならびに該画像処理方法を実行するためのプログラムに関する。特に、P D L 处理と R I P 处理の負荷分散の制御方法を有する画像処理方法及び画像処理装置ならびに該画像処理方法を実行するためのプログラムに関するものである。

【 0 0 0 2 】

ここで、P D L 处理とは、頁記述言語(Page Description Language)を解析してディスプレイリストを生成する処理である。また、R I P 处理とは、ラスタイメージプロセッサ(Raster Image Processor)でディスプレイリストをビットマップデータに展開する処理である。

40

【背景技術】

【 0 0 0 3 】

複数の C P U を使用する、これまでの印刷処理における P D L 处理と R I P 处理の並列処理やパイプライン処理においては、負荷の偏りとして、

- (1) R I P 处理時に描画領域により負荷の偏りが発生する、
  - (2) P D L 处理と R I P 处理に負荷の偏りが発生する、
- といったものがあげられる。

50

**【0004】**

それらの結果、並列処理を行ったとしても遅い方の処理に律速してしまい、最適な並列処理やパイプライン処理が行われない問題がある。

**【0005】**

このような問題の解決策として、下記のような提案がなされている。

**【0006】**

(1)に対応する解決策として、特許文献1では、入力されたPDLデータからDLを生成する際に、PDLデータから各描画領域の処理の負荷を予測して負荷補正值を生成する。そして、負荷が均等になるように複数の描画処理部へ描画領域の担当範囲を割り当てる事によって、問題の解決を図っていた。

10

**【0007】**

(2)に対応する解決案として、特許文献2では、各プロセッサの描画処理をビットマップまで展開した処理(ビットマップ展開)とするか、描画パラメータ(ベクター情報)の更新のみの処理(空描画)とするかを選択する。この処理によって、複数のプロセッサへの描画処理の負荷分散を行うという提案がなされている。

【特許文献1】特開平10-52950

【特許文献2】特開2005-63027

**【発明の開示】****【発明が解決しようとする課題】****【0008】**

20

しかしながら、上述した従来の技術では、以下のような問題が生じていた。

**【0009】**

まず、特許文献1では、DLを生成する際に同時に補正值を生成しているが、この補正值の生成自体がオーバーヘッドになり、印刷処理全体のパフォーマンスを落とす可能性がある。単純なデータの場合、そのオーバーヘッドもそれほど大きくないが、複雑なデータになればなるほど補正值の生成に時間がかかると予想される。また、PDL処理やRIP処理を行うCPUの個数が複数だった場合のCPUの割り当てには言及がされておらず、正確な処理時間の予測が出来ない問題がある。

**【0010】**

一方、特許文献2では、同じPDLデータを複数のプロセッサ(CPU)で並列にインタプリタ処理する為、無駄な処理(結果的に描画されないページの空描画:PDL処理も行われる)が発生します。また、特許文献2では、PDL処理を行う際にデータを保持するメモリ領域が各プロセッサ(CPU)毎に必要となるためメモリ使用量が多くなる。

30

**【0011】**

本発明は、上記問題点に鑑み、PDL処理とRIP処理の負荷のバランスを高速かつ容易にとることが出来る、効率的なパイプライン処理を実現する画像処理方法及び画像処理装置ならびに該画像処理方法を実行するプログラムを提供する。

**【0012】**

また、PDL処理ではビットマップ展開されていないDLを生成して、RIP処理に使用するメモリ量を少なく抑えることが出来る画像処理方法及び画像処理装置ならびに該画像処理方法を実行するためのプログラムを提供する。

40

**【課題を解決するための手段】****【0013】**

上記の問題を解決するために、本発明の画像処理方法は、ページ記述言語(PDL)で記述されたPDLデータからディスプレイリストを生成し、該ディスプレイリストをレンダリングしてビットマップデータを生成する画像処理方法であって、決定手段が、PDLデータからディスプレイリストを生成するPDL処理の開始に先だって、前記PDL処理とディスプレイリストをレンダリングしてビットマップデータを生成するRIP処理の負荷を均等とするために、前記PDL処理で重なり除去済みのディスプレイリストを生成するか、あるいは重なり除去を行っていないディスプレイリストを生成するかを決定する決

50

定工程と、前記決定工程で重なり除去を行っていないディスプレイリストの生成が決定された場合は、前記 PDL 处理で重なり除去を行っていないディスプレイリストを生成し、前記 RIP 处理で前記重なり除去を行っていないディスプレイリストを基にビットマップデータを生成する第 1 のビットマップデータ展開工程と、前記決定工程で重なり除去済みのディスプレイリストの生成が決定された場合は、前記 PDL 处理で重なり除去済みのディスプレイリストを生成し、前記 RIP 处理で前記重なり除去済みのディスプレイリストを基にビットマップデータを生成する第 2 のビットマップデータ展開工程とを有することを特徴とする。

#### 【 0014 】

また、上記画像処理方法の各工程をコンピュータに実行させるためのプログラム、及び該プログラムを記憶したコンピュータで読み取り可能な記憶媒体を提供する。 10

#### 【 0015 】

また、本発明の画像処理装置は、ページ記述言語 ( PDL ) で記述された PDL データからディスプレイリストを生成し、該ディスプレイリストをレンダリングしてビットマップデータを生成する画像処理装置であって、前記 PDL データからディスプレイリストを生成する PDL 处理で重なり除去を行っていないディスプレイリストを生成し、前記 RIP 处理で前記重なり除去を行っていないディスプレイリストを基にビットマップデータを生成する第 1 のビットマップデータ展開手段と、前記 PDL 处理で重なり除去済みのディスプレイリストを生成し、前記ディスプレイリストをレンダリングしてビットマップデータを生成する RIP 处理で前記重なり除去済みのディスプレイリストを基にビットマップデータを生成する第 2 のビットマップデータ展開手段と、前記 PDL データへの前記 PDL 处理の開始に先だって、前記 PDL 处理と RIP 处理の負荷を均等とするために前記 PDL 处理で前記重なり除去済みのディスプレイリストと前記重なり除去を行っていないディスプレイリストの生成を決定する決定手段とを有することを特徴とする。 20

#### 【 0016 】

また、本発明の画像処理装置は、ページ記述言語 ( PDL ) で記述された PDL データからディスプレイリストを生成する PDL 处理部と、前記 PDL 处理部により生成されたディスプレイリストをレンダリングしてビットマップデータを生成する RIP 处理部と、前記 PDL 处理部の能力と前記 RIP 处理部の能力と前記 RIP 处理部の処理待ち状態とに応じて、前記 PDL 处理部に、重なり除去済みのディスプレイリストを生成させるか、前記重なり除去が行われていないディスプレイリストを生成させるか、を決定し、重なり除去が行われていないディスプレイリストが前記 PDL 处理部によって生成された場合、前記 RIP 处理部に、該重なり除去が行われていないディスプレイリストを基にビットマップデータを生成させ、重なり除去済みのディスプレイリストが前記 PDL 处理部によって生成された場合、前記 RIP 处理に、該重なり除去済みのディスプレイリストを基にビットマップデータを生成させる制御手段とを有することを特徴とする。 30

#### 【 発明の効果 】

#### 【 0017 】

本発明においては、 PDL 处理において、次ぎの条件に従ってレベル処理を行わない ( レベル情報の残った ) DL を生成するか、レベル処理済みの ( レベル情報の除去された ) DL を生成するかを切り替える。切り替える条件は、 PDL 处理及び RIP 处理のキューにたまっている処理待ちデータの数と、 PDL 处理または RIP 处理を行うことが可能な CPU の個数とである。 40

#### 【 0018 】

この結果、 PDL 处理と RIP 处理の負荷のバランスを高速かつ容易にとることができ、効率的なパイプライン処理を実現することが出来る。

#### 【 0019 】

また、 PDL 处理を行う際にデータを保持する領域は 1 つですむので、メモリ量を少なく抑えることが出来る。

#### 【 発明を実施するための最良の形態 】

10

20

30

40

50

## 【0020】

以下、本発明を実施するための最良の実施形態について、添付図面を参照して詳細に説明する。

## 【0021】

## [実施形態1]

<本実施形態の画像処理装置の構成>

本実施形態を適用するに好適な1Dカラー系MFP (Multi Function Peripheral: マルチファンクション周辺機器) の構成例について、図1を用いて説明する。

## 【0022】

10

1Dカラー系MFPは、原稿露光部101、レーザ露光部102、感光ドラム103a、作像部103、定着部104、給紙/搬送部195、及び、これらを制御する不図示のプリンタ制御部から構成される。

## 【0023】

原稿露光部101は、原稿台に置かれた原稿に対して照明を当てて原稿画像を光学的に読み取り、その像を電気信号に変換して画像データを作成する。

## 【0024】

レーザ露光部102は、前記画像データに応じて変調されたレーザ光などの光線を等角速度で回転する回転多面鏡(ポリゴンミラー)に入射させ、反射走査光として感光ドラム103aに照射する。

20

## 【0025】

作像部103は、感光ドラム103aを回転駆動し、帯電器によって帯電させ、前記レーザ露光部102によって感光ドラム103a上に形成された潜像をトナーによって現像化し、そのトナー像をシートに転写する。その際に転写されずに感光ドラム103a上に残った微小トナーを回収するといった一連の電子写真プロセスを実行して作像する。

## 【0026】

その際、シートが転写ベルトの所定位置に巻きつき、4回転する間に、マゼンタ(M)、シアン(C)、イエロー(Y)、ブラック(K)のトナーを持つそれぞれの現像ユニット(現像ステーション)が入れ替わる。そして、順次前述の電子写真プロセスを繰り返し実行する。4回転の後、4色のフルカラートナー像を転写されたシートは、転写ドラムを離れ、定着部104へ搬送される。

30

## 【0027】

定着部104は、ローラやベルトの組み合わせによって構成され、ハロゲンヒータなどの熱源を内蔵し、前記作像部103によってトナー像が転写されたシート上のトナーを、熱と圧力によって溶解して定着させる。

## 【0028】

給紙/搬送部105は、シートカセットやペーパーデッキに代表されるシート収納庫を1つ以上持っている。給紙/搬送部105は、不図示のプリンタ制御部の指示に応じて、シート収納庫に収納された複数のシートの中から1枚分離し、作像部103から定着部104へ搬送する。シートは、作像部103の転写ドラムに巻きつけられ、4回転した後に定着部104へ搬送される。4回転する間に前述のYMC各色のトナー像がシートに転写される。また、シートの両面に画像形成する場合は、定着部104を通過したシートを再度作像部103へ搬送する搬送経路を通り制御する。

40

## 【0029】

不図示のプリンタ制御部は、MFP全体を制御するMFP制御部と通信して、その指示に応じて制御を実行する。同時に、前述の原稿露光、レーザ露光、作像、定着、給紙/搬送の各部の状態を管理しながら、全体が調和を保って円滑に動作できるよう指示を行う。

## 【0030】

<プリンタ制御部のハードウェア構成例>

図2は、本実施形態におけるMFPのプリンタ制御部(コントローラ)のハードウェア

50

構成例を示すブロック図である。

【0031】

図2において、プリンタ制御部200は、画像入力デバイスであるスキャナ201や画像出力デバイスであるプリンタエンジン202と接続し、画像データの読み取りやプリント出力のための制御を行う。また、プリンタ制御部200は、LAN\_N1や公衆回線204と接続することで、画像情報やデバイス情報をLAN\_N1経由で入出力するための制御を行う。

【0032】

CPU205は、MFP全体を制御するための中央処理装置である。RAM206は、CPU205が動作するためのシステムワークメモリであり、入力された画像データを一時記憶するための画像メモリでもある。さらに、ROM207はブートROMであり、システムのブートプログラムが格納されている。HDD208はハードディスクドライブであり、各種処理のためのシステムソフトウェア及び入力された画像データを等格納する。PDL処理部203は、入力されたページ記述言語（以下、PDL）データを解析し、CPU205の指示に従ってレベル処理をしないディスプレイリスト（以下、DL）とレベル処理済みのDLとを切り替えて、DLを生成する。CPU205の指示は、後述のように、RAM206やHDD208に記憶されたPDL処理とRIP処理のキューの数や、PDL処理とRIP処理に割り当てられたCPUの数により切り替えられる。

【0033】

操作部I/F209は、画像データ等を表示可能な表示画面を有する操作部210に対するインターフェース部であり、操作部210に対して操作画面データを出力する。また、操作部I/F209は、操作部210から操作者が入力した情報をCPU205に伝える役割をする。ネットワークインターフェース211は、例えばLANカード等で実現され、LAN\_N1に接続して外部装置との間で情報の入出力を行う。さらに、モデム212は公衆回線204に接続し、外部装置との間で情報の入出力を行う。

【0034】

以上のユニットがシステムバス213上に配置されている。イメージバスI/F214は、システムバス213と画像データを高速で転送する画像バス215とを接続するためのインターフェースであり、データ構造を変換するバスブリッジである。

【0035】

画像バス215上には、ラスタイメージプロセッサ（RIP処理部）216、デバイスI/F217、スキャナ画像処理部218、プリンタ画像処理部219が接続される。また、画像編集用画像処理部220、貯メモリ221、カラーマネージメントモジュール230が接続される。RIP処理部216は、DLのベクトルデータをイメージに展開して、貯メモリ221に一次記憶する。

【0036】

デバイスI/F部217は、スキャナ201やプリンタエンジン202とプリンタ制御部200とを接続し、画像データの同期系／非同期系の変換を行う。

【0037】

また、スキャナ画像処理部218は、スキャナ201から入力した画像データに対して、補正、加工、編集等の各種処理を行う。プリンタ画像処理部219は、プリント出力する画像データに対して、プリンタエンジンに応じた補正、解像度変換等の処理を行う。画像編集用画像処理220は、画像データの回転や、画像データの圧縮伸長処理等の各種画像処理を行う。CMM230は、画像データに対して、プロファイルやキャリブレーションデータに基づいた、色変換処理（色空間変換処理ともいう）を施す専用ハードウェアモジュールである。プロファイルとは、機器に依存した色空間で表現したカラー画像データを機器に依存しない色空間（例えばL a bなど）に変換するための関数のような情報である。キャリブレーションデータとは、カラー複合機3におけるスキャナ201やプリンタエンジン202の色再現特性を修正するためのデータである。

【0038】

10

20

30

40

50

## &lt;プリンタ制御部のソフトウェア構成例&gt;

図3は、MFPの動作を制御するプリンタ制御部のソフトウェアの構成例を示すプロック図である。図3で、貯メモリ(RAM)206やドキュメント記憶部(HDD)208を除く各部は、CPU205あるいはPDL処理やRIP処理を行う複数のCPUによるソフトウェア処理により実現される。図3で図2と同様の機能を果たす構成要素は、同じ参照番号で示している。

## 【0039】

ネットワークI/F211は、外部との入出力のためのインターフェースである。プロトコル制御部301は、ネットワークプロトコルを解析して送信することによって外部との通信を行う制御部である。

10

## 【0040】

PDL処理部203は、PDLデータを解析し、より処理しやすい形式のDLに変換する処理部である。PDL処理部203において生成されたDLは、データ描画部(RIP処理部)216に渡されて処理される。RIP処理部303は、PDL処理部203で生成されたDLをビットマップデータに展開するものであり、展開されたビットマップデータは貯メモリ221に逐次描画されて行く。貯メモリ221はレンダラが展開するビットマップデータを一次的に保持する揮発性のメモリである。

## 【0041】

パネル入出力制御部305は、操作パネルからの入出力を制御するものである。ドキュメント記憶部306はデータファイルを記憶するハードディスク等の二次記憶装置である。スキャン制御部307は、スキャナ201から入力した画像データに対して、補正、加工、編集などの各種処理を行う。印刷制御部308は、貯メモリ221の内容をビデオ信号に変換処理し、プリンタエンジン202へ画像転送を行なう。プリンタエンジン202は、受け取ったビデオ信号を記録紙に永久可視画像形成するための印刷機構部である。

20

## 【0042】

## &lt;本実施形態の画像処理の機能構成例&gt;

図4は、本実施形態のプリンタ制御部での画像処理の機能構成例を示す図である。図4で図2や図3と同じ構成要素は、同じ参照番号で示している。以下、画像データの流れに従って説明する。

## 【0043】

30

図4の401は、外部からのPDLデータを受信するPDLデータ受信部である。PDLデータ受信部401で受信されたPDLデータは、順にPDL処理待ちキュー404にキューイングされる。PDL処理待ちキュー404にキューイングされたPDLデータは、ファーストイン・ファーストアウト(FIFO)でPDL処理部203に読み出されて、PDL処理部203の複数のCPUで並列処理される。この時に、PDL処理切替部403において、PDL処理部203で重なり除去を行うか否かが決定されて、重なり除去処理フラグとしてPDL処理部203に通知される。

## 【0044】

ここで、重なり除去処理フラグは、PDL処理待ちキュー404の待ちキュー数、後述のPDL処理部405で動作可能なCPU数、RIP処理待ちキュー406の待ちキュー数、RIP処理部で動作可能なCPU数、に従って決定される。なお、本実施形態では、キュー数とCPU数として単純化しているが、キュー数は負荷量を示す一例であり、CPU数は処理量を示す一例に過ぎない。さらに多用な条件を考慮すれば、より正確な負荷量と処理量とが算出可能であり、キュー数とCPU数に限定されない。本発明は、かかる負荷量と処理量とに基づいてPDL処理で重なり除去を行うか否かを切り替えることが、特徴ある構成である。

40

## 【0045】

PDL処理部203で重なり除去処理フラグに従って処理された、重なり除去済みのDLあるいは重なり除去を行わないDLは、RIP処理待ちキュー406に重なり除去処理フラグと共にキューイングされる。なお、重なり除去処理フラグでなくDL内に重なり除

50

去か否かを記憶させ、RIP処理部407が処理前にDLを解析して判断する構成であつてもよい。

【0046】

RIP処理待ちキュー406のDLは、ファーストイン・ファーストアウト(FIFO)でRIP処理部216に読み出されて、重なり除去か否かに従ってRIP処理部216の複数のCPUで並列にビットマップデータに展開処理される。RIP処理部216で展開処理されたビットマップデータは、貯メモリ221に一時記憶され、1頁の画像形成のためプリンタエンジン202に出力される。

【0047】

(本実施形態の画像処理に必要なデータ及びプログラム例)

10

図5は、本実施形態のプリンタ制御部が画像処理に必要とするデータ及びプログラム例を示す図である。なお、図5には、本実施形態に特徴的なデータ及びプログラムのみを示し、他の汎用的なデータ及びプログラムは省かれている。また、図5で図2と同じ構成要素は、同じ参照番号で示している。

【0048】

図5で、RAM206には以下のデータが一時記憶される。206aは、画像メモリの記憶領域であり、受信されたPDLデータが一時記憶される。206bは、PDL処理において重なり除去を行うか否かを決める重なり除去処理フラグの記憶領域である。かかる記憶領域206aに記憶されているPDLデータが後述のPDL処理待ちキュー208aにキューイングされる。206cは、PDL処理部で使用可能なCPU数とRIP処理で使用可能なCPU数を記憶するCPUの数の記憶領域である。206dは、PDL処理部の処理を待っているキュー数とRIP処理を待っているキュー数を記憶する各処理の待ちキューの数の記憶領域である。

20

【0049】

図5で、HDD208には以下のデータ及びプログラムが格納される。208aは、PDL処理部203でDLを生成するためのPDLデータをFIFO方式で格納するPDL処理待ちキューである(図4の404参照)。208bは、RIP処理部216でDLからビットマップデータを展開するための重なり除去処理フラグ付きDLをFIFO方式で格納するRIP処理待ちキューである(図4の406参照)。また、208cは、以下の図13Aに示すフローチャートに従うPDL処理切替プログラムを格納する領域である。かかるPDL処理切替プログラムはRAM206にロードされてCPU205により実行される。

30

【0050】

図5で、貯メモリ221にはRIP処理部216で展開されたビットマップデータ221aが一時記憶される。

【0051】

<本実施形態のプリンタ制御部のデータ処理例>

次に、重なり除去を行っていないDLと重なり除去済みDLの生成、及びDLからのビットマップデータの生成における、PDL処理部とRIP処理部の負荷について、図6A及び図6Bに基づいて説明する。本実施形態では、図6A及び図6Bのように、PDL処理部とRIP処理部の負荷が異なる2つのビットマップデータ展開を準備して、この2つのビットマップデータ展開を負荷が均等隣るように割り当てる。なお、図2乃至図5と同様の構成要素には、同じ参照番号を付加している。

40

【0052】

なお、PDLデータによるプリントとは、PC(Personal Computer)上のアプリケーションソフトから印刷を指示した場合に、PC上のプリンタドライバによって生成されたPDLデータを受け取って出力する、プリンタ動作のことである。

【0053】

(重なり除去を行っていないDLの生成：第1のビットマップデータ展開)

図6Aは、重なり除去を行っていないDLを生成する場合のプリンタ制御部でのデータ

50

フローを示している。

【0054】

受信したPDLデータは、PDL処理待ちキュー404にキューイングされた後、まず、PDL処理部203でPDLデータ解析処理D701によって解析される。次に、貞の先頭からスキャンラインごとにエッジの交差点を抽出するエッジ処理D702を行う。エッジ処理D702が終了した時点でPDL処理部202から出力されるデータは、重なり除去を行っていないDLである。重なり除去を行っていないDLは、RIP処理待ちキュー406にキューイングされる。

【0055】

次に、RIP処理部216において、重なり除去を行っていないDLを入力データとして、レベル処理D703を行う。レベル処理D703では、エッジ処理D702で抽出した交差点間(スパンと言う)に含まれる塗りつぶし部分(ファイルと言う)が持つレベル情報を基に、データを最上位から最下位に向かってデータをソートする。そして、上位レベルにより隠される不要なデータを除去する。次に、コンポジット処理D704を行う。コンポジット処理とは、ソート処理が終わったファイルが持つ重ね合わせ情報(ラスタオペレーションや透過処理)を基に複数のファイルを最終的に1つのファイルにする処理のことである。最後に、ピクセル生成処理D705においてエッジ処理D702で算出したスパン分のピクセルを生成する処理を行い、ビットマップデータが展開される。ビットマップデータは貞メモリ221に一時記憶されプリンタエンジン202に転送され、最終的に紙上に画像が出力される。

10

20

【0056】

(重なり除去済みのDLの生成：第2のビットマップデータ展開)

次に、図6Bは、重なり除去済みのDLを生成する場合のプリンタ制御部でのデータフローを示している。

【0057】

重なり除去されていないDL生成のデータフローと異なる点は、エッジ処理D707に引き続き、レベル処理D708及びコンポジット処理D709もPDL処理部203で行うことにより、重なり除去済みのDLが生成される点である。その結果、PDL処理部203からRIP処理待ちキュー406に出力されるDLが重なり除去済みDLであるので、RIP処理部216ではそのままピクセル生成処理D710を行うだけで、ビットマップデータが生成できる。

30

【0058】

図6A及び図6Bを比較すると、その負荷の大小が明らかである。すなわち、図6AのようにPDL処理で重なり除去を行っていないDLを生成する場合は、レベル処理とコンポジット処理の無いPDL処理の負荷が小さく、レベル処理とコンポジット処理が追加されたRIP処理の負荷が大きくなる。一方、図6BのようにPDL処理で重なる除去済みのDLを生成する場合は、レベル処理とコンポジット処理が必要なPDL処理の負荷が大きく、レベル処理とコンポジット処理の無いピクセル生成処理だけのRIP処理の負荷が小さくなる。

40

【0059】

<DLのデータ構造例>

次に、図7A及び図7Bを用いて、重なり除去を行っていないDLと重なり除去済みDLの構造について、説明を行う。図7A及び図7Bでは、左側の絵が最終的に出力されるイメージであり、真ん中と右側がDLにどのような状態でデータが含まれているかを示している。

【0060】

(重なり除去を行っていないDLの構成例)

図7Aは、重なり除去を行っていないDLのデータ構成例である。

【0061】

重なり除去を行っていないDLでは、犬の絵の上下の透過グラデーション部分について

50

、ビットマップと透過グラデーションの合成されたビットマップになってない。その代わり、透過重ね合わせ情報 801 からグラデーション 802 とビットマップ 803 とがリンクされている構造になっている。また、グラデーションも、実際のビットマップではなく、スタートの色、エンドの色、幅など、グラデーションの定義という形で生成された DL 中に含まれている。

#### 【0062】

(重なり除去済みの DL の構成例)

一方、図 7B は、重なり除去済み DL のデータ構成例である。

#### 【0063】

重なり除去済み DL では、ビットマップとグラデーションとが透過で重ね合わされた後のビットマップとして生成された DL に含まれているので、レベルが 2 つ以上存在することはない。また、グラデーションもビットマップの状態になったものが生成された DL に含まれている。

10

#### 【0064】

ここでいうレベルとは、DL に含まれるオブジェクトが持つ上下関係のことで、各オブジェクトは一意のレベル番号 (No.) を持つ。処理中のスキャンラインに複数のオブジェクトが重なったときは、レベル No. が大きいオブジェクトの方が上にくる。レベル処理では、処理中のスキャンラインのオブジェクトが 1 つの場合、重ね合わせ処理を行う必要がないので、そのオブジェクトのフィルがそのままレンダリングされる。しかし、2 つ以上ある場合、そのオブジェクトの上下関係をレベル No. を基に算出し、さらに重ね合わせ情報を基に重ね合わせ処理を行う必要がある。最終的に出力されるオブジェクトは、重ね合わせ処理後のフィルが出力される。

20

#### 【0065】

<重ね合わせ情報の重なり除去とレンダリングの例>

ここで、重なり情報 801、802、803 がどのようにレンダリングされるかを説明する。

#### 【0066】

図 8 の 161、162 はそれぞれ DL に含まれる情報で、161 は、ビットマップと重ね合わせられるグラデーションの透過情報を示す。162 は、透過情報と重ね合わせられるビットマップ情報を示す。

30

#### 【0067】

透過情報 161 を基に、DL に含まれる幅高の大きさで、フィル情報 (Fill In Info) に含まれる色から色 (この場合は透過値) へ変わるように 1 ピクセルずつ透過を表わすビットマップ 163 を生成していく。図 8 では、黒が透過 “0” %、白が透過 “100” % である。また、ビットマップ情報 162 を基にオブジェクト画像のビットマップ 164 が生成される。透過を表わすビットマップ 163 とオブジェクト画像のビットマップ 164 とは、1 ピクセルずつ透過情報 161 に含まれる重ね合わせ情報を基に重ね合わせの計算が行われる。その結果、重なり除去済みのビットマップ 165 が生成される。

#### 【0068】

本説明では、ビットマップとグラデーションとの重ね合わせだったため、それぞれビットマップ展開していた。しかし、一面塗りつぶし同士の重ね合わせの場合は、ビットマップ展開する必要はなく、1 ピクセルのみの重ね合わせの計算を行えばよい。

40

#### 【0069】

< PDL 处理 / RIP 处理と他処理における CPU の関係 >

図 9において、PDL 处理 / RIP 处理において使用する CPU 数の例について説明する。

#### 【0070】

前述したように、プリント処理は、PDL データから DL を生成する PDL 处理 203 と DL を展開してビットマップを生成する RIP 处理 216 とに分けられる。各処理を行うために割り当てられた CPU の個数はシステム構成によって決められるが、図 9 の上段

50

の例 1においては、PDL処理に2個のCPU、RIP処理に2個のCPUが割り当てられているとする。

【0071】

ところで、MFPでは、プリント処理以外にも、コピー、FAX、スキャンなどの処理が並列に動作することが考えられる。そのため、プリント処理とその他の処理がCPUを共有する場合、図9の下段の例2のように、RIP処理216に2個割り当てられていたCPUのうち、1つのCPUがFAX処理1001に利用されることが考えられる。その結果、図9の上段の例1に比べて、下段の例2ではRIP処理216の能力が半分になつたと考えることができる。

【0072】

なお、本例では、利用可能な（処理を実行可能な）CPUの個数の変化は、ある頁のRIP処理が終了してから他の頁のPDL処理が開始される前に発生し、同じ頁のPDL処理とRIP処理との間に使えるCPUの個数が変化することは考えていません。また、本例では、煩雑をさけるため各CPUの能力は同等と仮定しているので、PDL処理とRIP処理でのCPUの処理能力は単純にCPUの個数に影響される。

【0073】

しかしながら、各処理で使用可能なCPUの個数や、他の処理が動作した場合にどのCPUを利用するかは、上記例に限定されません。かかるCPUの割り当て数は、CPU205によりRAM206のCPUの数206cを使って管理される。また、CPUの処理能力が同等でない場合には、個数のみで処理能力を判断するのではなく、CPUの能力も含めて判断する。更に、PDL処理及びRIP処理の能力はメモリ容量などにも影響されるので、メモリ空容量などの他のパラメータも考慮に入れてPDL処理における重なり除去の有無を判断してもよい。

【0074】

＜DLのつくり分けの判断の具体例＞

図10及び図11を使って、PDL処理における重なり除去の有無の2種類のDLの生成の判断についての具体的な説明を行う。

【0075】

かかる判断は、PDL処理待ちキュー404とRIP処理待ちキュー406にたまっているデータを、現在のPDL処理とRIP処理で使用可能なCPUで処理した場合、どちらが全てのデータを早く処理し終われるかという基準で行う。PDL処理が早く終わると予測された場合、PDL処理の負荷が大きい様に、PDL処理待ちキュー404から読み出してPDL処理するPDLデータからは、重なり除去済みDLを生成する様に指示する。一方、RIP処理の方が早く終わると予測された場合、RIP処理の負荷が大きくなる様に、PDL処理待ちキュー404から読み出してPDL処理するPDLデータからは、重なり除去を行っていないDLを生成する様に指示する。

【0076】

（重なり除去を行うと判断する状況例）

図10は、PDL処理もRIP処理もCPUが1個で、4頁の印刷中、1～3頁はイメージデータ、4頁目はグラフィックデータである場合の例を示す。図10中、“S”は重なり除去の処理データを示し、“M”は重なり除去を行わない処理データを示す。

【0077】

本例では、同じCPUの個数なので、単純にキューにたまっているデータが多い方の処理負荷が大きいと判断する。その結果、先頭頁のPDL処理開始時点では、PDL処理の負荷が大きいので、図10の上段の様に、PDL処理の負荷が軽い重なり除去を行っていないDLの生成を指示する。1～3頁は、そのまま重なり除去を行っていないDLの生成が指示される。

【0078】

4頁目のPDL処理開始時点では、図10の下段の様に、先頭頁がRIP処理部216で処理されているが、重なり除去を行っていないDL（1M）なのでRIP処理部21

10

20

30

40

50

6の負荷は大きい。そのため、RIP処理待ちキュー406には既にPDL処理部203で生成された重なり除去を行っていないDL(2M, 3M)がスプールされている。

#### 【0079】

ここで、4項目のPDL処理における重なり除去の有無を判断する。PDL処理部203の負荷は、4項目がグラフィックデータであるので、レベル処理を行うとかなり大きいと予測される。しかしながら、RIP処理部216は現在先頭頁のビットマップデータへの展開を行っており、更にRIP処理待ちキュー406には2つの重なり除去を行っていないDL(2M, 3M)がキューイングされている。PDL処理部203とRIP処理部216の処理能力は、CPUの数から考慮すると同等であるので、RIP処理部216の負荷がより大きいと判断する。

10

#### 【0080】

その結果、4項目のPDL処理においては重なり除去を行うように指示する。なお、かかる判断例は、上述のように、CPUの能力や他の条件(例えば、メモリ容量など)によって変わり得る。また、重なり除去(レベル処理)をPDL処理中に行う場合と、RIP処理中に行う場合の処理速度やメモリ使用量などの違いがある場合などは、それらも判断の条件となり得る。

#### 【0081】

(重なり除去を行わないと判断する状況例)

図11は、PDL処理がCPU1個、RIP処理がCPU2個で、属性に変化のない6頁の印刷を行う例を示す。図11中、“S”は重なり除去の処理データを示し、“M”は重なり除去を行わない処理データを示す。

20

#### 【0082】

まず、先頭頁のPDL処理開始時点では、PDL処理の負荷が大きいので、図11の上段のように、PDL処理の負荷が軽い重なり除去を行っていないDLの生成を指示する。1~4頁は、そのまま重なり除去を行っていないDLの生成が指示される。

#### 【0083】

図11の5項目のPDL処理開始時点では、図11の下段のように、PDL処理待ちキュー403には5, 6頁の2頁がキューイングされている。一方、RIP処理待ちキュー406にたまっている頁数は、重なり除去を行っていないDL(3M, 4M)である。従って、頁数だけでは両処理の負荷は同じであるが、RIP処理に使用できるCPUの個数がPDL処理の2倍なので処理能力も2倍とみなす。従って、PDL処理は1CPU当たり2頁、RIP処理は1CPU当たり1頁となり、PDL処理の方が負荷が大きい状態であると判断する。その結果、PDL処理部203には負荷の軽い重なり除去を行っていないDLの生成が指示される。

30

#### 【0084】

なお、本例においても、CPUの能力の違いや、重なり除去を行っていないDLと重なり除去済みのDLの展開の負荷の違いなどは考慮せずに、条件をキューイング数とCPU数に単純化している。

#### 【0085】

<実施形態1の処理手順例>

40

図12A乃至図12Cのフローチャートを参照して、実施形態1のPDL処理とRIP処理の負荷分散について、その処理手順を説明する。

#### 【0086】

(CPU205によるPDL処理での重なり除去の有無の指示)

図12Aは、CPU205が実行するPDL処理の重なり除去の有無の指示の手順例を示すフローチャートである。

#### 【0087】

CPU205は、ステップS1001においてPDLデータを待って、受信したPDLデータ206aを受け付け、PDL処理待ちキュー404にキューイングする。次に、CPU205は、ステップS1002において、PDL処理部203でDL生成を開始する

50

時点の、PDL処理待ちキュー404とRIP処理待ちキュー406にたまっているデータの頁数を取得する。次に、CPU205は、ステップS1003において、DL生成を開始する時点の、PDL処理とRIP処理それぞれに使用可能なCPUの個数を取得する。次に、CPU205は、ステップS1004において、ステップS1002とS1003で取得したデータを基に、その時点でPDL処理とRIP処理から負荷の大きい処理を判別する。なお、上記データの頁数の取得及びCPUの個数の取得を第1の取得という。

#### 【0088】

次に、CPU205は、ステップS1004の判別を基に、ステップS1005において生成するDLを重なり除去を行っていないDLとするか、重なり除去済みDLとするかを決定する。次に、CPU205は、ステップS1006において、PDL処理部203に今からPDL処理するPDLデータから生成されるDLのDLタイプ（（図4では、重なり除去処理フラグ：重なり除去を行っていないDLか、重なり除去済みDLか）を指示する。

#### 【0089】

なお、図12Aでは、煩雑さを避けるため、PDL処理部203でDL生成を開始する時点のタイミングが明瞭に図示されていない。実際には、PDL処理待ちキュー404へのキューイングと、PDL処理部203でのDL生成の開始とは非同期で行われる。かかる非同期の処理は、図12Aから容易に変形が可能である。

#### 【0090】

（PDL処理部のCPUによる処理）

図12Bは、PDL処理部のCPUが実行する処理の手順例を示すフローチャートである。

#### 【0091】

まず、PDL処理部のCPUはステップS1007で、PDL処理待ちキュー404から先頭のPDLデータを読み出すと共にCPU205からのDLタイプを受信する。次に、PDL処理部のCPUはステップS1008で、DLタイプに基づいて、重なり除去を行っていないDLの生成か、重なり除去済みDLの生成かを判断する。

#### 【0092】

重なり除去を行っていないDLの生成の場合（S1008のYes）、PDL処理部のCPUはステップS1009で重なり除去を行っていないDLを生成する。一方、重なり除去済みDLの生成の場合（S1008のNo）、PDL処理部のCPUはステップS1010で重なり除去済みDLを生成する。PDL処理部のCPUはステップS1011で、生成されたDLをRIP処理待ちキュー406にキューイングする。

#### 【0093】

（RIP処理部のCPUによる処理）

図12Cは、RIP処理部のCPUが実行する処理の手順例を示すフローチャートである。

#### 【0094】

まず、RIP処理部のCPUはステップS1012で、RIP処理待ちキュー406から先頭のDLをDLタイプと共に読み出す。次に、RIP処理部のCPUはステップS1013で、読み出したDLに基づいてDLタイプに従ってDLのレンダリングを行う。RIP処理部のCPUはステップS1014で、ステップS1013で展開されたビットマップデータを貯メモリ221に出力する。

#### 【0095】

<実施形態1の作用効果>

（メモリ使用量の低減）

特許文献2等の従来例では、PDL処理を行う際にデータを保持するメモリ領域が各プロセッサ（CPU）毎に必要となるが、本実施形態によれば、PDL処理を行う際にデータを保持する領域は1つですむ。すなわち、本実施形態のように、PDL処理とRIP処

10

20

30

40

50

理のいずれかで重なり除去を行う切り替え（レベルでの負荷分散）を行えば、メモリ使用量が少なくなる。

【0096】

（CPUへの負荷の低減）

特許文献2等の従来例では、同じPDLデータを複数のプロセッサ（CPU）で並列にインタプリタ処理する為、無駄な処理（結果的に描画されないページの空描画：PDL処理も行われる）が発生する。しかし、本実施形態では、PDL処理はあくまでも1回で、CPUの個数に応じて変わるのは計算速度などであり、CPUへの負荷が低減される。

【0097】

（負荷分散による処理速度の改善）

10

図13Aは、特許文献2のビットマップ展開と空描画との切り替え（上段）と、PDL処理とRIP処理の固定化による処理（下段の2つ）とのビットマップデータ展開の処理速度を示す図である。図13Bは、本実施形態の各処理のCPUが1個の場合の重なり除去処理の分散（上段）と、各処理のCPUが2個の場合の重なり除去処理の分散（下段）とのビットマップデータ展開の処理速度を示す図である。

【0098】

なお、図13A及び図13Bでは、比較のため、同じ属性の4頁のデータを処理する例を示し、1頁の処理を5コマ（内訳は、PDL処理2コマ、レベル処理など1コマ、RIP処理2コマ）と仮定した場合の例とする。

【0099】

20

図13Aの上段は、従来技術である特許文献1に従った処理である。CPU（プロセッサ）を3個（P1～P3）有し、各プロセッサはビットマップ展開かベクター情報を生成するのみの空描画（PDL処理のみと仮定した）かを選択して、並列処理を行う例である。

【0100】

各プロセッサは、5コマのビットマップ展開と2コマのPDL処理のみとを選択し、最初に対象頁の処理を開始したプロセッサのみが5コマのビットマップ展開し、他のプロセッサは2コマのPDL処理のみで次ぎの頁の処理に移る。かかる処理手順で4頁をビットマップ展開する時間は、14コマの時間に相当する。

【0101】

30

図13Aの下段の2つは、上図がPDL処理で重なり除去済みのDLを生成するよう固定した場合の例であり、下図がPDL処理で重なり除去を行わないDLを生成するよう固定した場合の例である。かかる例では、本実施形態のように、処理の途中で重なり除去処理（レベル処理、コンポジット処理）がPDL処理とRIP処理との分散されることはない。

【0102】

いずれの場合も、5コマの内の2コマがPDL処理部とRIP処理部とで並列処理されるのみで、ビットマップ展開するには共に14コマの時間を要する。

【0103】

これに対して、図13Bには、本実施形態による重なり除去をPDL処理部とRIP処理部のいずれで行うかを制御する方法による4頁のビットマップ展開に要する時間を示す。

40

【0104】

図13Bの上段は、PDL処理部とRIP処理部のいずれもがCPUが1個の場合の例、図13Bの下段は、PDL処理部とRIP処理部のいずれもがCPUが2個の場合の例である。

【0105】

図13Bの上段では、1頁から3頁ではPDL処理部で重なり除去を行わないDLを生成するが、4頁目にはRIP処理部の負荷が大きくなるためPDL処理部で重なり除去済みのDLを生成する。その結果、9コマでPDL処理部とRIP処理部の両方で重なり除

50

去が並列に行われ、13コマの時間で4頁のビットマップ展開が終了する。

【0106】

図13Bの下段では、1頁と2頁ではPDL処理部で重なり除去を行わないDLを生成するが、3,4頁目にはRIP処理部の負荷が大きくなるためPDL処理部で重なり除去済みのDLを生成する。その結果、1,2頁のRIP処理部における重なり除去と、3,4頁のPDL処理部における重なり除去が並列処理されて、7コマの時間で4頁のビットマップ展開が終了する。

【0107】

なお、図13A及び図13Bの例では、4頁のみの処理でPDL処理とRIP処理とを同じ負荷とする等の簡略化をしているので、僅かな改善しか見られない。しかし、処理する頁数が増加したりPDL処理とRIP処理の負荷の相違などを考慮すると、大幅なビットマップ展開の処理時間の改善が明らかである。

10

【0108】

<1ページ平均の処理能力を考慮する例>

なお、上記実施形態1では、待ちキュー数とCPUの個数に基づいて、PDL処理で重なり除去を行うかを判断した。

【0109】

しかし、更に、PDL処理とRIP処理の1頁平均の処理能力が取得可能な場合には、次のような処理も可能である。すなわち、まず、1頁平均の処理能力を取得する。そして、PDLとRIPの1頁当たりの平均処理能力と、PDL処理とRIP処理の順番待ちしているデータの個数と、処理可能なCPUの個数とから、PDL処理で重なり除去を行うか否かを判断してもよい。なお、かかる1頁平均の処理能力の取得を第2の取得という。

20

【0110】

例えば、処理能力の異なるCPU(例：500MHzのARMと3GHzのIntel CPU)が混在している構成の印刷処理装置の場合など、どのCPUに何の処理が割り当てられたかによって処理時間は大きく異なる。その限定方法としては、本処理前にCPU能力を取得して情報を保存しておき、PDL処理やRIP処理を行うCPUが決定した際、保存した情報から処理能力を取得する。

【0111】

<キューのサイズを考慮する例>

30

また、PDL処理とRIP処理を行うデータがたまるキューのサイズが異なる場合には、次のような処理も可能である。すなわち、まず、PDL処理待ちのデータがためられるキューとRIP処理待ちデータがためられるキューのサイズを取得する。そして、PDL処理のキューサイズ及びRIP処理のキューサイズと、PDL処理とRIP処理の順番待ちしているデータの個数と、処理可能なCPUの個数とから、PDL処理で重なり除去を行うか否かを判断してもよい。なお、かかるキューのサイズの取得を第3の取得という。

【0112】

上記CPUの能力と同様、PDL処理とRIP処理それぞれのキューサイズが、例えば256Mbyteか10Mbyteかでは、キューがたまる速度が大きく変わる(キューがフルになるとその前の処理は止まってしまう)。その限定方法としては、本処理前にそれぞれのキューサイズを取得して情報を保存しておき、PDL処理やRIP処理を行う際に、保存した情報から、PDL処理とRIP処理のそれぞれの処理能力を計算する為の情報とする。

40

【0113】

(CPUの処理能力及びキューサイズを考慮する具体例)

以下に、CPUの処理能力及びキューサイズを考慮する判断方法の具体例を説明する。

【0114】

まず、あらかじめ、能力に応じたポイントを設定しておく(負荷がかかるほどポイントは低く設定する)。

【0115】

50

CPUの処理能力テーブルの例を以下に示す。

【0116】

【表1】

CPU性能 (クロック周波数)	ポイント
500MHz 以下	0
500MHz~1GHz	1
1.1GHz~1.5GHz	2
1.6GHz~2GHz	3
2.1GHz~2.5GHz	4
2.6GHz 以上	5

10

【0117】

20

次に、キューサイズテーブルの例を示す。

【0118】

【表2】

キューサイズ	ポイント
10Mbyte 以下	0
11Mbyte~20Mbyte	1
21Mbyte~30Mbyte	2
31Mbyte~40Mbyte	3
41Mbyte~50Mbyte	4
51Mbyte 以上	5

30

【0119】

40

次に、あるページを処理する時点の能力を得点化する。

【0120】

例 : PDL処理 1Ghz 1ポイント  
 RIP処理 1.5Ghz 2ポイント  
 PDLキュー 20Mbyte 1ポイント  
 RIPキュー 50Mbyte 4ポイント

最後に、得点を元に重なり除去を行う所を決定する。

【0121】

PDL処理 = 計2ポイント (1+1=2)

50

RIP処理 = 計6ポイント (2+4=6)

この場合は、RIP処理の方が負荷が軽いので、重なり除去はRIP処理側で行うことを決定する。

【0122】

また、別ページを処理する時点の能力を得点化する。

【0123】

例 : PDL処理	1.5hz	2ポイント
RIP処理	1Ghz	1ポイント
PDLキュー	40Mbyte	3ポイント
RIPキュー	10Mbyte	0ポイント

10

次に、得点を元に重なり除去を行う所を決定する。

【0124】

PDL処理 = 計5ポイント (2+3=5)

RIP処理 = 計1ポイント (1+0=1)

この場合は、PDL処理の方が負荷が軽いので、重なり除去はPDL処理側で行うことを決定する。

【0125】

なお、上記具体例は2つの例を示したが、これに限定されず、より負荷の少なくなる処理に重なり除去を割り当てることが可能になることは明らかである。

【0126】

20

[実施形態1の変形例]

<任意の割合で、重なり除去済みDLと重なり除去なしDLを含む処理例>

ここで、本実施形態の変形例として、図14を参照して、任意の割合で重なり除去済みDLと重なり除去なしDLについて説明する。

【0127】

本実施形態1の変形例では、PDL処理待ちキュー404とRIP処理待ちキュー406に残っているキュー数、PDL処理部203とRIP処理部216で利用可能なCPUの個数から、PDL処理とRIP処理の負荷の大きさを確認する。ここで、例えば、PDL処理部203の負荷が70%、RIP処理部216の負荷が30%の状況だったと仮定する。この場合、より均等な負荷で処理を行うDLを生成しようとする場合、実施形態1のように100%をPDL処理かRIP処理かに分散するよりも、負荷に反比例して、PDL処理に30%、RIP処理に70%の負荷をかけるのが望ましい。

30

【0128】

図14の右側の例は、生成されたDLにおいて、1頁中の30%は重なり除去済みの領域、70%は重なり除去なしの領域が含まれるように処理する例を示した図である。RIP処理部におけるレンダリング時には、DLを参照して、重なり除去なし領域では、重ね合わせ処理を行うと共にその結果のファイルがレンダリングされ、重なり除去済み領域では、その領域のファイルがそのままレンダリングされる。

【0129】

40

<実施形態1の変形例の作用効果>

かかる実施形態1の変形例によれば、図13Bの処理の例でも、さらにPDL処理部とRIP処理部との負荷に応じて重なり除去が分散されるので、より短時間でビットマップ展開の処理が行われる。

【0130】

[実施形態2]

実施形態1では、重なり除去を行っていないDLと重なり除去済みDLの生成を、PDL処理部とRIP処理部の負荷を分散する目的で切り替えていた。それに対し実施形態2では、これらのDLの生成を、リアルタイムレンダリングと非リアルタイムレンダリングの観点で切り替える例を説明する。

【0131】

50

リアルタイムレンダリングとは、RIP処理の速度がプリンタのエンジンが紙を排紙する速度と同期するRIP処理のことを言う。すなわち、RIP処理後にデータをスプールするためのメモリ領域（以下、スプールメモリ：本例では、ページメモリ221に相当する）が不要である。一方、非リアルタイムレンダリングとは、RIP処理の速度とプリンタのエンジンが紙を排紙する速度とが同期しないRIP処理のことを言う。この場合は、RIP処理後にデータをスプールするためのメモリ領域が必要になる。

#### 【0132】

リアルタイムレンダリングを行うためには、プリンタのエンジン速度にRIP処理が追いつかなければならぬため、RIP処理の負荷が軽いことが望まれる。そのため、重なり除去済みDLであれば、重なり除去が必要でないため、リアルタイムレンダリングに向いたDLであるといえる。一方、MFPでは省メモリ化のために、メモリ領域が他の処理と共有化されることが考えられる。上記で述べたスプールメモリに関しては同様で、通常はPDLデータのプリント処理に用いられているが、場合によっては他の処理（例えば、FAXやコピーなど）がスプールメモリを占有することが起きる。

10

#### 【0133】

＜実施形態2のDLのつくり分けの判断例＞

上記前提を基にして、図15を用いて実施形態2の場合の重なり除去を行っていないDLと重なり除去済みDLの作り分けの判断例を説明する。

#### 【0134】

図15の上段の問題のない例1では、スプールメモリをコピー処理で占有してしまっている。しかし、PDL処理で重なり除去済みDLを生成したため、リアルタイムレンダリングが可能でスプールメモリを必要とせず、コピー処理と並列動作が可能になる。

20

#### 【0135】

また、図15の中段の問題のない例2では、スプールメモリが他処理に使われていない。従って、PDL処理で重なり除去を行っていないDLを生成し、非リアルタイムレンダリングを行ってスプールメモリをプリント処理で用いても問題がない。

#### 【0136】

一方、図15の下段の問題の例では、スプールメモリがコピー処理で使用しているのにも関わらず、PDL処理でリアルタイムレンダリングが行えない重なり除去を行っていないDLを生成してしまっている。そのため、スプールメモリが解放されるまで、プリント処理に待ちが発生してしまう。

30

#### 【0137】

＜実施形態2の画像処理装置の構成例＞

実施形態2の画像処理装置の構成は、基本的に実施形態1と同様であるので、繰り返すは避ける。

#### 【0138】

但し、図4に示した画像処理の機能構成では、PDL処理切替部403による重なり除去処理フラグの生成条件が、実施形態1の待ちキュー数とCPU数から変更される。すなわち、スプールメモリ（ページメモリ221）が他の処理（FAXやコピーなど）で専用中か否かが、重なり除去処理フラグの生成条件となる。また、図5では、スプールメモリ（ページメモリ221）他の処理で専用中か否かのフラグが保持される。

40

#### 【0139】

＜実施形態2の処理手順例＞

図16のフローチャートを参照して、実施形態2のPDL処理とRIP処理の負荷分散について、その処理手順を説明する。

#### 【0140】

（CPU205によるPDL処理での重なり除去の有無の指示）

図16のステップS1503において、CPU205は、PDLデータを待って、受信したPDLデータ206aを受け付け、PDL処理待ちキュー404にキューイングする。次に、CPU205は、ステップS1504において、PDL処理部203でDL生成

50

を開始する時点で、スプールメモリが他の処理に使われているかを確認する。次に、CPU205は、ステップS1505において、スプールメモリが他の処理に使われているか否かを判断して分岐する。

【0141】

スプールメモリが他の処理に使われていない場合 (S1505でYes) はリアルタイムレンダリングをする必要が無いため、CPU205はステップS1506において、重なり除去を行っていないDLの生成をPDL処理部203に指示する。スプールメモリが他の処理に使われていた場合 (S1505でNo) はリアルタイムレンダリングを行う必要があるので、CPU205はステップS1507において、重なり除去済みDLの生成をPDL処理部203に指示する。

10

【0142】

(PDL処理部及びRIP処理部のCPUによる処理)

以下、PDL処理部及びRIP処理部のCPUによる処理は、図12B及び図12Cと同様であるので、重複説明を避ける。

【0143】

<実施形態2の作用効果>

かかる実施形態2の処理によれば、RIP処理のレンダリングとプリンタエンジンの動作との同期が外れることによるプリントエラー (プリントデータのショートエラー) を無くして、効率の良いプリントが可能となる。

【0144】

20

[他の実施形態]

なお、実施形態1と2を独立した処理として説明したが、実施形態1と2とを組み合わせることにより、更に資源の有効利用と処理時間の短縮を図ることが出来る。また、実施形態1はキュー数 (負荷量) とCPU数 (処理能力) によるDLの重なり除去処理の有無、実施形態2はスプールメモリの空きか否かによるDLの重なり除去処理の有無を制御を示した。しかし、これに限定することなく、他の条件によるDLの重なり除去処理の有無を制御を行うことも本発明に含まれる。

【0145】

以上、様々な実施形態を詳述したが、本発明は、複数の機器から構成されるシステムに適用してもよいし、また、一つの機器からなる装置に適用してもよい。例えば、スキャナ、プリンタ、PC、複写機、複合機及びファクシミリ装置の如くである。

30

【0146】

本発明は、前述した実施形態の各機能を実現するソフトウェアプログラムを、システム若しくは装置に対して直接または遠隔から供給する。そして、そのシステム等に含まれるコンピュータが該供給されたプログラムコードを読み出して実行することによっても達成される。

【0147】

従って、本発明の機能・処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、上記機能・処理を実現するためのコンピュータプログラム自体も本発明の一つである。その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、OSに供給するスクリプトデータ等、プログラムの形態を問わない。

40

【0148】

プログラムを供給するための記録媒体としては、例えば、フレキシブルディスク、ハードディスク、光ディスク、光磁気ディスク、MO、CD-ROM、CD-R、CD-RWなどがある。また、記録媒体としては、磁気テープ、不揮発性のメモリカード、ROM、DVD (DVD-ROM、DVD-R) などもある。

【0149】

また、プログラムは、クライアントコンピュータのブラウザを用いてインターネット/インターネットのウェブサイトからダウンロードしてもよい。すなわち、該ウェブサイト

50

から本発明のコンピュータプログラムそのもの、もしくは圧縮され自動インストール機能を含むファイルをハードディスク等の記録媒体にダウンロードしてもよいのである。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるウェブサイトからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるWWWサーバも、本発明の構成要件となる場合がある。

#### 【0150】

また、本発明のプログラムを暗号化してCD-ROM等の記憶媒体に格納してユーザに配布してもよい。この場合、所定条件をクリアしたユーザにのみ、インターネット/インターネットを介してウェブサイトから暗号化を解く鍵情報をダウンロードさせる。そして、その鍵情報を暗号化されたプログラムを復号して実行し、プログラムをコンピュータにインストールしてもよい。

#### 【0151】

また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現されてもよい。なお、そのプログラムの指示に基づき、コンピュータ上で稼動しているOSなどが、実際の処理の一部または全部を行ってもよい。もちろん、この場合も、前述した実施形態の機能が実現され得る。

#### 【0152】

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

#### 【図面の簡単な説明】

#### 【0153】

【図1】本実施形態の画像形成装置(MFP)の構造例を示す側断面図である。

【図2】本実施形態におけるプリンタ制御部の構成例を示すブロック図である。

【図3】本実施形態におけるコントローラの構成の一例を示すブロック図である。

【図4】実施形態1のプリンタ制御部の機能構成例を示すブロック図である。

【図5】実施形態1で使用するプログラム及びデータの記憶例を示す図である。

【図6A】本実施形態における重なり除去を行っていないディスプレイリストについての処理例を示した図である。

【図6B】本実施形態における重なり除去済みディスプレイリストについての処理例を示した図である。

【図7A】本実施形態における重なり除去を行っていないディスプレイリストの詳細な構成例を示す図である。

【図7B】本実施形態における重なり除去済みディスプレイリストの詳細な構成例を示す図である。

【図8】本実施形態における重なり除去を行っていないディスプレイリストが重ね合わせ処理を行いレンダリングされる処理例を示した図である。

【図9】実施形態1におけるCPUの個数に関する説明をする図である。

【図10】実施形態1におけるCPU数が同じ場合の、重なり除去を行っていないディスプレイリストと重なり除去済みディスプレイリストを作り分ける基準例を示す図である。

【図11】実施形態1におけるCPU数が異なる場合の、重なり除去を行っていないディスプレイリストと重なり除去済みディスプレイリストを作り分ける基準例を示す図である。

【図12A】実施形態1におけるCPU205の処理手順例を示すフローチャートである。

【図12B】実施形態1におけるPDL処理部のCPUの処理手順例を示すフローチャートである。

トである。

【図12C】実施形態1におけるRIP処理部のCPUの処理手順例を示すフロー チャートである。

【図13A】従来例及び重なり除去を固定化した場合のPDL処理とRIP処理の負荷に偏りがある場合のパイプライン処理の例を示す図である。

【図13B】実施形態1の負荷分散方法を適応した場合のPDL処理とRIP処理のパイプライン処理の例を示した図である。

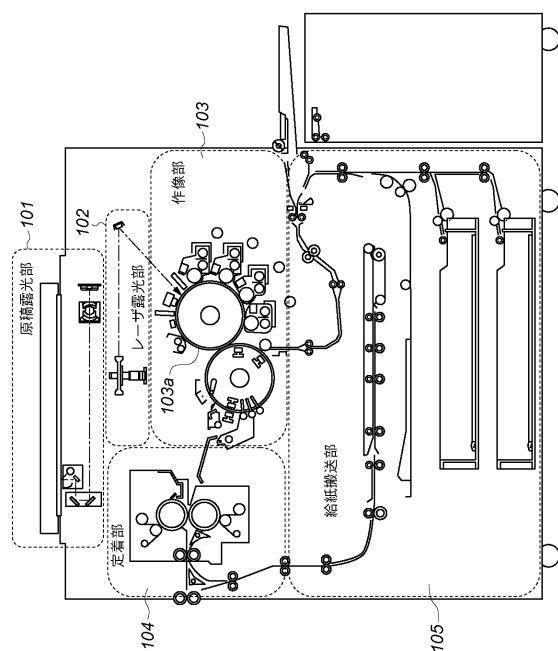
【図14】実施形態1の変形例における、重なり除去済み部分と重なり除去を行わない部分とが所定の割合で含まれたディスプレイリストの生成例を示した図である。

【図15】実施形態2で解決される課題として、問題のない例と問題のある例とを示した図である。

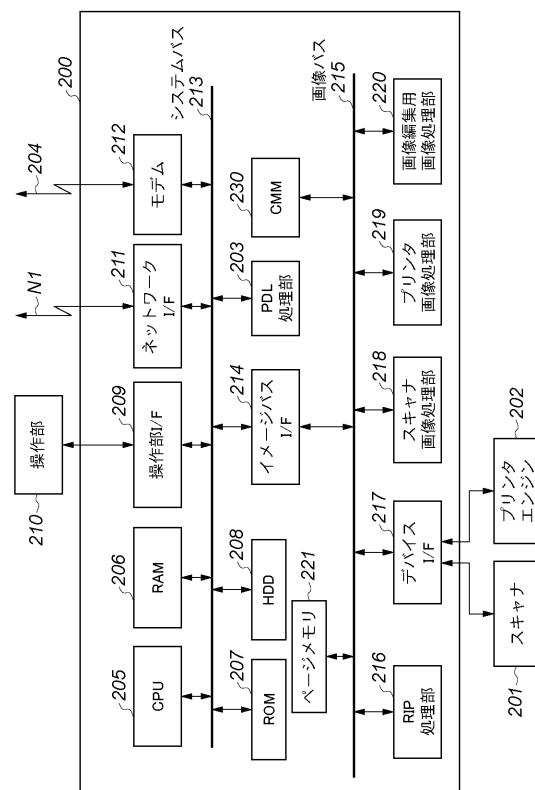
【図16】実施形態2におけるCPU205の処理手順例を示すフロー チャートである。

10

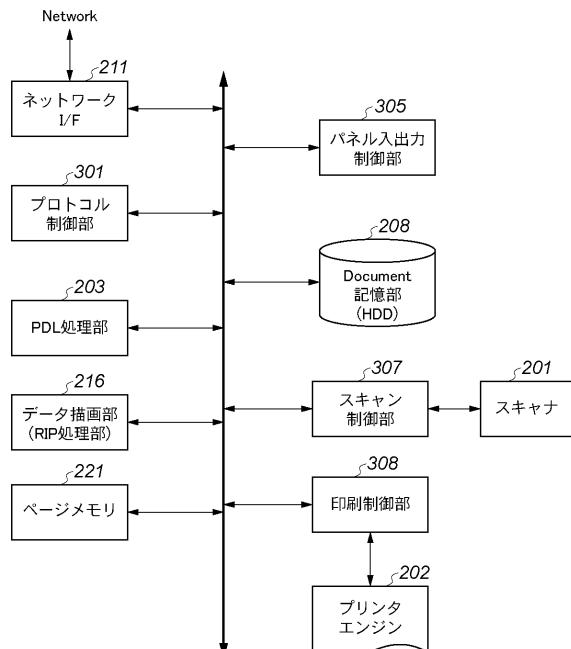
【図1】



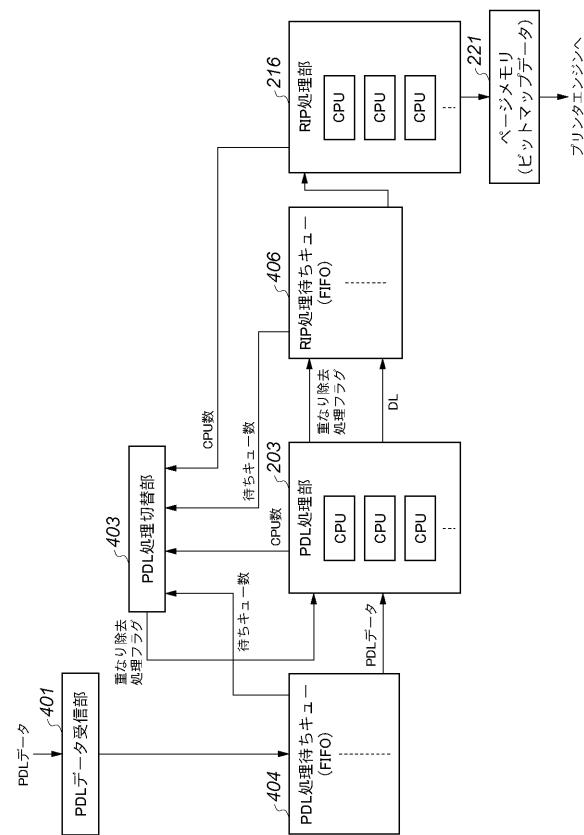
【図2】



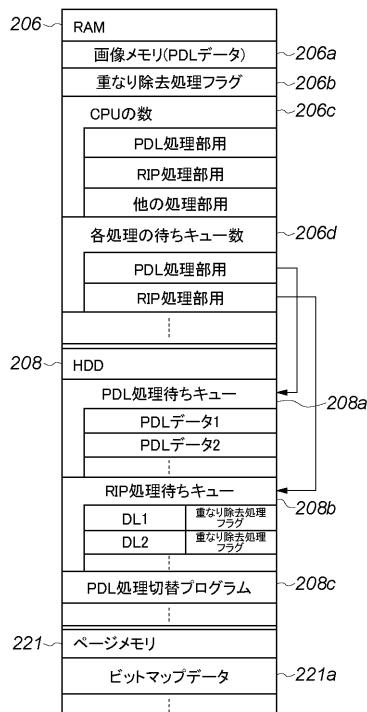
【図3】



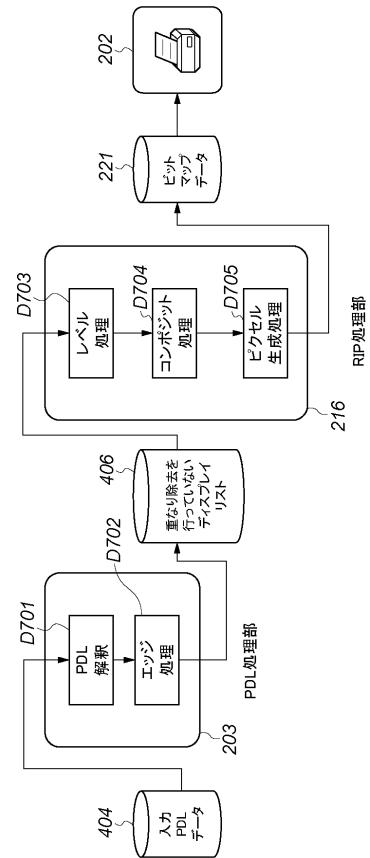
【図4】



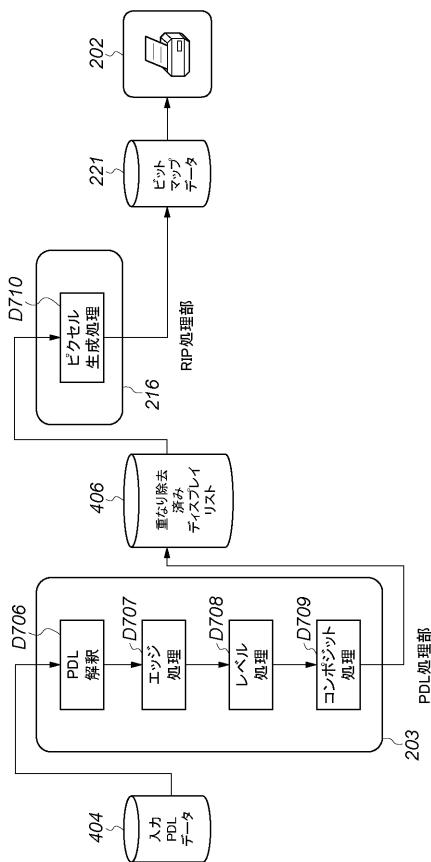
【図5】



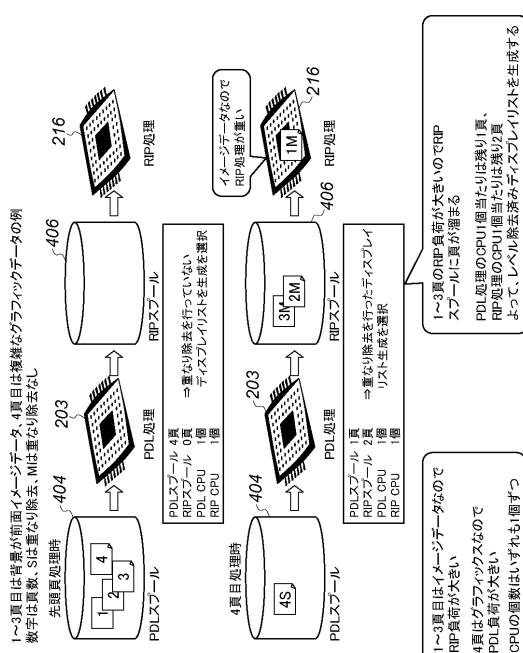
【図6 A】



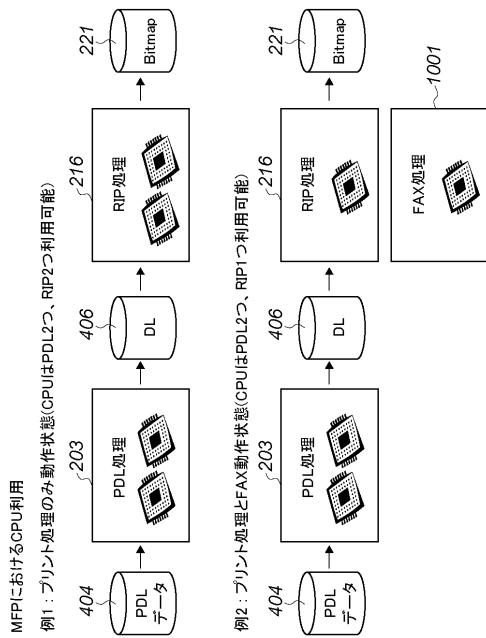
【図 6 B】



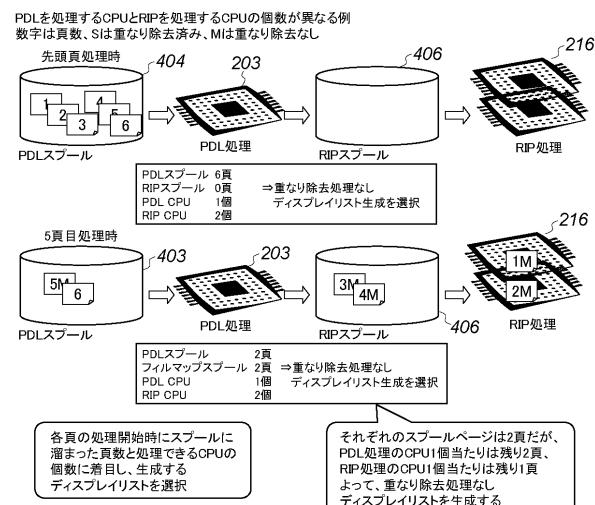
【図 10】



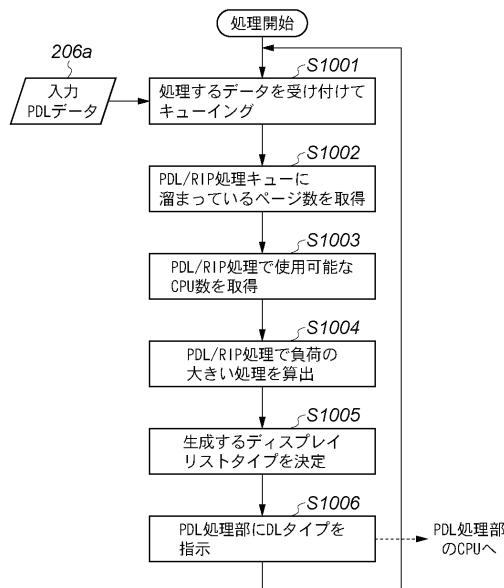
【図 9】



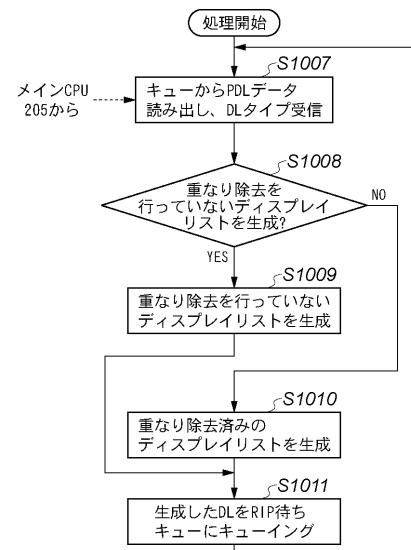
【図 11】



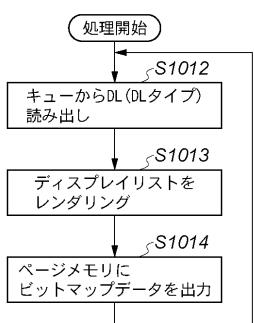
【図12A】



【図12B】



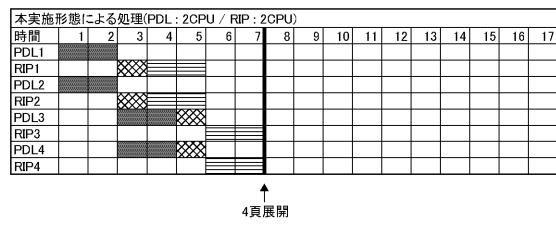
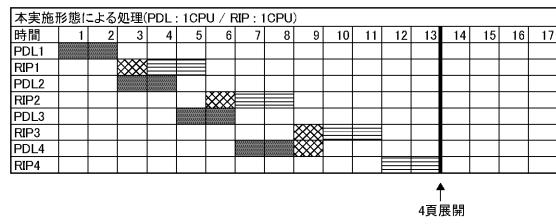
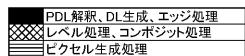
【図12C】



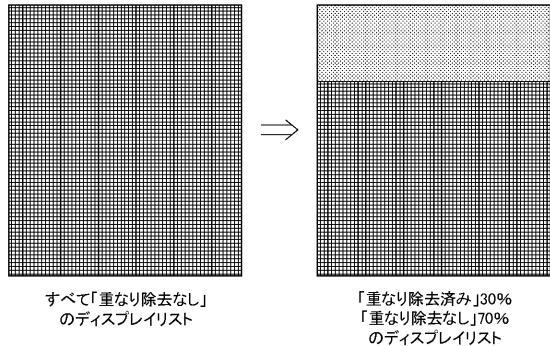
【図13A】



【図13B】

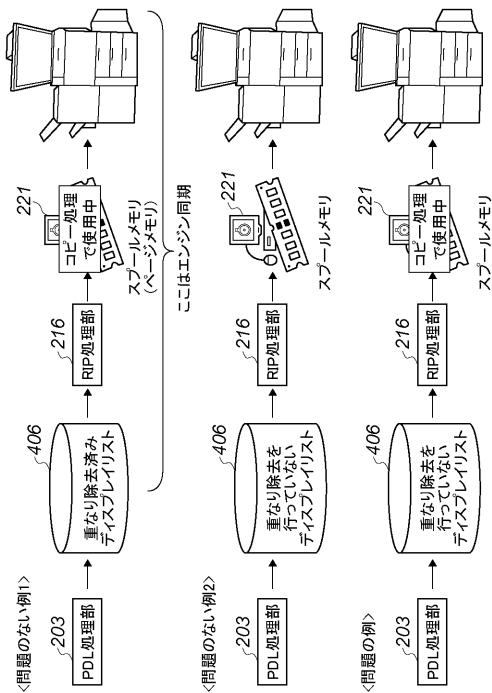


【図14】

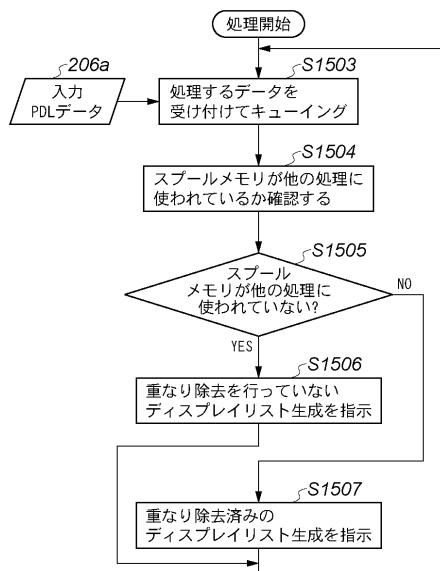


「重なり除去済み」30%  
「重なり除去なし」70%  
のディスプレイリスト

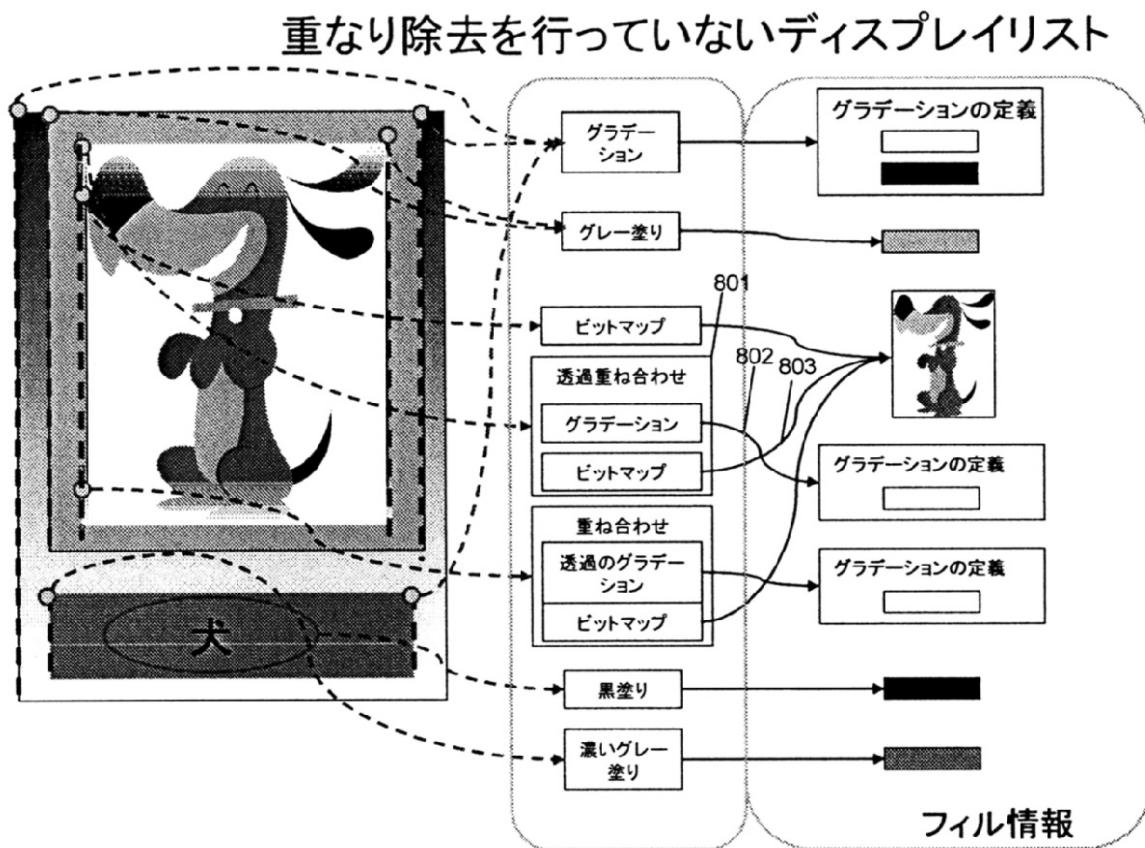
【図15】



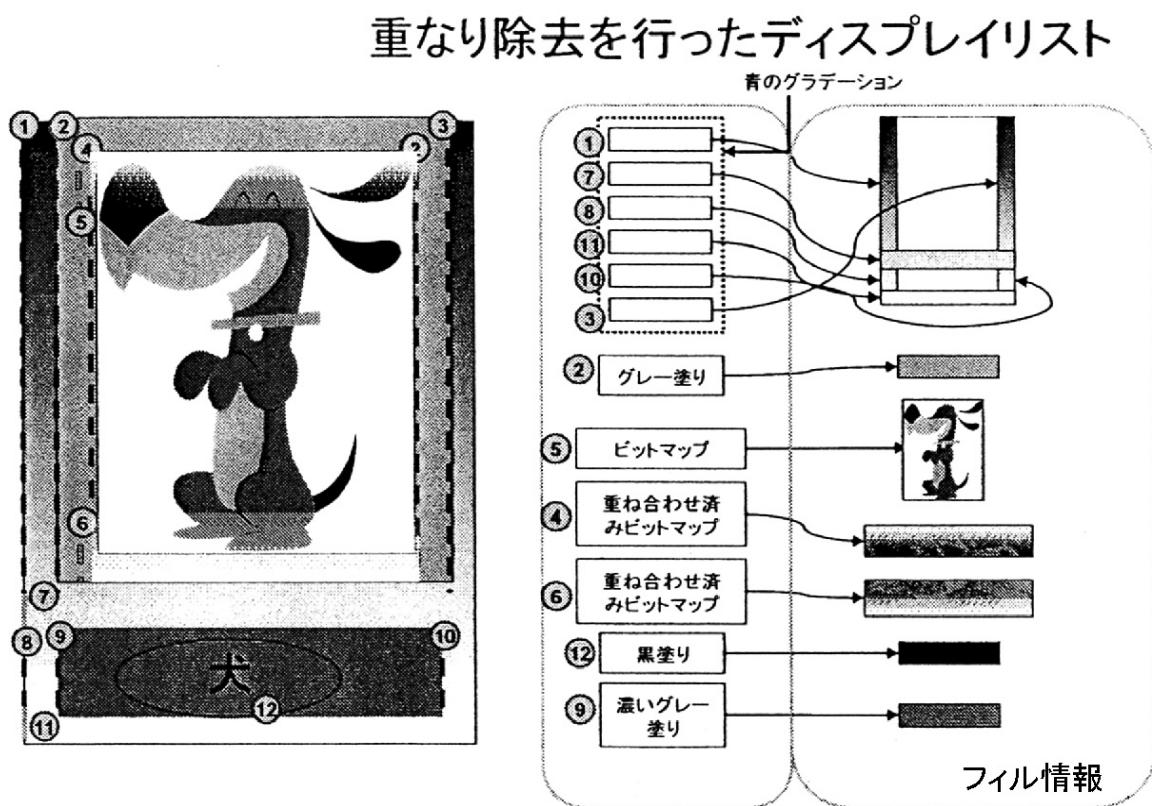
【図16】



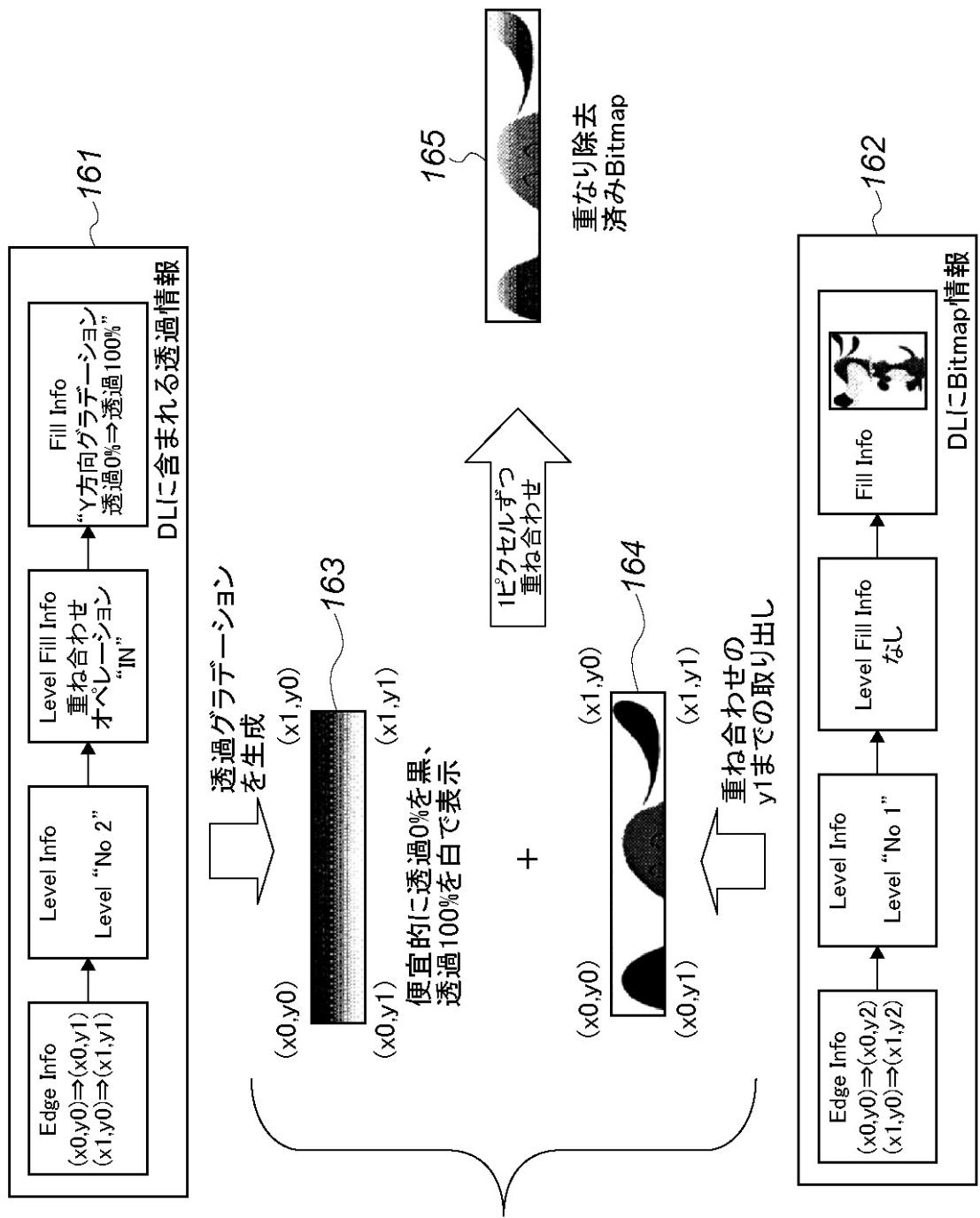
【図 7 A】



【図 7 B】



【図8】



---

フロントページの続き

(72)発明者 伊藤 盛一

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 牧島 元

(56)参考文献 特開2005-063027 (JP, A)

特開平10-052950 (JP, A)

特開平10-021020 (JP, A)

特開2006-178607 (JP, A)

国際公開第2007/064915 (WO, A1)

特開平09-190539 (JP, A)

特開2004-152267 (JP, A)

特開2002-103697 (JP, A)

特開2005-284384 (JP, A)

特開2009-005377 (JP, A)

(58)調査した分野(Int.Cl., DB名)

B 41 J 5 / 30

G 06 F 3 / 12