

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5641711号
(P5641711)

(45) 発行日 平成26年12月17日(2014.12.17)

(24) 登録日 平成26年11月7日(2014.11.7)

(51) Int.Cl.		F I			
B 4 1 J	5/30	(2006.01)	B 4 1 J	5/30	Z
H 0 4 N	1/00	(2006.01)	H 0 4 N	1/00	I O 7 A
G 0 6 F	3/12	(2006.01)	G 0 6 F	3/12	Z

請求項の数 13 (全 21 頁)

(21) 出願番号	特願2009-120395 (P2009-120395)	(73) 特許権者	000001007
(22) 出願日	平成21年5月18日 (2009.5.18)		キヤノン株式会社
(65) 公開番号	特開2010-264731 (P2010-264731A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成22年11月25日 (2010.11.25)	(74) 代理人	100076428
審査請求日	平成24年5月18日 (2012.5.18)		弁理士 大塚 康德
		(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】 画像形成装置、制御方法、及びプログラム

(57) 【特許請求の範囲】

【請求項1】

P D Lで記述された複数の描画命令を含む、P D Lデータを受信する受信手段と、
前記P D Lデータに含まれる前記複数の描画命令に対して決定処理を描画命令ごとに行
い、当該決定処理の結果に基づいて描画命令ごとの中間データを生成する生成手段と、
前記生成された中間データに基づいた描画を行うことでラスターイメージデータを生成
する描画手段と
を有し、

前記決定処理は、当該決定処理の対象の描画命令に対応するパラメータと、当該描画命
令の種類によって決定される閾値とを比較することで、当該描画命令とは別の描画命令が
生成されることになる所定の処理が、当該描画命令に適用されるか否かを決定する処理で
あることを特徴とする画像処理装置。

【請求項2】

P D Lで記述された複数の描画命令を含む、P D Lデータを受信する受信手段と、
前記P D Lデータに含まれる前記複数の描画命令に対して決定処理を描画命令ごとに行
い、当該決定処理の結果に基づいて描画命令ごとの中間データを生成する生成手段と、
前記生成された中間データに基づいた描画を行うことでラスターイメージデータを生成
する描画手段と
を有し、

前記決定処理は、当該決定処理の対象の描画命令に含まれるパラメータによって決定さ

れる値と、当該描画命令の種類によって決定される閾値とを比較することで、当該描画命令の種類に応じた所定の処理が当該描画命令に適用されるか否かを決定する処理であることを特徴とする画像処理装置。

【請求項 3】

前記生成手段は、

前記決定処理の結果が前記所定の処理が適用される旨を示す場合に、当該決定処理の対象の描画命令に前記所定の処理を適用することで別の描画命令を生成して、当該生成された描画命令から中間データを生成し、

前記決定処理の結果が前記所定の処理が適用されない旨を示す場合に、当該決定処理の対象の描画命令に前記所定の処理を適用せずに、当該描画命令から中間データを生成することを特徴とする請求項 1 または 2 に記載の画像処理装置。

10

【請求項 4】

前記パラメータは、描画命令によって描画される領域の面積を表すパラメータであることを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の画像処理装置。

【請求項 5】

前記描画命令の種類によって決定される閾値は、前記描画命令の種類によって異なることを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の画像処理装置。

【請求項 6】

前記描画命令の種類によって決定される閾値は、前記描画命令の種類によって異なる値を記憶するテーブルを用いて決定されることを特徴とする請求項 5 に記載の画像処理装置

20

【請求項 7】

前記描画命令は、シェーディング描画命令、タイル描画命令、パス描画命令を含むことを特徴とする請求項 1 乃至 6 の何れか 1 項に記載の画像処理装置。

【請求項 8】

前記シェーディング描画命令に対応する前記所定の処理は、前記シェーディング描画命令によって描画されるシェーディングのラスターイメージデータを描画する処理であることを特徴とする請求項 7 に記載の画像処理装置。

【請求項 9】

前記タイル描画命令に対応する前記所定の処理は、前記タイル描画命令によって描画されるタイルのラスターイメージデータを描画する処理である請求項 7 または 8 に記載の画像処理装置。

30

【請求項 10】

前記パス描画命令に対応する前記所定の処理は、前記パス描画命令に含まれる複数のパス点列中の点を削除する処理である請求項 7 乃至 9 の何れか 1 項に記載の画像処理装置。

【請求項 11】

P D L で記述された複数の描画命令を含む、P D L データを受信する受信工程と、前記 P D L データに含まれる前記複数の描画命令に対して決定処理を描画命令ごとに行い、当該決定処理の結果に基づいて描画命令ごとの中間データを生成する生成工程と、前記生成された中間データに基づいた描画を行うことでラスターイメージデータを生成する描画工程とを有し、

40

前記決定処理は、当該決定処理の対象の描画命令に対応するパラメータと、当該描画命令の種類によって決定される閾値とを比較することで、当該描画命令とは別の描画命令が生成されることになる所定の処理が、当該描画命令に適用されるか否かを決定する処理であることを特徴とする画像処理装置の制御方法。

【請求項 12】

P D L で記述された複数の描画命令を含む、P D L データを受信する受信工程と、前記 P D L データに含まれる前記複数の描画命令に対して決定処理を描画命令ごとに行い、当該決定処理の結果に基づいて描画命令ごとの中間データを生成する生成工程と、

50

前記生成された中間データに基づいた描画を行うことでラスターイメージデータを生成する描画工程と

を有し、

前記決定処理は、当該決定処理の対象の描画命令に含まれるパラメータによって決定される値と、当該描画命令の種類によって決定される閾値とを比較することで、当該描画命令の種類に応じた所定の処理が当該描画命令に適用されるか否かを決定する処理であることを特徴とする画像処理装置の制御方法。

【請求項 13】

請求項 1 乃至 10 の何れか 1 項に記載の各手段として、コンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本特許は、画像形成のための PDL 処理やレンダリング処理と呼ばれる画像形成装置、制御方法、及びプログラム処理に関する。特に、複数の処理装置を持つプリンタや複合機に代表される画像形成装置において、必要な PDL 処理、中間データ生成処理、及びレンダリング処理の高速化に関する。

【背景技術】

【0002】

従来、画像形成装置内で行われるレンダリングに関する処理は、複数の処理装置が各々画像形成にかかる処理を分担することで、画像形成を可能たらしめている。

【0003】

また、従来から画像形成処理に係る高速化手法として、異なるデバイス間における処理の分担による高速化が行われてきた。例えば非特許文献 2 では、情報処理装置（PC）と、画像形成装置（複合機）との間の処理にかかる高速化手法が示されている。この技術による画像形成処理の特徴は、情報処理装置と画像形成装置とで、画像形成に係る処理の負荷分散を行っている点にある。すなわち、ページレイアウト処理や、グラフィック処理、画像処理などの処理を情報処理装置と画像形成装置との間で処理コストの最適化を行い、情報処理装置からの印刷処理開始から印刷エンジンによって行われる印刷までにかかる時間を短縮している。

【0004】

また、上記のような画像形成処理を高速化する別の手法として、画像形成装置内の処理装置を複数にすることで高速化が行われてきた。例えば特許文献 1 では、二つの CPU とレンダリングを行う「レンダリングエンジン」の三つの処理装置で処理を分担し、大量のデータに対し並行処理を行うことで高速に処理する手法が示されている。この技術による画像形成処理は、異なる処理が可能な処理装置を複数接続し、画像形成に係る処理を分割して、各々の処理装置内で画像形成にかかる処理を行うことで高速に画像生成を行っている。

【先行技術文献】

【特許文献】

【0005】

【特許文献 1】特開平 10 - 307924 号公報

【非特許文献】

【0006】

【非特許文献 1】キヤノン株式会社、"Canon technology コントローラーアーキテクチャ"、[online]、平成 21 年、キヤノン株式会社、[平成 21 年 3 月 26 日検索]、インターネット<URL: http://web.canon.jp/technology/canon_tech/explanation/ir_controller.html>

【非特許文献 2】キヤノン株式会社、"Canon technology UFR/LIPS V"、[online]、平成 21 年、キヤノン株式会社、[平成 21 年 3 月 26 日検

10

20

30

40

50

索]、インターネット、<URL: http://web.canon.jp/technology/canon_tech/explanation/hspeed_processing.html>

【発明の概要】

【発明が解決しようとする課題】

【0007】

上述したように、従来技術では、複数の異なる処理を、複数の異なる処理装置を用いて実行することで、画像形成に係る処理の高速化を実現している。

【0008】

しかし、従来手法では、画像形成処理に係る、PDL解析処理(すなわち、PDLインタプリタ)は、PDLデータに基づきPDLデータ解析し中間データを生成する。そして、レンダリング処理(すなわち、レンダラー)は、生成した中間データに基づき、レンダリング処理を行い、ラスターイメージデータを生成する。上記、各々の処理ステップにおいて、その実行中の処理が次のステップの処理速度を低下せしめるか否かに関係なく処理を続行した結果、処理速度を低下せしめるという問題があった。より具体的には、従来技術に基づくレンダリング処理は、大量に重なった描画命令や、大量の奥行き情報を持つ描画命令を持つ中間データを受け取ると、その中間データが持つ奥行き情報や重なり情報の煩雑さによって処理速度が著しく低下する。従来技術では、このレンダラーの処理速度を著しく低下せしめるかどうかに関係なくPDL解析処理を行う。つまり、PDLインタプリタはたとえすでに生成された中間データに、解析中のPDLデータの解析に因る中間データの追加がレンダリング処理の速度を著しく低下させるとしても、PDL解析処理を続行する。

【0009】

よって、従来技術には、画像形成に係る処理中に、すでに生成された中間データに対する解析中のPDLデータに基づく中間データ追加が、その両者の特徴に依存してレンダリング処理の速度を著しく低下せしめるか否かを判断していない。上記判断をしていないことによって、画像形成に係る処理中に、レンダリング処理が高速になるように前もってPDL解析中に最適化処理を行うことができないという問題があった。

【課題を解決するための手段】

【0010】

本発明は、上記の従来技術の課題を鑑み、以下の構成を備える。すなわち、PDLで記述された複数の描画命令を含む、PDLデータを受信する受信手段と、前記PDLデータに含まれる前記複数の描画命令に対して決定処理を描画命令ごとに行い、当該決定処理の結果に基づいて描画命令ごとの中間データを生成する生成手段と、前記生成された中間データに基づいた描画を行うことでラスターイメージデータを生成する描画手段とを有し、前記決定処理は、当該決定処理の対象の描画命令に対応するパラメータと、当該描画命令の種類によって決定される閾値とを比較することで、当該描画命令とは別の描画命令が生成されることになる所定の処理が、当該描画命令に適用されるか否かを決定する処理である。

【発明の効果】

【0011】

本発明では、予めPDLデータ解析中において、レンダリング処理の最適化を行うことにより、画像形成に係る処理中にレンダリング処理が高速になる。

【図面の簡単な説明】

【0012】

【図1】本実施形態における情報処理装置と画像形成装置とを説明する図。

【図2】本実施形態におけるPDLデータ、及び、中間データを説明する図。

【図3】本実施形態における画像形成に係る処理の流れを説明する図。

【図4】本実施形態における画像形成装置を説明する図。

【図5】本実施形態におけるPDL解析処理からレンダリング処理までの流れを説明する図。

10

20

30

40

50

【図 6】本実施形態における P D L インタプリタによる高速化処理の流れを説明する図。

【図 7】本実施形態におけるソフトウェアモジュールの関係を示す図。

【図 8】線形、及び放射シェーディングを説明するための図。

【図 9】パターンを説明するための図。

【図 10】文字描画命令の例を説明するための図。

【図 11】P D L データによるオブジェクト描画を説明するための図。

【図 12】セルをタイル状に描画することを説明するための図。

【図 13】本実施形態における中間データ生成コストを求める処理のフローチャートを示す図。

【図 14】本実施形態における描画命令それぞれの中間データ生成コストを示す図。

10

【図 15】本実施形態に係る各処理の処理時間の経過を説明するための図。

【発明を実施するための形態】

【0013】

< 第一の実施形態 >

以下、本発明を実施するための形態について図面を用いて説明する。なお、本明細書では、P D L インタプリタが P D L データを得て P D L データを解析する処理を P D L 解析処理、もしくは P D L 処理と呼ぶ。また、本明細書では、中間データ生成部が上記 P D L データの解析に基づき中間データを生成し、レンダラーがこの中間データからラスターイメージデータを得る処理をレンダリング処理と呼ぶ。

【0014】

20

[画像形成に係る処理]

図 1 は本発明の第一の実施形態に係る画像形成装置 1 2 と画像形成装置 1 2 に接続された情報処理装置 1 1 の構成を示すブロック図である。情報処理装置 1 1 は、P C やパーソナルコンピュータ、ワークステーションなどが該当する。情報処理装置 1 1 は同装置上で P D L データ 1 0 0 を生成することが出来る。さらに情報処理装置 1 1 はネットワークや U S B に代表されるインタフェースを経由して P D L データ 1 0 0 を含むデータを情報処理装置 1 1 の外部の装置に転送することが出来る。画像形成装置 1 2 は、プリンタや複合機などと呼ばれる画像形成装置 1 2 であり、P D L データ 1 0 0 に代表されるデータをもとに、紙に代表されるような媒体に可視像を形成することが出来る。

【0015】

30

図 2 において、図 2 (a) は P D L データ 1 0 0、図 2 (b) は印刷データ 1 0 2 を説明する図である。P D L データ 1 0 0 は描画データ 1 0 0 - 1 と印刷制御データ 1 0 0 - 2 を含む。描画データ 1 0 0 - 1 は画像形成装置が形成する 0 以上の可視情報を持つ。印刷制御データ 1 0 0 - 2 は紙サイズや面付け処理、色変換処理などに係る情報を持つ。

【0016】

次に、画像形成装置 1 2 の上述した紙などの媒体に可視像を形成するための、画像形成に係る処理を説明する。図 3 は、画像形成装置 1 2 が行う画像形成に係る処理のフロー図である。まず、s 1 0 でネットワーク装置 1 2 - 1 が情報処理装置 1 1 から P D L データを得る。ネットワーク装置 1 2 - 1 は I E E E 8 0 2 . 3 に代表される手法で他装置と通信を行うことで、上記 P D L データ含むデータを得ることが出来る。また、この s 1 0 の処理は U S B 装置 1 2 - 2 を用いて他装置と通信を行い、上記 P D L データ含むデータを得ても本発明を実施できる。この U S B 装置は、U S B 2 . 0 に代表される手法で他装置と通信可能である。このほかにも画像形成装置 1 2 を I E E E 1 3 9 4 a や R S - 2 3 2 C または、I E E E 1 2 8 4 に準拠したインタフェースを介して他装置と接続することで、上記 P D L データ含むデータを得ても本発明は実施可能である。上述したような手法で、画像形成装置 1 2 が情報処理装置 1 1 に代表される他装置から得た P D L データは、次に画像形成装置 1 2 内の画像形成処理装置 1 2 - 3 が受け取る (s 1 1)。画像形成処理装置 1 2 - 3 は、この P D L データに基づいて印刷データを生成する。

40

【0017】

次に、s 1 1 で画像形成処理装置 1 2 - 3 が P D L データ 1 0 0 を得る。画像形成処理

50

装置 1 2 - 3 は得た P D L データに基づいて s 1 2 で印刷データ 1 0 2 を生成する。次に s 1 4 で、印刷エンジン 1 2 - 4 がこの印刷データ 1 0 2 を得る。印刷データ 1 0 2 は図 2 (b) が示すように、ラスタイメージデータと、印刷エンジン制御データを持つ。印刷エンジンは得た印刷データ 1 0 2 をもとに画像形成を行い、紙などの媒体上に可視像を形成する。

【 0 0 1 8 】

[画像形成処理装置内の処理]

次に画像形成の中でも、上述した画像形成処理装置 1 2 - 3 内で行われる図 3 の s 1 2 で示した処理についてより詳細に説明する。図 4 は画像形成装置 1 2 が有する画像形成処理装置 1 2 - 3 の構成の例を図示している。画像形成処理装置 1 2 - 3 は、C P U に代表される汎用プロセッサであるレンダラー装置 1 2 - 3 - 1 を備える。レンダラー装置 1 2 - 3 - 1 は中間データを得て、ラスタイメージデータを生成する。処理装置 1 2 - 3 - 2 及び 1 2 - 3 - 3 は、P D L 解析処理や、中間データの生成処理を行う。処理部としての機能を有し、C P U などがこれに該当する。なお、本実施形態では、画像形成処理装置 1 2 - 3 は処理装置 1 2 - 3 - 2、1 2 - 3 - 3 を二つ有するが、これに限定するわけではなく、二つ以上の複数個であれば本発明を実施することができる。また、本実施形態では画像形成処理装置 1 2 - 3 はレンダラー装置 1 2 - 3 - 1 を一つのみ有するが、これに限定するわけではなく、1 つ以上であれば本発明を実施できる。また、本実施形態ではレンダラー装置 1 2 - 3 - 1 が中間データを得てラスタイメージデータを生成するが、処理装置 (1 2 - 3 - 2、または 1 2 - 3 - 3) がどのように分担等をして処理を行っても、本発明を実施できる。また、レンダラー装置 1 2 - 3 - 1 は、中間データを得てラスタデータを生成する A S I C や D S P、D R P など C P U 以外の情報処理装置を用いても本発明を実施できる。

【 0 0 1 9 】

メモリコントローラ 1 2 - 3 - 5 は、メインメモリ 1 2 - 3 - 6 及び R O M 1 2 - 3 - 9 へのインタフェースである。即ち、メモリコントローラ 1 2 - 3 - 5 はメインメモリ 1 2 - 3 - 6 へのデータの読み込み及び書き込みを可能たらしめる。また、メモリコントローラ 1 2 - 3 - 5 は R O M 1 2 - 3 - 9 へのデータの読み込みを可能たらしめる。I O コントローラ 1 2 - 3 - 8 は画像形成処理装置 (1 2 - 3) 外の装置とのインタフェースである。即ち、I O コントローラ 1 2 - 3 - 8 は画像形成処理装置 (1 2 - 3) 外の装置とのデータの交換を可能たらしめる。ローカルバス 1 2 - 3 - 7 はレンダラー装置 1 2 - 3 - 1、処理装置 (1 2 - 3 - 2 及び 1 2 - 3 - 3)、メモリコントローラ 1 2 - 3 - 5、I O コントローラ 1 2 - 3 - 8 を接続する。即ち、ローカルバス 1 2 - 3 - 7 はこれら接続した装置間でデータの交換を可能たらしめる。以上、本実施形態では説明した構成にて画像形成処理装置 1 2 - 3 は、P D L データ 1 0 0 に基づいて印刷データを生成する。

【 0 0 2 0 】

次にこの印刷データ生成に係る処理ステップを、図 5 を用いながら説明する。先ず、画像生成処理装置 1 2 - 3 の処理装置 (1 2 - 3 - 2、または 1 2 - 3 - 3) は P D L データ 1 0 0 の解析処理を行う (s 1 2 0)。次いで処理装置 (1 2 - 3 - 2、または 1 2 - 3 - 3) は上記 P D L データ 1 0 0 の解析から中間データを生成する (s 1 2 1)。ついで、この中間データをもとにレンダラー装置 1 2 - 3 - 1 がラスタイメージデータ 1 3 - 5 を生成する。

【 0 0 2 1 】

なお、この P D L データ 1 0 0 は複数の種類がある。例えば A d o b e S y s t e m s、I n c o r p o r a t e d の A d o b e P D F や P o s t S c r i p t、C a n o n I n c . の L I P S など知られている。処理装置 (1 2 - 3 - 2、または 1 2 - 3 - 3) 上で動作する P D L インタプリタ 1 3 - 1 及び中間データ生成部 1 3 - 2 は、これら異なるデータフォーマットが存在する P D L データ 1 0 0 から、レンダラーが解析可能な共通のデータ形式である中間データ 1 0 2 を生成する。この中間データ 1 0 2 に基づき、レンダラー 1 3 - 3 はレンダラー装置 1 2 - 3 - 1 を使いラスタデータ 1 3 - 5 を生

10

20

30

40

50

成している。

【 0 0 2 2 】

図 7 に上記の流れの概要を示した本発明の実施形態におけるソフトウェアモジュールの関係を示す。また、

[P D L データ解析処理における高速化処理]

次に、図 5 にて示した上記 P D L データの解析の処理 (s 1 2 0) について、図 6 を用いながら説明する。まず、P D L インタプリタは入力可能な P D L データ 1 0 0 の少なくとも一部があるか否かを判定する (s 1 2 0 - 1)。もし P D L データ 1 0 0 がなければ、P D L 処理を終了する。次に P D L インタプリタは、もし P D L データ 1 0 0 の少なくとも一部があれば、P D L データを入力する (s 1 2 0 - 2)。次に P D L インタプリタは、入力した P D L データは、P D L インタプリタが高速化処理可能な処理か否かを判定する (s 1 2 0 - 3)。これにより高速化処理判定を実施する。本実施形態において、高速化処理が可能な処理は以下のものがある。

- ・圧縮データにかかる処理
- ・シェーディング描画命令処理
- ・タイル描画命令処理
- ・パス描画命令処理
- ・色変換処理
- ・フォント描画命令処理
- ・ビットマップ描画命令処理
- ・エッジ描画命令処理

しかし、これらの処理に限定するわけではなく、本発明が適用可能な処理は、その他 P D L インタプリタが実行する処理などにも適用可能である。なお、本実施形態において、便宜上、各種処理に対し、高速化処理を施し、中間データを生成するためのコストを高速化処理コストと称し、高速化処理を施さず、中間データを生成するコストを中間データ生成コストと称する。また、本実施形態では、高速化処理を施した場合と施さない場合とのコストを比較し、その差異により、適用を判定する。この処理コスト判定手段について、以下詳細に説明する。

【 0 0 2 3 】

もし、P D L データが高速化可能な命令であったならば、s 1 2 0 - 4 に進む。もし P D L が高速化不可能な命令であったならば、s 1 2 0 - 1 1 に進む。次に、P D L インタプリタは、注目 P D L データ群の高速化処理計算コストを計算する (s 1 2 0 - 4)。各処理のより詳細なコスト計算方法は、次の項にて述べる。

【 0 0 2 4 】

次に、P D L インタプリタは、注目 P D L データ群の中間データ生成コストを求める (s 1 2 0 - 5)。次に、P D L インタプリタは上記高速化処理コストと中間データ生成コストとを比較する (s 1 2 0 - 6)。この比較の結果、中間データ生成コストが大きければ、s 1 2 0 - 7 に進む。逆に中間データ生成コストが高速化処理コスト以下であれば s 1 2 0 - 1 1 に進む。なお、高速化処理コストが大きい場合は、当該注目 P D L データの処理を、高速化処理を適用せず継続した方が、適していることを示しており、中間データ生成コストが大きい場合は、当該注目 P D L データの処理を高速化処理を適用して行う方が適していることを示している。

【 0 0 2 5 】

次に、P D L インタプリタ 1 3 - 1 は各処理装置 (1 2 - 3 - 2 及び 1 2 - 3 - 3) の空き状況 (占有率) を問い合わせる (s 1 2 0 - 7)。次に、P D L インタプリタは上記取得した各処理装置の空き状況と予め定義している閾値との比較を行う (s 1 2 0 - 8)。もし、空き状況が、上記閾値を超えていれば s 1 2 0 - 9 に進む。もし上記閾値を越えていなければ s 1 2 0 - 1 1 に進む。次に P D L インタプリタは高速化処理のためのスレッドを生成する (s 1 2 0 - 9)。次に P D L インタプリタは生成したスレッドの中で高速化するために注目した P D L データに対して高速化処理を行う (s 1 2 0 - 1 0)。そ

10

20

30

40

50

の後、PDLインタプリタは中間データ生成部にPDLインタプリタの処理結果を送る（s 1 2 0 - 1 1）。そして、s 1 2 0 - 1へ戻り、上記の処理を繰り返す。

【 0 0 2 6 】

〔注目PDLデータ群の高速化処理コストを求める手法〕

次に、PDLインタプリタ13-1が、s 1 2 0 - 6において、上記中間データ生成コストと比較するための、高速化処理コストを求める手法を述べる。PDLインタプリタ13-1が、注目PDLデータ群の高速化処理コストを求める手法（s 1 2 0 - 4）の実施形態を説明する。すなわち、高速化処理コスト決定手段の機能を実現する。

【 0 0 2 7 】

PDLインタプリタ13-1は、PDLデータ100中の高速化処理可能な命令があると（s 1 2 0 - 3）、高速化処理コストを計算する（s 1 2 0 - 4）。本実施形態では以下の命令が高速化処理の対象となる。

- ・圧縮データにかかる処理
- ・シェーディング描画命令処理
- ・タイル描画命令処理
- ・パス描画命令処理
- ・色変換処理
- ・フォント描画命令処理
- ・ビットマップ描画命令処理
- ・エッジ描画命令処理

10

20

本実施形態では、上記それぞれの処理に対して、以下の手法（算出式）にて計算することで高速化処理コストを計算する。

【 0 0 2 8 】

<圧縮データにかかる処理>

圧縮データの高速化処理にかかるコストを C_{cmp} 、圧縮データのサイズを cmp_size 、圧縮データの圧縮方式 cmp_type に対応したコストの重み付けパラメータ（負荷パラメータ）を $weight[cmp_type]$ 、圧縮データ処理の高速化処理にかかる初期コストを cmp_init とすると、以下の式から圧縮データの高速化処理コストを求めることができる。

【 0 0 2 9 】

【数1】

$$C_{cmp} = cmp_size \times weight[cmp_type] + cmp_init$$

30

【 0 0 3 0 】

なお、圧縮データの圧縮方式 cmp_type に対応する重み付けパラメータは、予め定義しておくものとする。

【 0 0 3 1 】

<シェーディング描画命令処理>

シェーディング描画命令の高速化処理にかかるコストを $C_{shading}$ 、シェーディングのサイズを $shading_size$ 、シェーディングタイプ $shading_type$ に対応したコストの重みづけパラメータを $weight[shading_type]$ 、シェーディング描画命令高速化処理にかかる初期コストを $shading_init$ とすると、PDLインタプリタ13-1は以下の式から圧縮データの高速化処理コストを求めることができる。

40

【 0 0 3 2 】

【数2】

$$C_{shading} = shading_size \times weight[shading_type] + shading_init$$

【 0 0 3 3 】

ここで、シェーディング描画命令のシェーディングタイプとは、シェーディング命令の種類を指し、例えば、以下のようなものがある。シェーディングタイプの一つに線形シェーディングがある。線形シェーディングによる描画結果の例を図8(a)に示す。図8(

50

a)にあるように、線形シェーディングによる描画領域は、二つ以上の制御点間を線形な数式で表わされる色変化に基づき描画する。

【0034】

また、他のシェーディングタイプの一つに放射シェーディングがある。放射シェーディングによる描画結果の例を図8(b)に示す。この図にあるように、放射シェーディングによる描画領域は、二つ以上の円、または楕円間を線形な数式で表わされる色変化に基づき描画する。

【0035】

<タイル描画命令処理>

タイル描画命令の高速化処理にかかるコストを C_{tile} 、タイル描画命令の処理コストの重み付けパラメータを $tile_weight$ 、タイル描画命令で描画する領域のサイズを $tile_area_size$ 、タイル描画命令中のセルのサイズを $tile_cell_size$ 、タイル描画命令高速化処理にかかる初期コストを $tile_init$ とすると、PDLインタプリタ13-1は以下の式からタイル描画命令の高速化処理コストを求めることができる。

【0036】

【数3】

$$C_{tile} = tile_weight \times \frac{tile_area_size}{tile_cell_size} + tile_init$$

【0037】

タイル描画命令で描画する領域とタイルのセルを、図9を使って説明する。タイル描画命令は、ある描画命令を一定の描画領域内で繰り返し描画する描画命令である。この繰り返すべき描画命令をセル300-1と呼ぶ(図9(a)参照)。そして、レンダラー13-3は、タイル描画命令中に示されるタイル描画命令で描画する領域300-2に上記タイルのセルを繰り返し描画することで、図9(b)に示されるタイル描画結果を生成する。

【0038】

<パス描画命令処理>

パス描画命令の高速化処理にかかるコストを C_{path} 、パス描画命令の処理コストの重み付けパラメータを $path_weight$ 、パス描画命令で描画するパス点列数を $path_size$ 、パス描画命令高速化処理にかかる初期コストを $path_init$ とすると、PDLインタプリタ13-1は以下の式からパス描画命令の高速化処理コストを求めることができる。

【0039】

【数4】

$$C_{path} = path_weight \times path_size + path_init$$

【0040】

<色変換処理>

色変換処理の高速化にかかるコストを C_{cms} 、色変換処理の処理コストの重み付けパラメータを cms_weight 、必要な色変換数を cms_size 、必要な色変換処理の種類を cms_type 、必要な色変換処理の種類 cms_type に対応した処理の重みづけパラメータの配列を $cms_weight[cms_type]$ 、色変換処理高速化処理にかかる初期コストを cms_init とすると、PDLインタプリタ13-1は以下の式から色変換処理の高速化にかかるコストを求めることができる。

【0041】

【数5】

$$C_{cms} = cms_weight \times cms_size \times cms_weight[cms_type] + cms_init$$

【0042】

なお、PDLインタプリタ13-1は、上記必要な色変換数以下の方法から求めることができる。例えば、ビットマップイメージに対しては、PDLインタプリタ13-1は画

10

20

30

40

50

素数から求めることができる。また、例えば、パス点列で表現される閉空間に内を一色に塗る描画命令に対しては、PDLインタプリタ13-1は必要な色変換数は1であると定めることができる。

【0043】

<フォント描画命令処理>

フォント描画命令処理の高速化にかかるコストをCfont、フォント描画命令の処理コストの重み付けパラメータをfont_weight、フォント描画命令中にあるフォントサイズの種類数をfont_max、このフォントサイズ種類数に対応した描画が必要なフォントの文字サイズの配列をfont_size[i]、フォントサイズの種類数に対応したフォントの文字数の配列をfont_num[i]、フォント描画命令高速化処理にかかる初期コストをfont_initとすると、PDLインタプリタ13-1は以下の式からフォント描画命令処理にかかるコストを求めることができる。

10

【0044】

【数6】

$$C_{font} = font_weight \times \sum_{i=0}^{font_max} font_size[i] \times font_num[i] + font_init$$

【0045】

なお、上記フォント描画命令と上述したfont_max, font_size, font_numが有する値については図10を使い例示しながら説明する。フォント描画命令中には同図に示されるような複数の描画すべき文字列200-0が含まれる。この例においては、フォント描画命令は三つに分類されるフォント描画単位200-1、200-2、200-3を含んでいる。そして、フォント描画命令はそれぞれのフォントサイズごとに描画すべき複数の文字を有する。例えば、フォントサイズが20の文字の集合200-1は5つの文字を有する。また、フォントサイズが30の文字の集合200-2は4つの文字を有する。また、フォントサイズが19の文字の集合200-3は4つの文字を有する。すなわち、図10の場合は、フォントサイズの種類数font_maxは以下の数値が与えられる。

20

font_max = 3

また同図の場合は、フォントサイズの種類数に対応した描画が必要なフォントの文字サイズの配列font_sizeは以下の三つの数値を持つ。

30

font_size[0] = 20

font_size[1] = 30

font_size[2] = 19

また同図の場合は、フォントサイズの種類数に対応したフォントの文字数の配列font_numは以下の三つの数値を持つ。

font_num[0] = 5

font_num[1] = 4

font_num[2] = 4

以上の値に基づいて、高速化処理コストを求めることが可能である。

40

【0046】

<ビットマップ描画命令処理>

ビットマップ描画命令処理の高速化にかかるコストをCbitmap、ビットマップ描画命令の処理コストの重み付けパラメータをbitmap_weight、ビットマップ描画命令中にあるビットマップのサイズをbitmap_size、ビットマップ描画命令中のビットマップの色数(色チャンネル数)をbitmap_channel、ビットマップ描画命令加速処理にかかる初期コストをbitmap_initとすると、PDLインタプリタ13-1は以下の式からビットマップ描画命令処理にかかるコストを求めることができる。

【0047】

【数 7】

$$C_{\text{bitmap}} = \text{bitmap_weight} \times \text{bitmap_size} \times \text{bitmap_channel} + \text{bitmap_init}$$

【0048】

<エッジ描画命令処理>

エッジ描画命令処理の高速化にかかるコストを C_{edge} 、エッジ描画命令処理が描画すべき領域サイズを edge_size 、エッジ描画命令処理中の色数を edge_channel 、エッジ描画命令処理中の色数（色チャンネル数） edge_channel と処理コストの重み付けパラメータを対応付ける配列を $\text{edge_weight}[\text{edge_channel}]$ 、エッジ描画命令加速処理にかかる初期コストを edge_init とすると、PDLインタプリタ 13-1 は以下の式からエッジ描画命令処理にかかるコストを求めることができる。

【0049】

【数 8】

$$C_{\text{edge}} = \text{edge_size} \times \text{edge_weight}[\text{edge_channel}] + \text{edge_init}$$

【0050】

以上、本実施形態において対象となるそれぞれの処理における、中間データ生成コストを求めることが可能である。

【0051】

[注目 PDL データ群の中間データ生成コストを求める手法 1]

PDLインタプリタ 13-1 は、s120-6 にて上記高速化処理コストとの比較対照である中間データ生成コストを求める。本実施形態における、PDLインタプリタ 13-1 が、注目 PDL データ群の中間データ生成処理コストを求める手法 (s120-5) を説明する。すなわち、これにより中間データ生成コスト決定手段を実現する。

【0052】

図 13 は注目 PDL データ群の中間データ生成コストを求める手法 (s120-5) の処理フローを説明する図である。s120-5-1 でまず、PDLインタプリタ 13-1 は既に PDLインタプリタ 13-1 及び中間データ生成部 13-2 によって生成された中間データのサイズを取得する。

【0053】

次に、PDLインタプリタ 13-1 は s120-5-2 で、求めた中間データのサイズが予め定義した閾値以上であるかを判定する。この判定に用いられる閾値は、予め当該画像形成処理装置の処理能力に依存して定められる。例えば、メインメモリ 12-3-6 の容量によって閾値を定める。また、処理装置 (12-3-2、12-3-3) の処理速度によって定めても、本実施形態を実施できる。また、メインメモリ 12-3-6 の容量と処理装置 (12-3-2、12-3-3) の処理速度との両方の性能を鑑みて定義しても、本実施形態を実施できる。

【0054】

次に、中間データのサイズが閾値以上であるかの判定が真であれば、PDLインタプリタ 13-1 は s120-5-3 に進む。また、中間データのサイズが閾値以上であるかの判定が偽であれば、PDLインタプリタ 13-1 は s120-5-4 に進む。PDLインタプリタ 13-1 は、上記判定結果に基づき、s120-5-3 で中間データ生成にかかるコストに設定可能な最大値を設定する。ここでいう最大値とは例えば以下の通りである。すなわち、32 bit のデータを扱える処理装置 (12-3-2、12-3-3) である場合には、PDLインタプリタ 13-1 は中間データ生成にかかるコストに対する変数に 32 bit の大きさのメモリを割り当てる。よって、PDLインタプリタ 13-1 は最大値として 2147483647 を設定する。

【0055】

また、PDLインタプリタ 13-1 は、上記判定に基づき、s120-5-4 で中間データ生成にかかるコストに設定できる最小値を設定する。ここでいう最小値とは例えば以下の通りである。すなわち、32 bit のデータを扱える処理装置 (12-3-2、12

10

20

30

40

50

- 3 - 3) である場合には、PDLインタプリタ13-1は中間データ生成にかかるコストに対する変数に32bitの大きさのメモリを割り当てる。よって、PDLインタプリタ13-1は最小値として-2147483648を設定する。

【0056】

[高速化処理コストと中間データ生成コストの比較処理]

次に、高速化処理コストと中間データ生成コストとの比較処理(s120-6)について、例を用いて説明する。図11はタイル描画命令とビットマップ描画命令とを有するPDLデータ100によって描画されるべき画像を例示している。すなわち、PDLデータ中のビットマップ描画命令に基づき、PDLインタプリタ13-1とPDL最適化モジュール13-6と中間データ生成部13-2とレンダラー13-4とによって描画される(10
ビットマップ300-3が描画される)。次に、PDLデータ中のセル300-1を領域300-2にしきつめるタイル描画命令に基づきPDLインタプリタ13-1とPDL最適化モジュール13-6と中間データ生成部13-2とレンダラー13-4とによって描画される(セル300-1が領域300-2に描画される)。次に、PDLデータ中のパス描画命令に基づきPDLインタプリタ13-1とPDL最適化モジュール13-6と中間データ生成部13-2とレンダラー13-4とによって描画される(パス300-4が描画される)。この結果、図11に示される描画結果は得られる。

【0057】

次に、このPDLデータ100の例における、中間データ生成コストの比較処理(s120-6)を示す。PDLインタプリタ13-1は、最初に描画されるビットマップ描画命令においては、先に行われるs120-5でメモリに最小値が設定されるので、s120-6では、偽と判定する。すなわち、PDL最適化モジュール13-6による高速化処理を行うことなく、中間データ生成部13-2が中間データ102を生成することとなる。

【0058】

また、PDLインタプリタ13-1は、次に描画されるタイル描画命令においては、先に行われるs14-0-5でメモリに最大値が設定されるので、s120-6では真と判定する。上記判定によって、次にPDLインタプリタ13-1は、s120-7に進む。

【0059】

ここでPDLインタプリタ13-1は、各処理装置12-3-2、12-3-3の空き状況を取得する。この空き状況は、例えば、OS13-7の機能を使うことによって得られる処理装置の占有率から求めることができる。これにより占有率取得手段を実現する。すなわち、処理装置が、既に他の処理によって高負荷にある場合には、高速化処理を行うための新しい処理を開始するだけの処理装置の余裕がない。よって、処理装置によって実行される高速化処理を行うべきではないと判定できる。そして、PDLインタプリタ13-1は、s120-8にて、予め定められた閾値と上記処理装置の占有率とを比較する。この比較によって、PDLインタプリタ13-1は、高速化処理を行うべきか判定できる。もしこの判定が真である場合には、PDLインタプリタ13-1は高速化処理のためのスレッドの生成あるいは初期化をOS13-7の機能を使うことによって実行する(s120-9)。この処理によって、処理装置に割り当てるべき高速化処理の処理単位を生成40
することができる。つまり、この処理により、処理単位生成判定手段を実現する。

【0060】

もちろん、本実施形態においては、処理単位をスレッドとして説明したが、この新しく割り当てる処理単位がスレッドではなくプロセスであっても本実施形態を実施できる。また、この新しく割り当てる処理単位がスレッドではなくタスクであっても本実施形態を実施できる。さらには、この新しく割り当てる処理単位がスレッドではなくその他並行して動作することができるルーチンや関数の集合であっても本実施形態を実施できる。

【0061】

次に、PDLインタプリタ13-1は上記割り当てた処理単位に注目しているPDLデータ中の命令群の処理を割り当て、高速化処理を行う(s120-10)。すなわち、P 50

D Lインタプリタ13-1は、上記生成した処理単位にタイル描画命令にかかる処理を割り当てる。そして、この新しく割り当てられた処理単位は、P D Lインタプリタ13-1によって割り当てられた高速化処理が実行される。なお、高速化処理に関する説明は次の項に記載する。そして、P D Lインタプリタ13-1は、新しい処理単位によって行われた処理に基づき中間データ生成部13-2にデータを送り、中間データ生成部13-3によって、中間データ102を生成する(s120-11)。次に、P D Lインタプリタ13-1は、次のP D Lデータ中の命令群を読み込み、処理を続行する(s120-1)。

【0062】

[新しい処理単位によって行われる高速化処理]

次に、P D Lインタプリタ13-1が、生成する新しい処理単位による高速化処理について以下に説明する(図6におけるs120-10)。なお、本実施形態における高速化処理の対象となる処理は先に述べた高速化処理コストを求めたものと同様である。これにより高速化処理実行手段を実現する。なお、高速化処理の内容は、負荷分散や、処理対象データの削減などP D Lデータに含まれる処理ごとに異なるため、個別に説明する。

【0063】

<圧縮データにかかる処理>

本実施形態における圧縮データにかかる処理に対する高速化処理は、例えば以下の通りである。即ち、上記新しい処理単位において、圧縮データの伸張ないし、その一部の処理を行う。これによって、P D Lインタプリタ13-1で行われる圧縮データにかかる処理の負荷を低減せしめる。

【0064】

<シェーディング描画命令処理>

本実施形態におけるシェーディング描画命令処理に対する高速化処理は以下の通りである。即ち、レンダラー装置12-3-1で行われるべきシェーディング処理を、処理装置(12-3-2、または12-3-3)上で行う。これによって、シェーディング描画命令処理の中間データ生成処理を分担し、その負荷を低減せしめる。

【0065】

<タイル描画命令処理>

本実施形態におけるタイル描画命令処理に対する高速化処理は、例えば以下の通りである。即ち、レンダラー装置12-3-1で行われるべきタイル描画命令に係るセル内部のラスターデータ生成処理を処理装置(12-3-2、または12-3-3)上で行う。これによってタイル描画命令処理の中間データ生成処理を分担し、その負荷を低減せしめる。

【0066】

<パス描画命令処理>

本実施形態におけるパス描画命令処理に対する高速化処理は、例えば以下の通りである。即ち、パス点列中の描画に不要な重複する点を削除する。また、パス点列中の三点が直線状に並んでいる場合には、描画に不要な中間点を削除する。これによってパス描画命令処理の中間データ生成処理の負荷を低減せしめる。

【0067】

<色変換処理>

本実施形態における色変換処理に対する高速化処理は、例えば以下の通りである。即ち、P D Lインタプリタ13-1は、中間データ生成部13-2で行われるべき色変換処理を行う。これによって色変換処理に係る中間データ生成処理の負荷を低減せしめる。

【0068】

<フォント描画命令処理>

本実施形態におけるフォント描画命令処理に対する高速化処理は、例えば以下の通りである。即ち、連続する文字列からなる描画命令をひとつのエッジ描画命令にまとめる。これによってフォント描画命令処理に係る中間データ生成処理の負荷を低減せしめる。

【0069】

10

20

30

40

50

<ビットマップ描画命令処理>

本実施形態におけるビットマップ描画命令処理に対する高速化処理は、例えば以下の通りである。即ち、画像形成に必要な解像度以上の情報を持つビットマップ描画命令処理の解像度変換をする。ここでの解像度変換とは、本実施形態ではバイキュービック法による画像解像度変換によって行われる。もちろん、上記解像度変換は最近傍法によって実装されても、本実施形態を実施できる。また、これに限定されず、解像度変換は線形補完法やその他の方法によって実装されても、本実施形態を実施できる。これによってビットマップ描画命令処理に係る中間データ生成処理の負荷を低減せしめる。

【0070】

<エッジ描画命令処理>

本実施形態におけるエッジ描画命令処理に対する高速化処理は、例えば以下の通りである。エッジ描画命令から両端の点のみを扱い、それ以外の点の処理を省略する。これによってエッジ描画命令処理に係る中間データ生成処理の負荷を低減せしめる。

【0071】

以上により、各処理ごとに高速化処理の手法を適用し、中間データ生成を行う。なお、ここで挙げた各処理への高速化の手法は一例であり、例えば、ビットマップ描画命令処理などで述べたように、他の方法にて実施しても良い。

【0072】

[第一の実施形態の効果]

以上、本発明によって以下のような効果がある。すなわち、一般に計算機は、既に多くの処理が行われメモリなどのリソースを多く消費している場合には、処理速度が著しく低下するか、処理を継続することが困難となる。よって、本発明では、PDLデータに定義された各種処理の独特な処理特性に注目し、予め異なる処理単位で高速化処理を行い、中間データ生成にかかる処理の負荷を低減する。これによって、より高速な画像形成処理を可能とする。具体的には本実施形態では、PDLインタプリタ13-1は適用する処理に必要なコストを比較する(s120-6)。すなわち、高速化処理を行わずに中間データを生成する場合のコストと、高速化処理を行って中間データを生成した場合のコストとを比較する。すなわち、PDLインタプリタ13-1、及び、中間データ生成部13-2が現状のままPDLデータに対し中間データ生成処理を続けた場合に、画像形成処理装置の制限によって処理速度が低下せしめるかどうかを判定する。

【0073】

ここでいう、制限とは上述したように、例えばメモリの容量による制限がある。もし、PDLデータの処理に対し高速化処理を行った方が、中間データ生成のコストが小さいと判定できた場合には、中間データを生成する前に、一旦新しい処理単位を生成すべきかを判定する(s120-7、120-8)。ここでは、PDLインタプリタ13-1は、新しく処理単位による処理を行う余地が処理装置あるかどうかを判定している(120-8)。もし、上記判定が真であるならば、PDLインタプリタ13-1は、新しい処理単位を処理装置に割り当てて、PDLデータ中の注目した命令群に係る処理を行う(s120-9、s120-10)。

【0074】

上述したこれらの処理によって、PDLインタプリタ13-1は、PDLデータに独特な処理特性に注目し、実行中の処理が次のステップの処理速度を低下するか否かを判定し、その上で処理速度を向上させている。つまり、図15に示すように、処理速度が向上することが実験的に明らかとなっている。すなわち、従来手法でタイル描画処理のみを行った時と異なり、先行する描画命令処理によって中間データが生成されていると、次に処理すべき描画命令による処理速度が著しく低下することとなっていた。これに対して本実施形態の手法では、上記PDLデータに独特な処理特性に注目し、著しい速度低下を防ぐことによって、処理速度を向上させる。

【0075】

以上、本発明では、大量に重なった描画命令や、大量の奥行き情報を持つ描画命令を持

10

20

30

40

50

つ中間データを受け取ると、その中間データが有する奥行き情報や重なり情報の煩雑さによって処理速度が著しく低下するという問題に対して、レンダラーの処理速度を著しく低下させるか否かを判定しながらPDL解析処理を行う。つまり、PDLインタプリタはすでに生成された中間データの特徴と、解析中のPDLデータの特徴と、複数ある処理装置の占有率情報とを使い、新しく上記処理単位を上記処理装置が実行する処理として生成するか否かを判定する。上記判定に基づき、処理速度が著しく低下する条件を検知し、最適化をすることで、画像形成の処理全体に係る時間を短縮する。

【0076】

<第二の実施形態>

第一の実施形態において、注目PDLデータ群の中間データ生成コストを求めたが、第二の実施形態として、異なる中間データ生成コストを求める手法を説明する。

10

【0077】

[注目PDLデータ群の中間データ生成コストを求める手法2]

PDLインタプリタ13-1は、s120-6にて上記高速化処理コストと比較対照となる中間データ生成コストを求める。本実施形態において説明する手法は、PDLインタプリタ13-1が、注目PDLデータ群の中間データ生成コストを求める手法において第一の実施形態とは異なる手段である。

【0078】

中間データ生成コストをCdl、中間データ生成部13-2が生成する中間データ102に蓄積された描画命令数(処理数)をdl_op_num、中間データ生成コストの重み付けパラメータをdl_weightとすると、PDLインタプリタ13-1は、以下の式により中間データ生成コストを求め、決定することができる。

20

【0079】

【数9】

$$C_{dl} = dl_weight \times \log dl_op_num$$

【0080】

以上により、本実施形態においても、第一の実施形態と同様の効果を得ることが可能となる。

【0081】

30

<第三の実施形態>

第一の実施形態において、注目PDLデータ群の中間データ生成コストを求めたが、第三の実施形態として、異なる中間データ生成コストを求める手法を説明する。

【0082】

[注目PDLデータ群の中間データ生成コストを求める手法3]

PDLインタプリタ13-1は、s120-6にて、上記高速化処理コストと比較対照となる中間データ生成コストを求める。本実施形態において説明する手法は、PDLインタプリタ13-1が、注目PDLデータ群の中間データ生成コストを求める手法において第一の実施形態とは異なる手段である。

【0083】

40

中間データ生成コストをCdl、中間データ生成部13-2が生成する中間データ102に蓄積された描画命令数をdl_op_num、中間データ生成コストの重み付けパラメータをdl_weightとすると、PDLインタプリタ13-1は、以下の式により中間データ生成コストを求めることができる。

【0084】

【数10】

$$C_{dl} = dl_weight \times dl_op_num^n$$

【0085】

以上により、本実施形態においても、第一の実施形態、及び第二の実施形態と同様の効

50

果を得ることが可能となる。

【 0 0 8 6 】

< 第四の実施形態 >

第一の実施形態において、注目 P D L データ群の中間データ生成コストを求めたが、第四の実施形態として、異なる中間データ生成コストを求める手法を説明する。

【 0 0 8 7 】

[注目 P D L データ群の中間データ生成コストを求める手法 4]

P D L インタプリタ 1 3 - 1 は、s 1 2 0 - 6 にて、上記高速化処理コストと比較対照となる中間データ生成コストを求める。本実施形態において説明する手法は、P D L インタプリタ 1 3 - 1 が、注目 P D L データ群の中間データ生成コストを求める手法において第一の実施形態とは異なる手段である。

10

【 0 0 8 8 】

中間データ生成コストを C_{dl} 、中間データ生成部 1 3 - 2 が生成する中間データ 1 0 2 に蓄積された描画命令数を dl_op_num 、中間データ生成コストの重み付けパラメータを dl_weight とすると、P D L インタプリタ 1 3 - 1 は、以下の式により中間データ生成コストを求めることができる。

【 0 0 8 9 】

【 数 1 1 】

$$C_{dl} = dl_weight \times dl_op_num$$

20

【 0 0 9 0 】

以上により、本実施形態においても、第一乃至第三の実施形態と同様の効果を得ることが可能となる。

【 0 0 9 1 】

< 第五の実施形態 >

第一の実施形態において、注目 P D L データ群の中間データ生成コストを求めたが、第五の実施形態として、異なる中間データ生成コストを求める手法を説明する。

【 0 0 9 2 】

[注目 P D L データ群の中間データ生成コストを求める手法 5]

P D L インタプリタ 1 3 - 1 は、s 1 2 0 - 6 にて、上記高速化処理コストと比較対照となる中間データ生成コストを求める。本実施形態において説明する手法は、P D L インタプリタ 1 3 - 1 が、注目 P D L データ群の中間データ生成コストを求める手法において第一の実施形態とは異なる手段である。

30

【 0 0 9 3 】

中間データ生成コストを C_{dl} 、中間データ生成コストの重み付けパラメータを dl_weight および dl_weight_2 、比較対象となる描画命令の種類を表す値を $object_type$ 、上記オブジェクト種類と D L 生成処理のコストを対応付ける配列を $object_type_2_cost$ 、既に生成した描画命令数を $generated_dl_size$ とすると、P D L インタプリタ 1 3 - 1 は中間データ生成コストを求めることができる。なお、比較対象の描画命令とすでに生成した描画命令との種類は同一のものである。

40

【 0 0 9 4 】

【 数 1 2 】

$$C_{dl} = dl_weight \times object_type_2_cost[object_type] \times generated_dl_size^{dl_weight_2}$$

【 0 0 9 5 】

次に、比較対象となる描画命令の種類を表す値 $object_type$ 、及び、オブジェクト種類と D L 生成処理のコストを対応付ける配列 $object_type_2_cost$ について説明する。図 1 4 は描画命令の種類と描画命令の種類を表す値 $object_type$ との対応を表す。また、図 1 4 は描画命令の種類を表す値 $object_type$ と D L 生成処理のコストを対応付ける配列 $object_type_2_cost$ と

50

の対応を表す。画像形成装置 12 はあらかじめ、この表 (図 1 4) によって定義される情報に対応した配列等を有する。これによりコスト情報保持手段を実現する。

【 0 0 9 6 】

この図 1 4 における予め定義された情報によって、例えば、圧縮データにかかる描画命令であれば、PDL インタプリタ 1 3 - 1 はDL生成処理コストが4であることが得られる。

【 0 0 9 7 】

同様に、定義された情報により、第一の実施形態にて挙げた処理それぞれのDL生成処理コストは、シェーディング描画命令処理は10、タイル描画命令処理は10、タイル描画命令処理は1、パス描画命令処理は1、色変換命令処理は2、フォント描画命令処理は2、ビットマップ描画命令処理は7、エッジ描画命令処理は3であることを取得できる。

10

【 0 0 9 8 】

以上により、本実施形態においても、第一乃至第四の実施形態と同様の効果を得ることが可能となる。

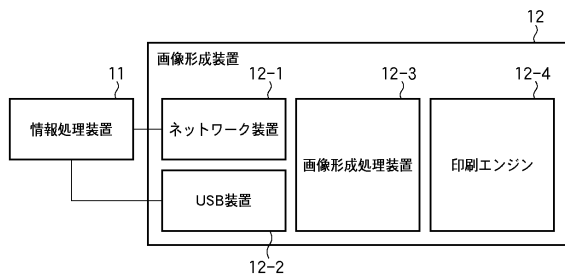
【 0 0 9 9 】

< その他の実施形態 >

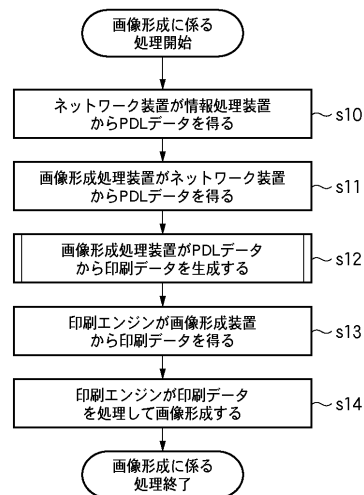
なお、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムをコンピュータ (またはCPU やMPU) が読み出し実行することによっても、達成される。この場合、プログラムは図示したフローの手順を実現するためのものを含むこととなる。

20

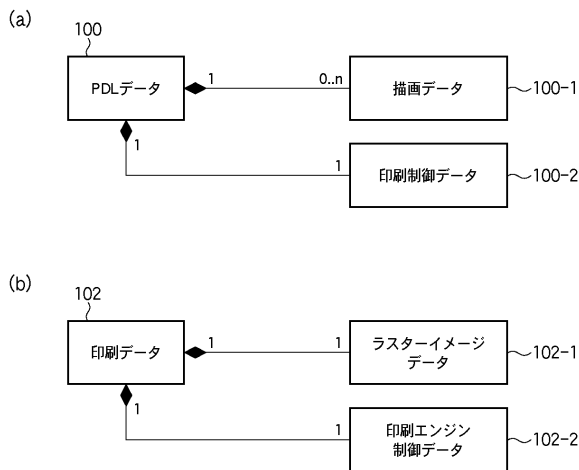
【 図 1 】



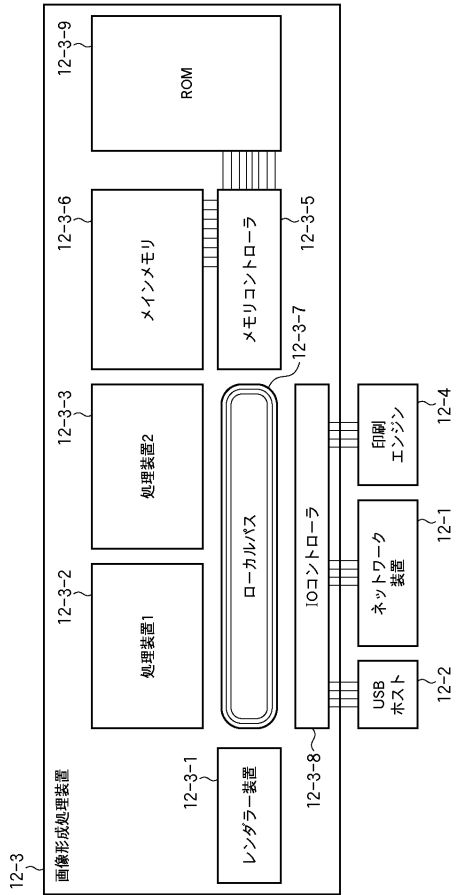
【 図 3 】



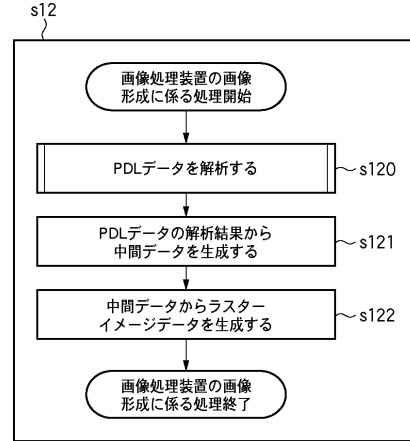
【 図 2 】



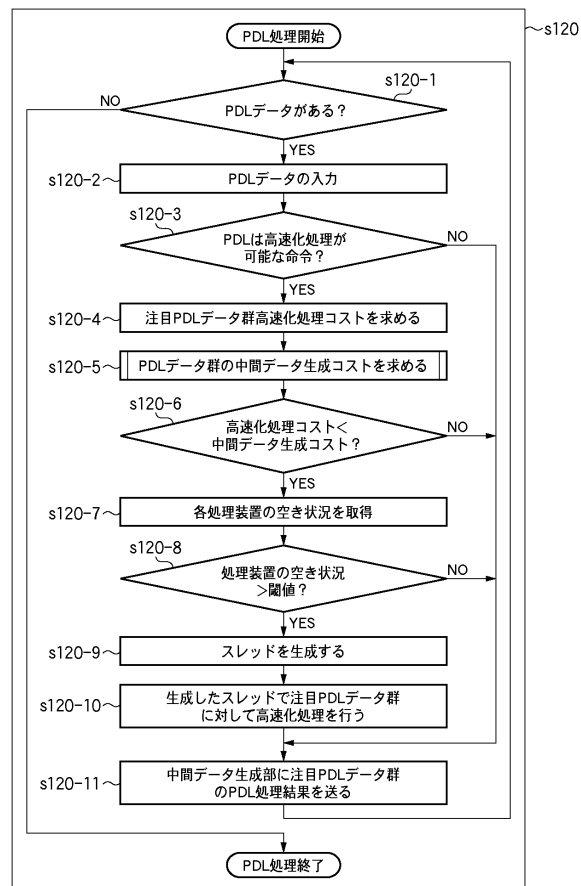
【図4】



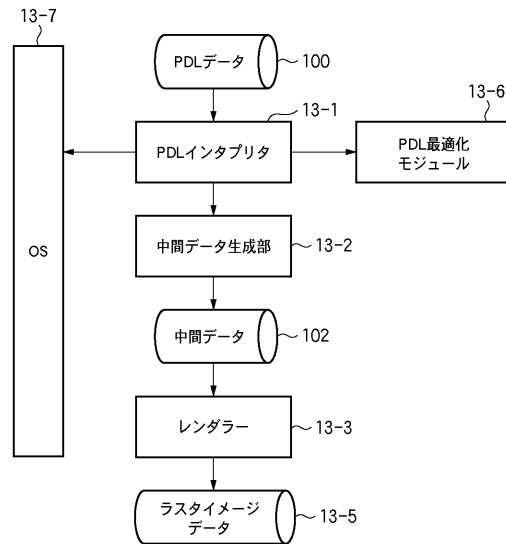
【図5】



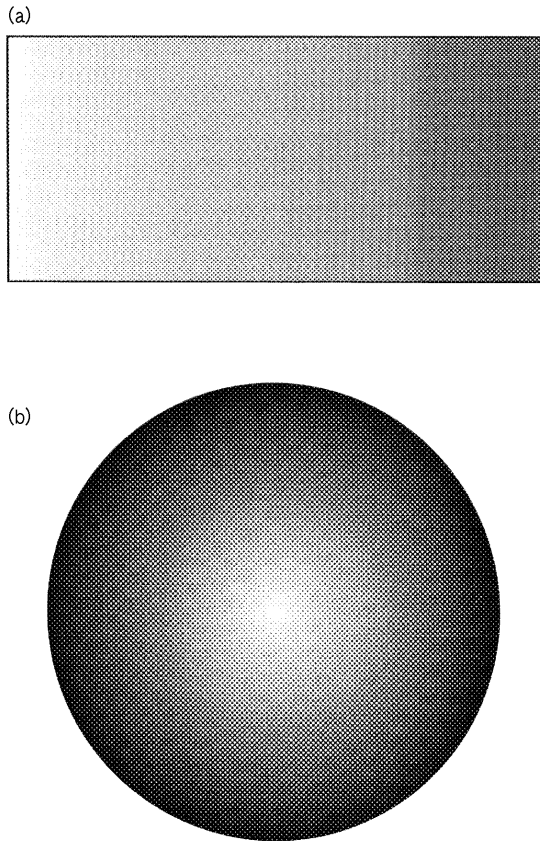
【図6】



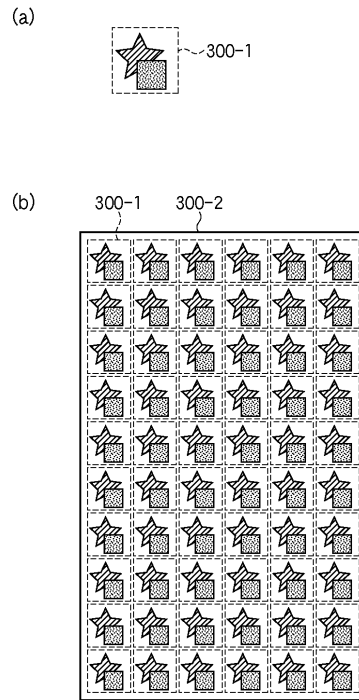
【図7】



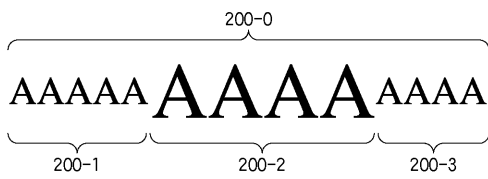
【 図 8 】



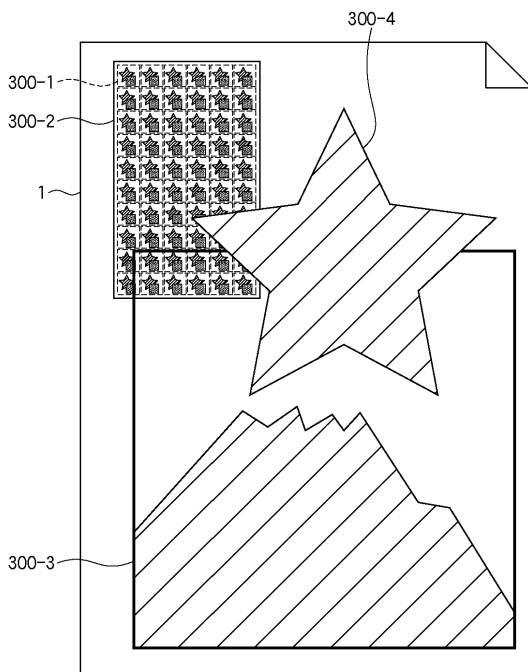
【 図 9 】



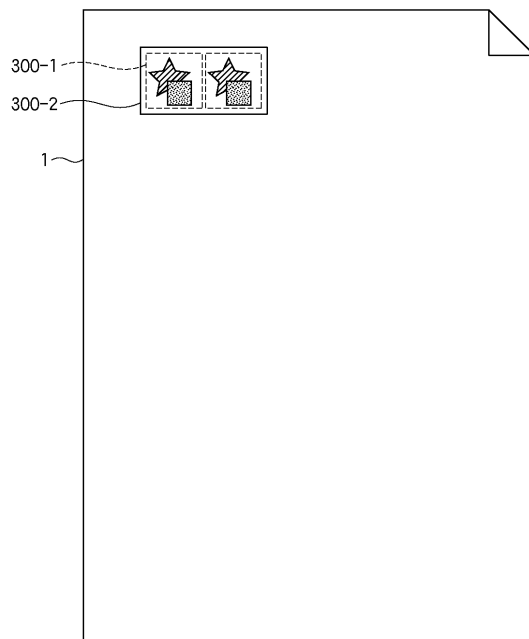
【 図 1 0 】



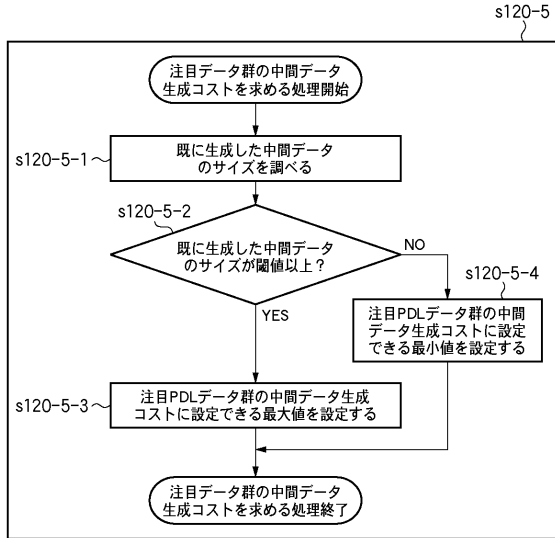
【 図 1 1 】



【 図 1 2 】



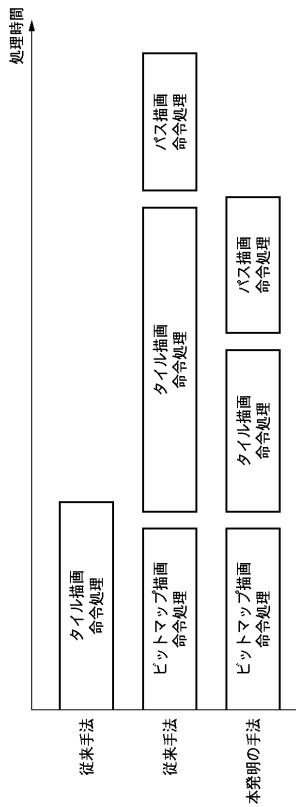
【図 13】



【図 14】

描画命令の種類	描画命令の種類を表す値	DL生成処理のコスト
圧縮データにかかる処理	1	4
シェーディング描画命令処理	2	10
タイル描画命令処理	3	3
バス描画命令処理	4	1
色変換処理	5	2
フォント描画命令処理	6	2
ビットマップ描画命令処理	7	5
エッジ描画命令処理	8	3
その他/不明	0	0

【図 15】



フロントページの続き

(72)発明者 武石 大樹
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2008-149543(JP,A)
特開2000-089622(JP,A)
特開2008-18577(JP,A)
特開2008-299480(JP,A)
特開2004-82709(JP,A)
特開平10-307924(JP,A)

(58)調査した分野(Int.Cl., DB名)
B41J 5/30
G06F 3/12
H04N 1/00