

(19) 日本国特許庁(JP)

(12) 公 開 特 許 公 報(A)

(11) 特許出願公開番号
特開2006-155018
(P2006-155018A)

(43) 公開日 平成18年6月15日(2006.6.15)

(51) Int.Cl.			F I			テーマコード (参考)
GO6T	11/00	(2006.01)	GO6T	11/00	100A	2C187
B41J	5/30	(2006.01)	B41J	5/30	Z	5B021
GO6F	3/12	(2006.01)	GO6F	3/12	B	5B050
GO6T	1/00	(2006.01)	GO6T	1/00	A	5B080

審査請求 未請求 請求項の数 4 O L (全 9 頁)

(21) 出願番号	特願2004-342066 (P2004-342066)	(71) 出願人	000001007
(22) 出願日	平成16年11月26日 (2004.11.26)		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	100090538
			弁理士 西山 恵三
		(74) 代理人	100096965
			弁理士 内尾 裕一
		(72) 発明者	松井 賢太
			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
		Fターム(参考)	2C187 BF09 BF14 BG02 BG05 BG06 BG14 BH30 5B021 CC05 DD06 LB07 5B050 BA07 BA18 BA20 5B080 FA02 FA05 FA09 GA00

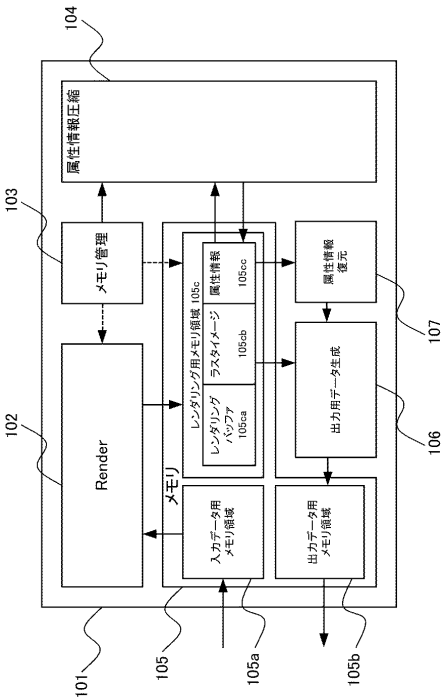
(54) 【発明の名称】 画像形成装置

(57) 【要約】

【課題】 レンダリング後のラスターイメージには、後段の画像処理のために文字、図形、イメージといった属性情報が付加される。複雑なデータをレンダリングする際、レンダリング用のメモリ領域を確保する為に、この属性情報を削減しても画質が劣化しない方法で、属性情報を削減する方法を提供する。

【解決手段】 入力データのフォーマットに従ってレンダリング後の画質に最も影響の少ない属性情報を削減する方法、又は出力データのフォーマットに従ってレンダリング後の画質に最も影響の少ない属性情報を削減する方法を採用する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

描画コマンドを元にラスターイメージを生成する ラスターイメージ生成手段と、
ラスターイメージに対して属性情報を付加する属性情報付加手段と、
ラスターイメージの生成時にメモリ不足を検知する メモリ不足検知手段と、
メモリ不足を検知した際には属性情報を削減する属性情報削減手段を備え、
属性情報を削減する際に、削減する属性情報を切り替えることを特徴とする画像形成装置。

【請求項 2】

前記属性情報削減手段は、入力データの形式に応じて削減する属性情報を切り替えることを特徴とする請求項 1 の画像形成装置。 10

【請求項 3】

前記属性情報削減手段は、出力データの形式に応じて削減する属性情報を切り替えることを特徴とする請求項 1 の画像形成装置。

【請求項 4】

入力データのデータ構造を解釈する入力データ解釈手段と、
入力データ解釈手段によって解釈されたデータから入力データの特徴を判断するデータ構造判断手段とを備え、

データ構造判断手段の判断によって、削減する属性情報を切り替えることを特徴とする請求項 1 の画像形成装置。 20

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、描画コマンドに従って生成されたラスターデータに対して付加される属性情報を削減する手段を備える画像形成装置に関する。

【背景技術】**【0002】**

レンダリング後のラスターイメージに対して、画像処理を行なう際、属性(文字、図形、イメージ)毎に切り替えたいという要望がある。この要望に答えるためにラスターイメージの各ピクセルに対して属性情報を割り当てている。しかし、描画処理が複雑なデータの場合、メモリの消費量が上がり、レンダリング用のメモリが不足するため、レンダリング用のメモリ領域を確保するために属性情報を削除し、全面に同じ属性値を設定する方式がとられていた。 30

【0003】

又、別の従来例としては、図形の描画処理をより高速に行うことが可能な画像処理装置(特許文献1)をあげることが出来る。

【特許文献1】特開2001-160148号公報**【発明の開示】****【発明が解決しようとする課題】****【0004】**

レンダリング後のラスターイメージには、後段の画処理のために文字、図形、イメージといった属性情報が付加される。しかし、メモリ消費量の大きい複雑なデータのレンダリングの場合、レンダリング用のメモリ領域を確保する為に、この属性情報を削減する必要がある。属性情報の削減の仕方によっては画質が著しく劣化する可能性があるため、画質劣化を抑えつつ属性情報を削減する方法が求められている。 40

【課題を解決するための手段】**【0005】**

描画コマンドを元にラスターイメージを生成する ラスターイメージ生成手段と、ラスターイメージに対して属性情報を付加する属性情報付加手段と、ラスターイメージの生成時にメモリ不足を検知する メモリ不足検知手段と、メモリ不足を検知した際には属性情 50

報を削減する属性情報削減手段を備え、属性情報削減の際に、削減する属性情報を切り替えることを特徴とする画像形成装置。

【0006】

前記属性情報削減手段は、入力データの形式によって削減する属性情報を切り替えることを特徴とする画像形成装置。

【0007】

前記属性情報削減手段は、出力データの形式に応じて削減する属性情報を切り替えることを特徴とする画像形成装置。

【0008】

さらに入力データのデータ構造を解釈する入力データ解釈手段と、入力データ解釈手段によって解釈されたデータから入力データの特徴を判断するデータ構造判断手段とを備え、データ構造判断手段の判断によって、削減する属性情報を切り替えることを特徴とする画像形成装置。 10

【発明の効果】

【0009】

実施例1に示したように本発明によって、レンダリング時、レンダリング用のメモリを確保する為に属性情報を削除する場合でも、入力データのフォーマットに従ってレンダリング後の画質に最も影響の少ない属性情報を削減するので、画質劣化の少ない出力結果を得ることが可能になる。

【0010】

実施例2に示したように本発明によって、レンダリング時にレンダリング用のメモリを確保する為に属性情報を削除する場合でも、出力データのフォーマットに従ってレンダリング後の画質に最も影響の少ない属性情報を削減するので、画質劣化の少ない出力結果を得ることが可能になる。 20

【0011】

実施例3に示したように本発明によって、レンダリング時、レンダリング用のメモリを確保する為に属性情報を削除する場合でも、レンダリング後の画質に最も影響の少ない属性情報を削減するので、画質劣化の少ない出力結果を得ることができる。

【発明を実施するための最良の形態】

【0012】

(実施例1) 30

図1は実施例1、2、3の画像形成装置の内部構造を示す図である。

【0013】

図において、101は、描画コマンドからラスタイメージを生成するレンダラ102と、メモリの管理を行うメモリ管理部103と、属性情報の圧縮を行う属性情報圧縮部104と、メモリ105と、出力データを生成する出力データ生成部106と、圧縮された属性情報を復元する属性情報復元部107からなる画像形成装置である。また、メモリ105は、描画コマンドを受け取る入力データ用メモリ領域105aと、出力データ用メモリ領域105bと、レンダリング用メモリ領域105cで構成され、レンダリング用メモリ領域105cはレンダリング用のバッファ105caと、レンダリング時に生成されるラスタイメージを格納する領域105cbと、レンダリング時に付加される属性情報を格納する領域105ccで構成される。 40

【0014】

図10は実施例1、2、3で使用する属性情報である。

【0015】

本実施例では1ピクセルにつき4ビットを割り当て、各ビットのON/OFFでそのピクセルの属性値を表す。

【0016】

すなわち、文字属性がONのピクセルは、そのピクセルが文字の一部であることを示し、細線属性がONのピクセルは、そのピクセルが細い線の一部であることを示し、グレイ 50

補償属性がONのピクセルは、そのピクセルが純粋な黒色であることを示し、イメージ属性がONのピクセルは、そのピクセルがイメージの一部であることを示す。

【0017】

本実施例における入力から出力までのデータの流れを図3に示す。

【0018】

ホスト(301)よりデータが入力される。

【0019】

本実施例では、入力データのフォーマットとしてRGBフォーマット(305a)、CMYKフォーマット(305b)、EXIFフォーマット(305c)を想定する。

【0020】

EXIFフォーマットは、標準化されたデジタルカメラ用の画像ファイルの規格で、各社のデジタルカメラに採用されている。TIFF形式で画像についての情報や撮影日時などの付加情報を記録できる。また画像形式はRGB無圧縮方式やJPEG方式など複数の形式をサポートしている。

【0021】

ホスト(301)より入力された入力データは、描画コマンドで構成されたPDLに変換される。

【0022】

PDL(302)はその描画コマンドを解釈して、レンダリング可能なデータを生成する。

【0023】

レンダリング(303)で、実際にレンダリング処理が行なわれ、ラスターイメージが生成されるが、その際に属性情報として文字(306a)、細線(306b)、グレイ補償(306c)、イメージ(305d)が付加される。通常は全属性情報が付加させるが、レンダリング処理時のメモリ使用状況によっては、いずれかの属性情報を削減する必要がある。

【0024】

本実施例における、出力データを生成するまでの手順を図2のフローチャートを用いて説明する。S201からS207は各ステップを示す。

【0025】

ステップS201において、入力データのフォーマットをチェックする。

【0026】

ステップS202において、描画コマンドを受け取り、ステップS203へ進む。

【0027】

ステップS203において、レンダリング処理を行い、且つ、属性情報を付加し、ステップS204へ進む。

【0028】

ステップS204において、描画コマンドの最後であるか確認し、描画コマンドの最後でない場合はステップS205へ進む。

【0029】

ステップS205において、次のコマンドを実行するためのメモリ容量が残っているかどうかを確認し、残っていればステップS203へ進みこれを繰り返す。次のコマンドを実行するメモリ容量が残っていなければ、ステップS206へ進む。

【0030】

ステップS206において、S201でチェックした入力データのフォーマットに従い、削除する属性情報を決定する。

【0031】

本実施例では、予め入力データフォーマット毎に削除する属性情報を設定しておく。図4は入力データのフォーマットと削除する属性情報の対応させた表である。

【0032】

10

20

30

40

50

例えば入力データフォーマットが E X I F の場合、主にはイメージ画像が送られてくるので、最も必要ないと考えられる「細線」の属性が削除対象になっている。

【 0 0 3 3 】

ステップ S 2 0 7 において、ステップ S 2 0 6 で選択した属性情報を実際に削減する（実施例 2）

本実施例における入力から出力までのデータの流れを図 6 に示す。

【 0 0 3 4 】

ホスト（ 6 0 1 ）より入力されるデータには、ホストのドライバによって出力すべきフォーマットが指定されている。本実施例では、その出力フォーマットとして、カラー（ 6 0 5 a ）、モノクロ（ 6 0 5 b ）、文書（ 6 0 5 c ）、表（ 6 0 5 d ）を想定する。

10

【 0 0 3 5 】

ホスト（ 6 0 1 ）より入力された入力データは、描画コマンドで構成された P D L に変換される。

【 0 0 3 6 】

P D L （ 6 0 2 ）はその描画コマンドを解釈して、レンダリング可能なデータを生成する。

【 0 0 3 7 】

レンダリング（ 6 0 3 ）で、実際にレンダリング処理が行なわれ、ラスターイメージが生成されるが、その際に属性情報として文字（ 6 0 6 a ）、細線（ 6 0 6 b ）、グレイ補償（ 6 0 6 c ）、イメージ（ 6 0 6 d ）が付加される。通常は全属性情報が付加させるが、レンダリング処理時のメモリ使用状況によっては、いずれかの属性情報を削減する必要がある。

20

【 0 0 3 8 】

本実施例における、出力データを生成するまでの手順を図 5 のフローチャートを用いて説明する。S 5 0 1 から S 5 0 7 は各ステップを示す。

【 0 0 3 9 】

ステップ S 5 0 1 において、出力データのフォーマットをチェックする。

【 0 0 4 0 】

ステップ S 5 0 2 において、描画コマンドを受け取り、ステップ S 5 0 3 へ進む。

【 0 0 4 1 】

ステップ S 5 0 3 において、レンダリング処理を行い、且つ、属性情報を付加し、ステップ S 5 0 4 へ進む。

30

【 0 0 4 2 】

ステップ S 5 0 4 において、描画コマンドの最後であるか確認し、描画コマンドの最後でない場合はステップ S 5 0 5 へ進む。

【 0 0 4 3 】

ステップ S 5 0 5 において、次のコマンドを実行するためのメモリ容量が残っているかどうかを確認し、残っていればステップ S 5 0 3 へ進みこれを繰り返す。次のコマンドを実行するメモリ容量が残っていなければ、ステップ S 5 0 6 へ進む。

【 0 0 4 4 】

ステップ S 5 0 6 において、S 5 0 1 でチェックした出力データのフォーマットに従い、削除する属性情報を決定する。

40

【 0 0 4 5 】

本実施例では、予め出力データフォーマット毎に削除する属性情報を設定しておく。図 7 は出力データのフォーマットと削除する属性情報の対応させた表である。

【 0 0 4 6 】

例えば入力データフォーマットが表の場合、あまり重要視されないと考えられる「イメージ」の属性が削除対象になっている。

【 0 0 4 7 】

ステップ S 5 0 7 において、ステップ S 5 0 6 で選択した属性情報を実際に削減する。

50

【 0 0 4 8 】

(実施例 3)

本実施例における入力から出力までのデータの流れを図 9 に示す。

【 0 0 4 9 】

ホスト (9 0 1) よりデータが入力される。

【 0 0 5 0 】

ホスト (9 0 1) より入力された入力データは、描画コマンドで構成された P D L に変換される。

【 0 0 5 1 】

P D L (9 0 2) はその描画コマンドを解釈して、レンダリング可能なデータを生成するが、描画コマンドを分析することで入力データの特徴を予め把握する (9 0 3)。 10

【 0 0 5 2 】

レンダリング (9 0 4) で、実際にレンダリング処理が行なわれ、ラスターイメージが生成されるが、その際に属性情報として文字 (9 0 6 a)、細線 (9 0 6 b)、グレイ補償 (9 0 6 c)、イメージ (9 0 6 d) が付加される。通常は全属性情報が付加させるが、レンダリング処理時のメモリ使用状況によっては、いずれかの属性情報を削減する必要がある。

【 0 0 5 3 】

本実施例における、出力データを生成するまでの手順を図 8 のフローチャートを用いて説明する。S 8 0 1 から S 8 0 6 は各ステップを示す。 20

【 0 0 5 4 】

ステップ S 8 0 1 において、描画コマンドを受け取り、ステップ S 8 0 2 へ進む。

【 0 0 5 5 】

ステップ S 8 0 2 において、描画するページ内に存在する描画オブジェクトの面積比をオブジェクト属性毎に計測する。

【 0 0 5 6 】

ステップ S 8 0 3 において、レンダリング処理を行い、且つ、属性情報を付加し、ステップ S 5 0 4 へ進む。

【 0 0 5 7 】

ステップ S 8 0 4 において、描画コマンドの最後であるか確認し、描画コマンドの最後でない場合はステップ S 8 0 5 へ進む。 30

【 0 0 5 8 】

ステップ S 8 0 5 において、次のコマンドを実行するためのメモリ容量が残っているかどうかを確認し、残っていればステップ S 8 0 3 へ進みこれを繰り返す。次のコマンドを実行するメモリ容量が残っていなければ、ステップ 8 0 6 へ進む。

【 0 0 5 9 】

ステップ S 8 0 6 において、ステップ S 8 0 2 で計測した属性別の描画オブジェクトの面積比を参照し、最も面積比が少ない属性情報を実際に削減する。

【 図面の簡単な説明 】

【 0 0 6 0 】

【 図 1 】 実施例 1、2、3 の画像形成装置の内部構造を示す図。

【 図 2 】 実施例 1 におけるデータの処理手順を示すフローチャート。

【 図 3 】 実施例 1 におけるデータの流れを示す図。

【 図 4 】 入力データフォーマットと削除する属性情報の対応表。

【 図 5 】 実施例 2 におけるデータの処理手順を示すフローチャート。

【 図 6 】 実施例 2 におけるデータの流れを示す図。

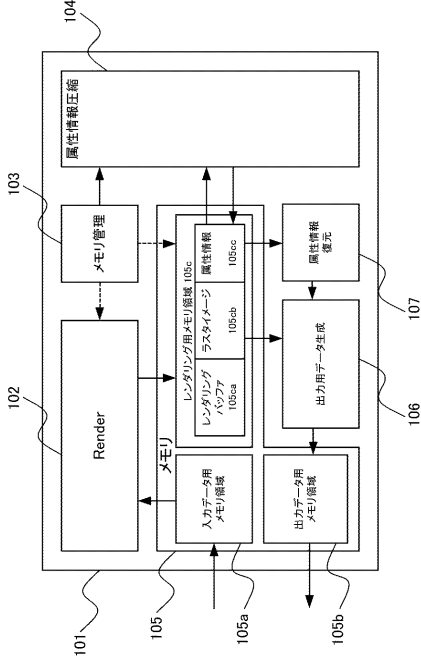
【 図 7 】 出力データフォーマットと削除する属性情報の対応表。

【 図 8 】 実施例 3 におけるデータの処理手順を示すフローチャート。

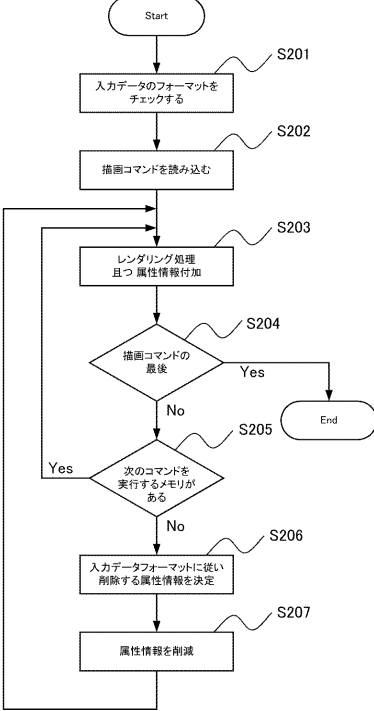
【 図 9 】 実施例 3 におけるデータの流れを示す図。

【 図 10 】 実施例 1、2、3 で使用する属性情報。

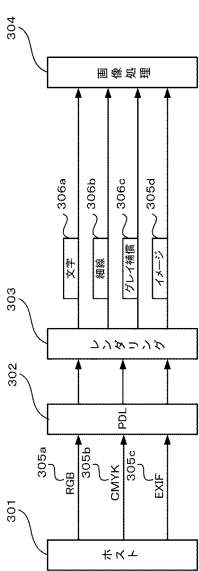
【図 1】



【図 2】



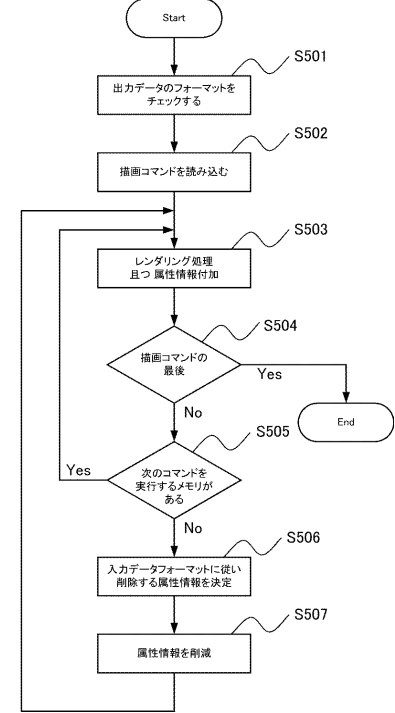
【図 3】



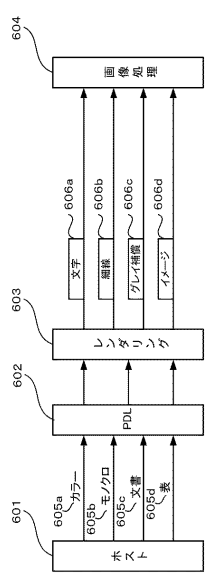
【図 4】

入力データ フォーマット	削除する属性情報
RGB	細線
CMYK	グレイ補償
EXIF	細線

【 図 5 】



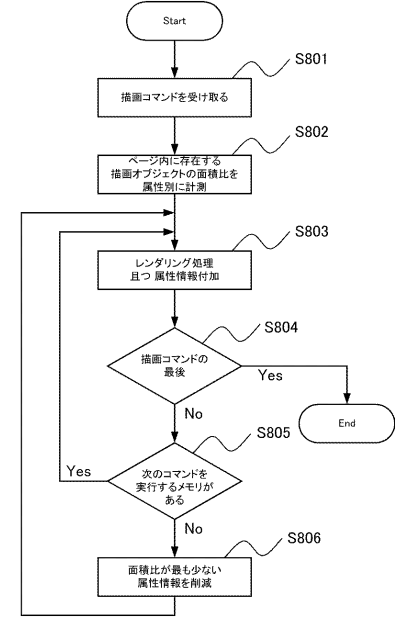
【 図 6 】



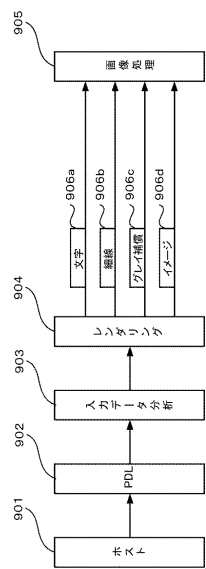
【 図 7 】

出力データ フォーマット	削除する属性情報
カラー	細線
モノクロ	グレイ補償
文書	細線
表	イメージ

【 図 8 】



【図 9】



【図 10】

ピクセルの属性情報

文字	罫線	グレイ補償	イメージ
----	----	-------	------

4bit/pixel