

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7653808号
(P7653808)

(45)発行日 令和7年3月31日(2025.3.31)

(24)登録日 令和7年3月21日(2025.3.21)

(51)国際特許分類

F I

H 0 4 N	1/32 (2006.01)	H 0 4 N	1/32	1 4 4
H 0 4 N	1/409(2006.01)	H 0 4 N	1/409	
H 0 4 N	1/401(2006.01)	H 0 4 N	1/401	
H 0 4 N	1/407(2006.01)	H 0 4 N	1/407	

請求項の数 16 (全28頁)

(21)出願番号	特願2021-39527(P2021-39527)
(22)出願日	令和3年3月11日(2021.3.11)
(65)公開番号	特開2022-139237(P2022-139237)
	A)
(43)公開日	令和4年9月26日(2022.9.26)
審査請求日	令和6年3月8日(2024.3.8)

(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	110003281 弁理士法人大塚国際特許事務所
(72)発明者	山田 順季 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
(72)発明者	矢野 健太郎 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	松永 隆志

最終頁に続く

(54)【発明の名称】 情報処理装置とプログラム及び画像処理方法

(57)【特許請求の範囲】**【請求項1】**

画像データに基づいて画像を形成する画像形成手段と、画像を画像データとして読み取る読み取り手段とに接続された情報処理装置であって、

ページ記述言語で記述された原稿データに基づいて画像データを生成する生成手段と、前記原稿データを符号化して前記画像データに多重化する多重化手段と、

前記画像形成手段に前記画像データを出力させる出力手段と、

前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元する復元手段と、

前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存する保存手段と、を有し、

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、

除外された前記オブジェクトは前記画像形成手段により印刷されることを特徴とする情報処理装置。

【請求項2】

画像データに基づいて画像を形成する画像形成手段と接続された情報処理装置であって、ページ記述言語で記述された原稿データに基づいて画像データを生成する生成手段と、

前記原稿データを符号化して前記画像データに多重化する多重化手段と、

前記画像形成手段に前記画像データを出力させる出力手段と、を有し

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、
除外された前記オブジェクトは前記画像形成手段により印刷されることを特徴とする情報処理装置。

【請求項 3】

ページ記述言語で記述された原稿データが多重化された画像を画像データとして読み取る読み取り手段と接続された情報処理装置であって、
前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元する復元手段と、

前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存する保存手段と、を有し、

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、
除外された前記オブジェクトは画像形成手段により印刷されることを特徴とする情報処理装置。

【請求項 4】

画像データに基づいて画像を形成する画像形成手段と、画像を画像データとして読み取る読み取り手段とに接続された情報処理装置であって、
ページ記述言語で記述された原稿データに基づいて画像データを生成する生成手段と、

前記原稿データを符号化して前記画像データに多重化する多重化手段と、
前記画像形成手段に前記画像データを出力させる出力手段と、

前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元する復元手段と、

前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存する保存手段と、を有し、

前記原稿データから生成された前記画像データを複数の領域に分割した領域それぞれに含まれたオブジェクトのデータ量が所定値を超えている領域については、当該領域に含まれたオブジェクトを除いた原稿データが、前記画像データに多重化されることを特徴とする情報処理装置。

【請求項 5】

請求項 4 に記載の情報処理装置であって、
前記オブジェクトを除いた原稿データには、当該オブジェクトの領域を含む描画命令が含まれることを特徴とする情報処理装置。

【請求項 6】

請求項 1 または請求項 3 乃至 5 のいずれか一項に記載の情報処理装置であって、
前記保存手段により保存された前記画像データと前記原稿データとにに基づいて画像を再生する再生手段を更に有することを特徴とする情報処理装置。

【請求項 7】

請求項 6 に記載の情報処理装置であって、
前記再生手段は、保存された前記原稿データに含まれていないオブジェクトがある場合には、当該オブジェクトに対応する、前記保存手段に保存された前記画像データにより前記オブジェクトを再生することを特徴とする情報処理装置。

【請求項 8】

請求項 1 乃至 7 のいずれか一項に記載の情報処理装置であって、
前記原稿データは、符号化されて前記画像の高周波成分に多重化されることを特徴とする情報処理装置。

【請求項 9】

10

20

30

40

50

請求項 1 乃至 7 のいずれか一項に記載の情報処理装置であって、前記原稿データは、マトリクス型二次元コードに符号化され、前記マトリクス型二次元コードに相当する所定の色成分の画素を置換することで前記画像に多重化されることを特徴とする情報処理装置。

【請求項 1 0】

請求項 1 乃至 9 のいずれか一項に記載の情報処理装置であって、画像形成手段を更に有することを特徴とする情報処理装置。

【請求項 1 1】

請求項 1 乃至 10 のいずれか一項に記載の情報処理装置であって、読み取り手段を更に有することを特徴とする情報処理装置。

【請求項 1 2】

請求項 1 乃至 9 のいずれか一項に記載の情報処理装置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。

10

【請求項 1 3】

画像データに基づいて画像を形成する画像形成手段と、画像を画像データとして読み取る読み取り手段とに接続された、生成手段と多重化手段と出力手段と復元手段と保存手段とを有する情報処理装置により実行される画像処理方法であって、

前記生成手段が、ページ記述言語で記述された原稿データに基づいて画像データを生成し、

20

前記多重化手段が、前記原稿データを符号化して前記画像データに多重化し、

前記出力手段が、前記画像形成手段に前記画像データを出力させ、

前記復元手段が、前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元し、

前記保存手段が、前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存し、

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、

除外された前記オブジェクトは前記画像形成手段により印刷されることを特徴とする画像処理方法。

30

【請求項 1 4】

画像データに基づいて画像を形成する画像形成手段と接続された、生成手段と多重化手段と出力手段とを有する情報処理装置により実行される画像処理方法であって、

前記生成手段が、ページ記述言語で記述された原稿データに基づいて画像データを生成し、

前記多重化手段が、前記原稿データを符号化して前記画像データに多重化し、

前記出力手段が、前記画像形成手段に前記画像データを出力させ、

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、

除外された前記オブジェクトは前記画像形成手段により印刷されることを特徴とする画像処理方法。

40

【請求項 1 5】

ページ記述言語で記述された原稿データが多重化された画像を画像データとして読み取る読み取り手段と接続された、復元手段と保存手段とを有する情報処理装置により実行される画像処理方法であって、

前記復元手段が、前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元し、

前記保存手段が、前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存し、

50

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、除外された前記オブジェクトは画像形成手段により印刷されることを特徴とする画像処理方法。

【請求項 16】

画像データに基づいて画像を形成する画像形成手段と、画像を画像データとして読み取る読み取り手段とに接続された、生成手段と多重化手段と出力手段と復元手段と保存手段とを有する情報処理装置により実行される画像処理方法であって、

前記生成手段が、ページ記述言語で記述された原稿データに基づいて画像データを生成し、

10

前記多重化手段が、前記原稿データを符号化して前記画像データに多重化し、

前記出力手段が、前記画像形成手段に前記画像データを出力させ、

前記復元手段が、前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元し、

前記保存手段が、前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存し、

前記原稿データから生成された前記画像データを複数の領域に分割した領域それぞれに含まれたオブジェクトのデータ量が所定値を超えている領域については、当該領域に含まれたオブジェクトを除いた原稿データが、前記画像データに多重化されることを特徴とする画像処理方法。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、例えば情報処理装置とプログラム及び画像処理方法に関する。

【背景技術】

【0002】

従来、デジタル原稿をプリンタによって印刷した出力画像をスキャナで読み取り、読み取るデジタル原稿として保存するスキャン機能は知られている。その際、印刷工程及びスキャン工程を経て生成された読み取るデジタル原稿は元のデジタル原稿から色・形状いずれも劣化する事が知られている。

30

【0003】

そこで、特許文献1では、デジタル原稿中の色情報にプリンタの出力特性を加味して出力画像色を推定し、その推定色を印刷画像中に多重化情報として埋め込み、スキャン画像から多重化情報を抽出して色の劣化を復元する例が記載されている。

【先行技術文献】

【特許文献】

【0004】

【文献】特許第5180551号公報

【文献】特許第4187749号公報

【文献】特開2008-186165公報

40

【発明の概要】

【発明が解決しようとする課題】

【0005】

しかしながら、特許文献1では、印刷工程及びスキャン工程、また印刷原稿への追記や汚れ等によって生じた色の劣化を、劣化前の状態に近づく様復元する事は出来るが、形状復元の精度には懸念が残る。

本発明は上記従来例に鑑みて成されたもので、媒体への画像形成に起因して生じた読み取るデジタル原稿における画像の劣化を抑制することを目的とする。

【課題を解決するための手段】

【0006】

50

上記目的を達成するために本発明は以下の構成を有する。すなわち、本発明の一側面によれば、画像データに基づいて画像を形成する画像形成手段と、画像を画像データとして読み取る読み取り手段とに接続された情報処理装置であって、

ページ記述言語で記述された原稿データに基づいて画像データを生成する生成手段と、前記原稿データを符号化して前記画像データに多重化する多重化手段と、前記画像形成手段に前記画像データを出力させる出力手段と、前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元する復元手段と、

前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存する保存手段と、を有し、

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、

除外された前記オブジェクトは前記画像形成手段により印刷されることを特徴とする情報処理装置が提供される。

本発明の他の側面によれば、画像データに基づいて画像を形成する画像形成手段と接続された情報処理装置であって、

ページ記述言語で記述された原稿データに基づいて画像データを生成する生成手段と、前記原稿データを符号化して前記画像データに多重化する多重化手段と、

前記画像形成手段に前記画像データを出力させる出力手段と、を有し

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、

除外された前記オブジェクトは前記画像形成手段により印刷されることを特徴とする情報処理装置が提供される。

【0007】

また本発明の他の側面によれば、ページ記述言語で記述された原稿データが多重化された画像を画像データとして読み取る読み取り手段と接続された情報処理装置であって、前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元する復元手段と、

前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存する保存手段と、を有し、

前記原稿データに含まれるオブジェクトが画像である場合、前記オブジェクトを除外することで得られた前記原稿データは前記画像データに多重化され、

除外された前記オブジェクトは画像形成手段により印刷されることを特徴とする情報処理装置が提供される。

本発明のさらなる側面によれば、画像データに基づいて画像を形成する画像形成手段と、画像を画像データとして読み取る読み取り手段とに接続された情報処理装置であって、

ページ記述言語で記述された原稿データに基づいて画像データを生成する生成手段と、前記原稿データを符号化して前記画像データに多重化する多重化手段と、

前記画像形成手段に前記画像データを出力させる出力手段と、

前記読み取り手段により読み取られた画像データから多重化された前記原稿データを復元する復元手段と、

前記読み取り手段により読み取られた前記画像データと前記復元手段により復元された前記原稿データとを保存する保存手段と、を有し、

前記原稿データから生成された前記画像データを複数の領域に分割した領域それぞれに含まれたオブジェクトのデータ量が所定値を超えている領域については、当該領域に含まれたオブジェクトを除いた原稿データが、前記画像データに多重化されることを特徴とする情報処理装置が提供される。

【発明の効果】

【0009】

本発明によれば、媒体への画像形成に起因して生じた読み取りデジタル原稿における画像の

10

20

30

40

50

劣化を抑制することが可能となる。

【図面の簡単な説明】

【0010】

【図1】システム構成図

【図2】第一の実施形態におけるPDL多重化印刷及び多重化PDL抽出フローを説明する図

【図3】PDL原稿の例を説明する図

【図4】従来印刷及び読み取りによる画像弊害例を説明する図

【図5】PDL多重化で画像変動を生成するマスクを説明する図

【図6】PDL多重化で形成されるパターンを説明する図

【図7】埋め込みパターンの空間周波数特性の一例を示す図

【図8】多重化の他の例を示す図

【図9】第二の実施形態におけるPDL多重化印刷フローを説明する図

【図10】第二の実施形態における多重化用PDL原稿例を説明する図

【図11】第二の実施形態における多重化用PDL生成フローを説明する図

【図12】第三の実施形態におけるタイル分割適応的PDL多重化印刷フローを説明する図

【図13】第三の実施形態におけるタイル分割PDL原稿例を説明する図

【図14】第三の実施形態におけるタイル分割多重化用PDL原稿例を説明する図

【図15】第三の実施形態における多重化用分割PDL生成フローを説明する図

【発明を実施するための形態】

【0011】

以下、添付図面を参照して実施形態を詳しく説明する。なお、以下の実施形態は特許請求の範囲に係る発明を限定するものではない。実施形態には複数の特徴が記載されているが、これらの複数の特徴の全てが発明に必須のものとは限らず、また、複数の特徴は任意に組み合わせられてもよい。さらに、添付図面においては、同一若しくは同様の構成に同一の参照番号を付し、重複した説明は省略する。

(第一の実施形態)

図1は、本発明の記録システムの構成を示すブロック図である。同図に示すように、この印刷装置は、MFP(多機能複写機あるいはデジタル複合機：Multi Function Printer)装置40と、そのホスト装置としてのパーソナルコンピュータ(PC)50を有して構成されるものである。MFP本体とは、本体にプリンタ機能及びスキャナ機能という複数の機能を持つプリンタの事を指し、両機能を連携させて行うコピー機能も具備して居るケースが多い。

【0012】

ホストPC50は、主に以下の要素を有して構成されるものである。CPU501は、HDD503やRAM502に保持されているプログラムに従った処理を実行する。RAM502は、揮発性のストレージであり、プログラムやデータを一時的に保持する。また、HDD503は、不揮発性のストレージであり、同じくプログラムやデータを保持する。DATA TRANSFER I/F(データ転送インターフェース)504はMFP本体40との間におけるデータの送受信を制御する。このデータ送受信の接続方式としては、USB、IEEE1394、LAN等の有線接続や、Bluetooth、Wi-Fi等の無線接続を用いることができる。KEY BOARD MOUSE I/F(キーボード・マウスインターフェース)505は、キーボードやマウス等のHID(Human Interface Device)を制御するI/Fであり、ユーザは、このI/Fを介して入力をすることができる。DISPLAY I/F(ディスプレイインターフェース)506は、ディスプレイ(不図示)における表示を制御する。

【0013】

一方、MFP本体40は、主に以下の要素を有して構成されるものである。CPU401は、ROM403やRAM402に保持されているプログラムに従い、図3以降で後述する各実施形態の処理を実行する。RAM402は、揮発性のストレージであり、プログラムやデータを一時的に保持する。また、ROM403は不揮発性のストレージであり、

10

20

30

40

50

図3以降で後述する各実施形態の処理で使用されるテーブルデータやプログラムを保持することができる。

【0014】

DATA TRANSFER I/F(データ転送インターフェース)404はPC50との間におけるデータの送受信を制御する。PRINTER Controller(プリンタコントローラ)405はプリントヘッドを含むプリンタ408を制御する。PRINTER Controller 405はHEAD Controller(ヘッドコントローラ)を含む。ヘッドコントローラは、プリンタユニット408の有する印刷ヘッドに対して記録データに基づいてヘッドに搭載されたヒータの加熱動作を制御し、インクを吐出する。これによって媒体上へと画像形成を行う。具体的には、HEAD Controllerは、RAM402の所定のアドレスから制御パラメータと記録データを読み込む構成とすることができます。そして、CPU401が、制御パラメータと記録データをRAM402の上記所定のアドレスに書き込むと、プリンタコントローラ405、特にHEAD Controllerにより処理が起動され、記録ヘッドに搭載されたヒータの加熱動作が行われる。10

【0015】

Image Processing Accelerator(画像処理アクセラレータ)406は、ハードウェアによって構成され、CPU401よりも高速に画像処理を実行するものである。具体的には、Image Processing Accelerator 406は、RAM402の所定のアドレスから画像処理に必要なパラメータとデータを読み込む構成とすることができます。そして、CPU401が上記パラメータとデータをRAM402の上記所定のアドレスに書き込むと、Image Processing Accelerator 406が起動され、所定の画像処理が行われる。なお、Image Processing Accelerator 406は必ずしも必要な要素でなく、プリンタの仕様などに応じて、CPU401による処理のみで上記のテーブルパラメータの作成処理および画像処理を実行してもよいことはもちろんである。20

SCANNER Controller(スキャナコントローラ)407は、スキャナユニット409に対して、原稿への光の照射と反射光をCCD等の撮像素子で取得した光量情報のSCANNER Controller 407への送信を指示する。具体的には、CPU401が、制御パラメータと読み取りデータ書き出しアドレスをRAM402の上記所定のアドレスに書き込むと、SCANNER Controller 407により処理が起動される。そして、スキャナユニット409に搭載されたLEDの発行制御と、スキャナユニット409からの光量情報取得、RAM402中の読み取りデータ書き出しアドレス以降への光量情報書き込みが行われる。30

【0016】

PRINTER Controller 405はさらにMOTOR Controller(モーターコントローラ)を有する。MOTOR Controllerは、プリンタユニット408に含まれる不図示の複数のモーターユニットに対して、モーター動作を制御する。モーターは上述の印刷ヘッドを記録用紙に対して相対的に移動させる為と、スキャナユニット409を読み取る原稿に対して相対的に移動させる為に用いられる。その他にも、MFPによっては記録ヘッドのメンテナンス用のモーターを具備する場合がある。40

【0017】

色別の画素数情報の多重化処理

図2は、第一の実施形態におけるPDL多重化印刷フローと多重化PDL抽出フローを説明する図である。図2(a)のステップs201~s205がPDLを多重化して印刷するフロー(簡単に多重化印刷処理とも呼ぶ)、図2(b)のステップs211~s214が多重化したPDLを抽出するフロー(簡単に抽出処理とも呼ぶ)である。いずれの処理もMFP本体40のCPU401によりRAM402等のメモリに格納されたプログラムを実行することで実現される。しかしながら、図2(a)のs1106と図2(b)のs1111それぞれの印刷と原稿読み取りをMFP本体40で実行し、そのほかの工程は50

ホストPC50で実行してもよい。これは他の実施形態についても同様である。まずは図2(a)のPDL多重化印刷フローについて説明する。

【0018】

ステップs201でPDL原稿データを取得する。ここで、PDLとはページ記述言語(Page Description Language)の略であり、ページ単位で描画命令の組から構成される。描画命令の種類はPDL仕様毎に定義されているが、本実施形態では、主に以下の4種類を例として用いる。かっこ内は命令のパラメータであり、X,Yは描画位置の座標を示す。

命令1) DOT描画命令(X,Y,色、点サイズ)

命令2) LINE描画命令(X1,Y1,X2,Y2,色、線種、線幅)

10

命令3) BOX描画命令(X1,Y1,X2,Y2,色、塗り形状)

命令4) TEXT描画命令(X1,Y1,色、フォント情報、文字列情報)

命令5) IMAGE描画命令(X1,Y1,X2,Y2,画像ファイル情報)

他にも、円弧を描くCIRCLE描画命令等、用途に応じて適宜描画命令が用いられる。

【0019】

一般的によく用いられるPDLとしては、Adobe社提案のPDF(Portable Document Format)や、Microsoft社提案のXPS、HP社提案のHP-GL/2等が存在するが、本発明の適用範囲はその限りでは無い。

【0020】

ページ記述言語の例

20

図3にPDL原稿の例を示す。図3中の300が原稿の1ページを表し、画素数としては、例えば横幅600画素、縦幅900画素とする。また、以下に図3の原稿に対応するPDLの例を示す。なおPDL原稿とはPDLで記述したデータであり、原稿データあるいは印刷データなどとも呼ぶ。

<PAGE=001>

```
<BOX> 50, 75, 200, 300, GRAY, STRIPE </BOX>
<BOX> 420, 75, 550, 300, BLACK, FULL </BOX>
<BOX> 50, 620, 200, 825, BLACK, NULL </BOX>
<TEXT> 50, 620, 200, 825, BLACK, STD-18, "ABCXYZ123" </BOX>
<LINE> 25, 25, 25, 700, BLACK, PLAIN, 3 </LINE>
<LINE> 25, 25, 450, 25, BLACK, PLAIN, 3 </LINE>
<IMAGE> 250, 350, 550, 825, "PORTRAIT.jpg" </IMAGE>
```

</PAGE>。

【0021】

1行目の<PAGE=001>は本実施形態におけるページ数を表すタグである。通常PDLは複数ページを記述可能に設計されているので、PDL中にページの区切りを示すタグが記述されている。本例では、9行目の</PAGE>タグまでが1ページ目である事を表す。本実施形態では図3中の原稿300に相当する。2ページ目が存在する場合には、上記PDLに続いて<PAGE=002>が記述される事となる。

【0022】

2行目の<BOX>タグから</BOX>タグまでがボックス描画命令であり、図3中のボックスオブジェクト301に相当する。最初の2座標が描画開始点である左上座標(X1,Y1)を示し、続く2座標が描画終了点である右下座標(X2,Y2)を示す。続いて色はGRAY(灰色:R=128, G=128, B=128)で、塗り形状は縞模様であるSTRIPE(縞模様)が指定されている。本実施形態では縞模様の向きについては常時右下方向への線としたが、線の角度や周期等をBOX命令中で指定出来る様になっていても良い。

【0023】

3行目のBOX描画命令は、図3中のボックスオブジェクト302に相当する。本命令の場合、2行目と異なり、色がBLACK(黒色:R=0, G=0, B=0)となっており、また塗り形状についてはベタ塗りであるFULL(ベタ)が指定されている。

40

50

【 0 0 2 4 】

4行目のBOX描画命令及び5行目のTEXT描画命令が、図3中のオブジェクト303に相当する。まず、BOX描画命令でテキスト回りの枠線を記述する。ここでは、BOX内部の塗り形状がNULL(空白)が指定されている。続いて5行目のTEXT描画命令においては、文字のフォントが"STD"（標準）である事、その文字サイズが18ポイントである事、そして記述すべき文字列が"ABCXYZ123"である事が記述されている。

【 0 0 2 5 】

続いて6行目及び7行目のLINE描画命令が、図3中のラインオブジェクト304に相当する。まず、6行目のLINE描画命令で縦線を描画する。ここでは、描画すべき線の種類がP LINE(実線)である事、及び、描画すべき線の太さが3画素線で有る事、が記述されている。同様に、7行目のLINE描画命令で横線を描画する。

10

【 0 0 2 6 】

続いて8行目のIMAGE描画命令が、図3中のイメージオブジェクト305に相当する。ここでは、当該領域に存在する画像のファイル名が"PORTRAIT.jpg"である旨が記載されており、これは、一般的に普及している画像圧縮フォーマットであるJPEGファイルである事を表す。

【 0 0 2 7 】

そして、9行目に記載の</PAGE>タグで当該ページの描画が終了した事を示す。

【 0 0 2 8 】

実際のPDLファイルとしては、上記に記載の描画命令群に加えて、"STD"フォントデータ、"PORTRAIT.jpg"画像ファイルを含めた物が一体となっているケースが多い。これは、フォントデータや画像ファイルを別にして管理する場合、描画命令だけでは文字・画像部分が形成出来ず、図3の画像を形成するのに情報が不十分だからである。

20

【 0 0 2 9 】

以上が図2中のステップs201で取得するPDLの説明である。

【 0 0 3 0 】

続いてステップs202に進み、ステップs201で取得したPDL原稿データのレンダリング処理を行う。これは、PDL中で記載されている各描画命令を実行し、画素毎の色情報で構成されるBitmap画像を形成するステップである。本実施形態では、上述した様に、図3に例示した画像が形成される。図3の原稿ページ300は横幅600画素、縦幅900画素から構成される為、本ステップで生成されるBitmap画像は 600×900 画素から構成される。また、各画素はR/G/B各8bitの256階調とする。

30

【 0 0 3 1 】

ステップs203では、多重化用画像を作成する。ここでは、ステップs201で取得したPDLファイルデータを、ステップs202で生成したレンダリング画像に重畠（あるいは多重化）する。これは、印刷画像をスキャンした際に画像データからPDLファイルデータを抽出可能とし、図4で説明した画像形状の劣化からの復元手段とする為である。なお多重化対象の画像データがカラー画像である場合には、特定の色成分を対象として多重化すればよい。本実施形態ではたとえばB（青）色成分を対象として、色別画素数情報（以下、画素数情報あるいは付加情報とも呼ぶ。）が多重化されるが、他の色成分を対象としてもよい。

40

【 0 0 3 2 】

PCのような情報処理装置で情報を扱うというのはバイナリデータを扱うことである。バイナリデータとは「0」もしくは「1」の情報であり、この「0」もしくは「1」の情報が連續でつながることにより、特定の意味を持つようになる。例えば、「hello」という情報をバイナリデータで扱う場合、文字コードの中の一つである「シフトJIS」を例にとると、「h」はバイナリデータの「01101000」に対応している。同様に「e」は「01100101」、「l」は「01101100」、「o」は「01101111」というバイナリデータに対応している。つまり「hello」という文字はバイナリデータで「0110100001100101101100011011000」

50

「1 1 0 1 1 1 1」と表現できる。逆に「0 1 1 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 0 1 1 0 0 1 1 0 1 1 1 1」というバイナリデータを取得できれば「h e l l o」という文字を取得することができる。この考えに基づけば多重化は、「0」もしくは「1」を判定できるようにデータを埋め込むことで実現可能であることがわかる。

【0 0 3 3】

ここで、「0」、「1」すなわち多重化される情報を生成するために、一例として図5(a) (b)の二つのマスクを考える。これは8ピクセル×8ピクセルで構成されたマスクであり、画像に対して対応するマスクの内容を加算することで、画像内の8ピクセル×8ピクセルの領域に対して周期性を持ったパターンを与えることができる。このパターンにより、0と1とを表現する。図5(a)が0、図5(b)が1を示すパターンである。基本的にデジタル画像は1色8bitで表現され0から255までのどれかの値が割り当てられる。この範囲外の値は画像データとしては使えないため、画素値の計算結果が0未満の場合や、256以上の値になった場合には一般的に0もしくは255を割り振り有効範囲内に収めている。図5のマスクでは画素値に±10の変化を与えているが、マスク領域における画像データの値が全て255であった場合、マスク後のその領域の値は245から265の範囲ではなく245から255の範囲となる。本説明は1色8bitと記載しているが、当然8bit以外もありうる。デジタル画像を扱う場合、何bitで表現しようが有効範囲は存在し、その範囲外になる変化は与えられない。図5に示したように、0と1というに対応するマスクによりPDL原稿画像をマスキング処理することで、8×8画素領域に1ビットの情報を多重化することができる。

10

20

【0 0 3 4】

マスクによって画像にどのようなパターンを与えるかを視覚的に示したのが図6である。図5のマスクにおける「10」の位置を黒、「0」の位置を灰色、「-10」の位置を白で表現しており、画像には図6のような斜め線が現れることとなる。このようにして、元の画像に視認しにくい高周波成分として符号(色成分と画素数)を多重化できる。ここで、図5(a)のマスクと図5(b)のマスクを画像全体に対して符号に対応付けて適用する疑似コードを下記に示す。

【0 0 3 5】

```

01: int i, j, k, l;
02: int width = 640, height=480;
03: unsigned char *data = 画像データ;
04: int **maskA = マスクデータ;
05: bool isMaskA = true;
06: for(j = 0; j < height; j+=8){
07:   for(i = 0; i < width; i+=8){
08:     for(k = 0; k < 8; k++){
09:       for(l = 0; l < 8; l++){
10:         if(isMaskA == true){
11:           data[(i+k)+(j+l)*width] += maskA[k][l];
12:         }
13:       }
14:     }
15:   }
16: }
```

30

40

上記のコードは、一例として640×480画素の画像データを対象として、8×8領域に順次着目しつつ、当該領域がマスクすべき領域であればその領域の各画素に対応するマスクの各画素の値を加算する手順を示している。マスク対象の領域であることはisMaskAにより指定され、適用するマスクはmaskAにより指定されている。このように、多重化

50

による情報の埋め込みは上記の手法で実現可能である。なお、付加情報が多重化された領域を符号領域と呼ぶこともある。このような多重化技法を用いて、印刷対象の画像データに、その元となったPDLを多重化する。PDLは文字列であるので、各文字を2進符号で表し、それを図5に示したマスクを用いて画像データに多重化する。その際には、上述したコードに従えば、多重化対象の領域においてisMaskAを真にセットし、それ以外の領域ではisMaskAを偽にセットする。そして多重化対象の領域においては、"0"を多重化する領域についてはmaskAとして図5(a)のマスクを、"1"を多重化する領域についてはmaskAとして図5(b)のマスクを指定する。このような操作を、たとえば画像のラスタ走査方向に沿って行うことで、画像の色別の画素数を多重化する。なお多重化は特定の色について行ってよい。

10

【0036】

続いて、ステップs204に進み、印刷画像生成を行う。印刷画像生成では任意の公知の手法を用いて構わないが本実施形態では以下の方法を用いた例を説明する。ここでは、ステップs203で生成されたRGB画素値で構成される多重化済Bitmap画像の各画素に対して、

- ・色変換
- ・インク色分解
- ・出力特性変換
- ・量子化

の4つの処理を行う。

20

【0037】

色変換は、多重化済Bitmap(ビットマップ)画像のRGB情報を、MFP40で好適に記録出来る様に変換を行う処理である。その理由として、一般的にPDLの描画命令に記載されている色は、ディスプレイで好適に表現出来る色値が設定されており、そのままの値をプリンタで出力した場合には異なる色が出力されてしまうからである。

【0038】

具体的には、入力画素値(Rin, Gin, Bin)の組み合わせに対して、好適な出力画素値(Rout, Gout, Bout)の組み合わせを算出する為に、出力色ごとに3次元のルックアップテーブルを用いる。理想的には、入力値であるRin, Gin, Binがそれぞれ256階調を持つ為、 $256 \times 256 \times 256$ の合計16,777,216組の出力値を持つテーブルTable1[256][256][256][3]を準備し、

30

```
Rout = Table1[Rin][Gin][Bin][0]
Gout = Table1[Rin][Gin][Bin][1]
Bout = Table1[Rin][Gin][Bin][2]
```

とする事で実現出来る。また、ルックアップテーブルのグリッド数を256グリッドから例えば16グリッド等に減らし、複数のグリッドのテーブル値を補間して出力値を決定する等、テーブルサイズを小さくする公知の工夫を用いても良い。

【0039】

インク色分解は色変換処理の出力値であるRout, Gout, Boutを、インクジェット記録する各インク色の出力値に変換する処理である。本実施形態では、シアン、マゼンタ、イエロー、ブラックの4色インクジェットを想定する。この変換においても種々の実現方法がある。本実施形態では色変換処理同様に、出力画素値(Rout, Gout, Bout)の組み合わせに対して、好適なインク色画素値(C, M, Y, K)の組み合わせを算出する。その為に、出力色ごとの3次元のルックアップテーブルTable2[256][256][256][4]を用いて、

40

```
C = Table2[Rout][Gout][Bout][0]
M = Table2[Rout][Gout][Bout][1]
Y = Table2[Rout][Gout][Bout][2]
K = Table2[Rout][Gout][Bout][3]
```

とする事でインク色分解を実現出来る。また、テーブルサイズを小さくする公知の工夫を用いても良い。

50

【 0 0 4 0 】

続いて出力特性変換は、各インク色の濃度を記録ドット数率に変換する。具体的には、例えば各色256階調の濃度を、各色1024階調のドット数率Cout, Mout, Yout, Koutに変換する。そのためには、各インク色の濃度に対する好適な記録ドット数率を設定した2次元のルックアップテーブルTable3[4][256]を用いて、

```
Cout = Table3[0][C]
Mout = Table3[1][M]
Yout = Table3[2][Y]
Kout = Table3[3][K]
```

とする事で実現出来る。また、ルックアップテーブルのグリッド数を256グリッドから例えば16グリッド等に減らし、複数のグリッドのテーブル値を補間して出力値を決定する等、テーブルサイズを小さくする公知の工夫を用いても良い。

【 0 0 4 1 】

続いて量子化は、各インク色の記録ドット数率Cout, Mout, Yout, Koutを、実際の各画素の記録ドットのOn/Offに変換する。量子化の方法としては、誤差拡散法やディザ法等、任意の手法を用いて良い。ディザ法を例にすると、

```
Cdot = Halftone[Cout][x][y]
Mdot = Halftone[Mout][x][y]
Ydot = Halftone[Yout][x][y]
Kdot = Halftone[Kout][x][y]
```

とする事で、各画素位置に応じた閾値と比較する事で、各インク色の記録ドットのOn/Offを実現する事が出来る。ここでは、各記録ドットの発生確率は、Cout/1023, Mout/1023, Yout/1023, Kout/1023となる。

以上の様にする事で、ステップs 2 0 3 の印刷画像生成が終了する。

【 0 0 4 2 】

次にステップs 2 0 5 で、ステップs 2 0 4 で生成した印刷画像の印刷を行う。以上で、PDL原稿を記録用紙上に印刷する事が出来る。

【 0 0 4 3 】

以上で、PDLファイルデータを多重化したPDL原稿を記録用紙上に印刷する事が出来る。ここで、図6のパターンをみると、 $2px \times (ピクセル)$ の斜め線によってパターンが描かれていることがわかる。このようにPDLファイルデータが埋め込まれたレンダリング画像に基づいて印刷物を生成する処理を「多重化エンコード処理」ともいう。なお上記例ではRGB画像に画素数情報を多重化したが、CMYKに変換後の画像データに多重化してもよい。たとえば黄色は視覚的に目立たない色であるため、黄色成分に多重化することで、より一層、画像に与える視覚的な影響を抑えることができる。たとえば色数を10とすれば、各色について色リストが $10 \times 3 \times 8 = 240$ ビット、画素数 $10 \times 2 \times 8 = 160$ ビット、総色数が8ビットとすれば、総量は408ビットとなる。1ビットが 8×8 画素で表されるので、51個の領域で表すことができる。本例では、復元時の位置決めのために領域間には隙間を設けないが、復元の方法次第で間隔を空けてもよい。また同じ情報を繰り返し多重化してもよい。また色数が増えれば情報量も増えるため、色別の画素数情報を符号化してロスレス圧縮し、それを多重化してもよい。

【 0 0 4 4 】**色別画素数情報抽出処理**

続いて、多重化PDL抽出フローを説明する。図2(b)のステップs 2 1 1でPDLファイルデータが多重化された印刷物を読み取る。まず、スキャナ装置に印刷原稿をセットし、ステップs 2 1 1で原稿読取を行う。具体的には、スキャナデバイスを制御し、原稿にLED光を照射し、反射光を各画素に対向したCCD等の撮像素子でアナログ電気信号に変換する。

【 0 0 4 5 】

次に、ステップs 2 1 2で上記このアナログ電気信号をデジタル化して、デジタルRG

10

20

30

40

50

B 値を入力する。Bitmap取得処理では任意の公知の手法を用いて構わないが本実施形態では以下の方法を用いた例を説明する。ここでは、ステップ s 2 1 1 で取得された R G B 画素値で構成される Bitmap 画像の各画素に対して、

- ・ MTF補正
- ・ 入力補正
- ・ シェーディング補正
- ・ 色変換

の 4 つの処理を行う。

【 0 0 4 6 】

MTF (Modulation Transfer Function) 補正は、スキャナの読み取り性能の内、解像度に関する補正を行う。具体的には、スキャナの読み取りはフォーカス位置からのズレやレンズ自体の性能限界等により、画像がボケてしまう為、フィルタ処理等である程度の復元を行う。実際には完全に復元する程強い強調処理をしてしまうと、白飛びや画像ノイズ・ゴミ画素の協調等、画像弊害の方が目立つ為、画質改善と弊害のバランスを取って設計を行う。説明を簡単にする為に、画像中央部を 5 倍し、上下左右の画素値を -1 倍するエッジ強調フィルタの例を以下に記載する。

$$R'[x][y] = R[x][y] \times 5 - R[x-1][y] - R[x+1][y] - R[x][y-1] - R[x][y+1]$$

$$G'[x][y] = G[x][y] \times 5 - G[x-1][y] - G[x+1][y] - G[x][y-1] - G[x][y+1]$$

$$B'[x][y] = B[x][y] \times 5 - B[x-1][y] - B[x+1][y] - B[x][y-1] - B[x][y+1]$$

入力補正是元々が光子量である CCD の出力値を、人間の眼の感度に合った明度に変換する処理である。具体的には、例えば各色 4096 谐調の R' G' B' 信号を、各色 1024 谐調の色強度値 R'', G'', B'' に変換する。そのためには、各インク色の濃度に対する好適な記録ドット数率を設定した 1 次元のルックアップテーブル Table4[4][4096] を用いて、

$$R'' = \text{Table4}[0][R']$$

$$G'' = \text{Table4}[1][G']$$

$$B'' = \text{Table4}[2][B']$$

とする事で実現出来る。また、ルックアップテーブルのグリッド数を 4096 グリッドから例えば 256 グリッド等に減らし、複数のグリッドのテーブル値を補間して出力値を決定する等、テーブルサイズを小さくする公知の工夫を用いても良い。

【 0 0 4 7 】

シェーディング補正はスキャナデバイスを構成するレンズ、LED、CCD の製造ばらつきや組付けのばらつきに起因して生ずる各画素位置での読み取り感度の違いによって生ずる色濃度ムラを低減する処理である。具体的には、例えば各色 1024 谐調の R'' G'' B'' 信号を、各色 256 谐調の色強度値 R''' G''' B''' に変換する。そのためには、スキャナレンズの配置されている方向 (X 方向) に対してそれぞれ、各 X 画素位置用の濃度変換用の 1 次元ルックアップテーブル Table5[x][3][1024] を用いて、

$$R''' = \text{Table5}[x][0][R'']$$

$$G''' = \text{Table5}[x][1][G'']$$

$$B''' = \text{Table5}[x][2][B'']$$

とする事で実現出来る。また、ルックアップテーブルのグリッド数を 1024 グリッドから例えば 256 グリッド等に減らし、複数のグリッドのテーブル値を補間して出力値を決定する等、テーブルサイズを小さくする公知の工夫を用いても良い。

【 0 0 4 8 】

最後に、色変換処理を行う。これは、印刷時とは逆に、ここまで算出してきた R''', G''', B''' がそれぞれスキャナデバイス固有の値となっている為、これをディスプレイ上で表示する為に好適な Rout, Gout, Bout 値にする為である。

この為には、印刷時の色変換同様、入力値である R''', G''', B''' がそれぞれ 256 谐調を持つ為、256 × 256 × 256 の合計 16,777,216 組の出力値を持つテーブル Table6[256][256][256][3] を準備し、

$$Rout = \text{Table6}[R'''][G'''][B'''][0]$$

10

20

30

40

50

```
Gout = Table1[R''][G''][B''][1]
Bout = Table1[R''][G''][B''][2]
```

とする事で実現出来る。また、ルックアップテーブルのグリッド数を 256 グリッドから例えば 16 グリッド等に減らし、複数のグリッドのテーブル値を補間して出力値を決定する等、テーブルサイズを小さくする公知の工夫を用いても良い。

以上の様にすることでステップ s 212 の Bitmap 取得が終了する。

【0049】

図 4 は印刷及び読み取りによる画像弊害例を説明する図である。図 4 (a) は図 3 の PDL 原稿をレンダリングした状態、つまり図 2 中のステップ s 202 の出力 Bitmap 画像のうち、オブジェクト 301 の左上部分を表している。

図 4 (b) は図 3 の PDL 原稿の印刷原稿をスキャンし、図 2 中のステップ s 212 で取得した Bitmap 画像のうち、図 4 (a) の位置に対応する部分を表している。

【0050】

両者を比較すると分かるように、図 4 (a) 中の直線 400 は、一様な濃度 ($R=128, G=128, B=128$) を持っているが、図 4 (b) 中の 401 は、濃度的には不均一で、直線のエッジ部分にも凸凹が目立つ。また、直線周辺に若干濃くなった画素が存在し、全体的にも背景画像が少し暗い画像となっている。

これは、PDL 原稿を印刷し、スキャンする事によって得る Bitmap 画像には、PDL 原稿に対して以下の様な画像劣化が生じているからである。

- ・ステップ s 204 で行う量子化に伴う形状の劣化（量子化による諧調数低下）
- ・ステップ s 205 で行う印刷に伴う形状の劣化（印刷精度・記録用紙）
- ・ステップ s 211 で行うスキャンに伴う形状の劣化（スキャナによるボケ）

更には実際には、

- ・印刷原稿の汚れ等の劣化による形状の劣化
- が考えられる。

【0051】

そこで本実施形態では、ステップ s 213 以降の処理を実行することで劣化した画像を補正し、スキャンにより得られた画質の低下を抑制する。続くステップ s 213 で多重化された PDL ファイルデータを抽出する。以下に、Bitmap 画像解析例の概要説明を行う。

【0052】

まず、画像データから多重化された情報（付加情報とも呼ぶ）が埋め込まれている位置の検出を行う。具体的には、画像データの空間周波数特性を解析することにより埋め込み位置の検出を行う。図 7 は、埋め込みに用いたパターンの空間周波数の特性を表す図である。横軸が水平方向の周波数、縦軸が垂直方向の周波数を表し、原点から遠ざかるにつれて高周波域となる。本実施形態では、図 6 に示すように、2 つのパターンを画像に埋め込んでいる。埋め込み例では RGB の B 成分に 10 の加算及び減算を行っている。これにより、図 6 (a) のパターンは、図 7 のライン A01 に大きなパワースペクトルを発生させる。同様に、図 6 (b) のパターンはライン A02 に大きなパワースペクトルを発生させる。このパワースペクトルを、たとえば 8×8 画素領域ごとに検出することにより、「0」か「1」のデータ抽出を行う。なお、検出の前処理として、エッジ検出を行うことにより、パワースペクトルの強調をすることも可能である。

【0053】

周波数解析によるデータの抽出では、画像データからの解析エリアの正確な切り出しが必要となる為、座標位置のズレを補正する処理が行われる。例えば、画像内からの 8 ピクセル \times 8 ピクセル領域の切り出しと周波数解析とを、縦横それぞれの方向に 1 ピクセルずつずらしながら縦横に繰り返す。そしてたとえば横 8 ピクセル、縦 8 ピクセルずつずらすまで計 64 回繰り返し、最もスペクトルが強くなる箇所を切り出しの基準位置とする方法がある。このとき周波数解析の対象となる領域は 16 ピクセル \times 16 ピクセルの領域となり、付加情報をすきまなく多重化した領域内であれば、少なくとも 1 つの付加情報がそ

10

20

30

40

50

に含まれるはずである。そのため上述したように基準となる領域の位置を決めることができる。位置検出が完了した後に、その位置を基準として、多重化の規則に従って付加情報の抽出を行う事で、抽埋め込まれていた「0」「1」数列が得られる。

【0054】

なお上記方法で基準位置を特定しようとする画像中の位置として、付加情報を埋め込んだ符号領域の先頭を選択すれば、特定された基準位置は符号領域の先頭を示すことになる。そうでない場合であっても、基準として位置が特定された領域が、多重化された符号列のなかでどの位置にあるかは概ね特定できるので、その位置を基準として多重化された符号列の先頭を特定することはできる。先頭として特定した 8×8 画素領域及びその前後の 8×8 画素領域について空間周波数を解析することで、付加情報が多重化された領域の先頭を特定できる。これにより付加情報の長さは固定的であってよいので、付加情報が多重化された領域全体を特定できる。こうして特定した符号領域全体を、図 7 に示した空間周波数の解析により、2 値データへと復号することができる。

【0055】

続くステップ s 214 で抽出した PDL ファイルを保存する。この様に PDL ファイルを印刷物のスキャン画像から抽出して保存する事で、印刷及びスキャンに伴う画像劣化の無い原稿を取得する事が出来る。また、抽出した PDL ファイルから生成した画像データを元に印刷を再び行う事で、画像劣化の無いコピーを行う事が可能となる。

【0056】

以上の構成および手順により、ビットマップ画像データに、その画像データの元となつた PDL データを多重化する。こうすることで、媒体上への画像の形成とその光学的な読み取りを経ても劣化しない PDL データを画像から復元でき、劣化のない画像を再生することができる。

【0057】

[多重化方法の他の例]

図 5 及び図 6 では、 8×8 画素単位に情報を埋め込む例を説明した。これは高周波領域に画像を埋め込む事によって、ユーザに視認されにくい埋込を行う例である。しかし、それら高周波領域への多重化だけでなく、任意の多重化方法を行って構わない。

【0058】

図 8 に多重化方法の他の例を示す。図 8 (a) は、マトリクス型二次元コードの 1 つである QR コード（登録商標）のパターン例である。本例では、PDL ファイルデータを QR コード化し、それを視認されにくい形で印刷物に多重化する。図 8 (b) は実際の印刷パターンを表していて、 8×8 画素単位に 1 ドットだけ記録するパターンとなっている。図 8 (a) 中 800 の黒色画素に対応するドットが図 8 (b) 中 801 の 1 ドットに対応する。また、図 8 (a) 中の空白画素に対応する位置には図 8 (b) 中にはドットを形成しない。

【0059】

こうする事で、視認されにくい多重化パターンを記録用紙上に構成する事が出来る。具体的には、図 2 中のステップ s 203 で PDL ファイルデータを図 8 (a) の様に QR コード化し、それを離間したドット群である図 8 (b) の様な印刷データとしてレンダリング画像に対して重畳する。また、離間ドットを構成するインクとしてはイエローインクが最も視認され辛いので、イエローインクで図 8 (b) のパターンを構成する様にする事で、より視認されにくい多重化パターンを記録用紙上に構成する事が出来る。ここで、元の画像にもイエロー成分が含まれていることもあり得るが、QR コードの領域は、そのイエロー成分を QR コード対応の図 8 (b) に示したパターンで置換することで、元の画像データのイエロー成分による干渉を防ぐことができる。これにより QR コードから PDL データに記述した通りの画像を復元できる。

【0060】

本方式の多重化方式の場合、ステップ s 213 の PDL データ抽出は、読み取った Bitmap 画像中、 8×8 画素単位でイエロードットが記録されているかを判定する。そ

して、図 8 (b) のイエロー成分のパターンから、図 8 (a) 相当のQRコードパターンを再構成し、それを復号する。こうする事でPDLファイルデータを抽出すればよい。再構成する際にはイエロードットを含む 8×8 領域を QR コードの 1 つの黒セルで置き換えるべき。なお多重化する情報は、第一の実施形態と同様、色ごとの画素数情報であってもよく、その場合には色別画素数情報を QR コード化して、図 8 (b) に示したように画像データに多重化すればよい。

【 0 0 6 1 】

なお 1 つの QR コードでは容量が足りない場合には、複数の QR コードに P D L データを分割して符号化し、多重化してもよい。この場合には、先頭位置に埋め込む QR コードに、分割された数を含めてよい。

10

【 0 0 6 2 】

また、他の多重化例としては、図 2 中のステップ s 2 0 4 の印刷画像生成処理中の量子化処理時に閾値変調を行って多重化する方法も存在する（特許文献 2 ）が、この手法を用いても良い。

【 0 0 6 3 】

以上種々の多重化例を説明して来たが、一般的に多重化すべき情報量が増えた場合、記録 Dot 数や変調量を増大させる必要が生じ、その結果として記録画像全体の画像劣化が増大する事となる。

【 0 0 6 4 】

また、上述した様に、図 3 の原稿ページ 3 0 0 は横幅 600 画素、縦幅 900 画素から構成される為、1 ページは全部で 540,000 画素から構成される。各画素の色を R/G/B 各 8 bit の 256 諧調で表現する場合には、各画素 3 Byte、1 ページ合計で 1,620,000 Byte = 1.6 M Bytes となる。それに対して、P D L 状態でのファイルサイズは、106,200 Bytes となり、約 6.6% 程度となる。以下にファイルサイズ試算の内訳を示す。

20

- ・描画命令部分 約 350 文字 (1 文字 2 Bytes として 700 Bytes)
- ・ "PORTRAIT.jpg" 300×475 画素、各画素 3 Bytes で全 427,500 Bytes
ここで圧縮率を 20% とすると 85,500 Bytes
- ・ "STD" フォントデータ 後述の理由で 20,000 Bytes と想定

ここで、フォントサイズの例としては、特許文献 3 に開示されている情報を元に試算した。特許文献 3 には、以下の様に記載されている。

30

「 2 0 ページで試算したところ、フォント埋め込みなしの P D F サイズが約 6 0 K B に対し、フォント埋め込みありの P D F サイズは、4 6 0 K B となり、9 0 0 % 近いサイズ増加となる。（ページあたりの新規文字が 7 5 % として計算）」

そこで、1 ページのフォント埋め込み量を $400 \text{ kBytes} / 20 = 20 \text{ kBytes}$ とした。

【 0 0 6 5 】

以上説明した様に、ステップ s 2 0 2 でレンダリングした画像と比べて、ステップ s 2 0 1 で取得した P D L サイズは約 6.6% と非常に小さいサイズとなっている。よって、印刷原稿上に多重化する際には、B i t m a p 画像よりも P D L ファイルを多重化する方が、多重化に伴う印刷画像全体の画像劣化が大幅に低減出来る事となる。

【 0 0 6 6 】

また、P D L を直接埋め込む事によって、原稿をスキャンする事で P D L ファイルを完全に復元する事も可能である。その場合、図 3 に対応付けた各描画命令が全て復元出来る為、例えば図 4 (b) の画像を完全に図 4 (a) と同じにする事が出来る。同様に、図 3 中のオブジェクト 3 0 3 や 3 0 4 の様な文字・細線についても復元が可能である。コピー処理は一般的に読み取りから印刷のステップを経る為、画像の劣化が避けられないが、スキャン時に元原稿 P D L を抽出出来れば、抽出した P D L を印刷する事によって、ほぼ完全なコピーが実現出来る事となる。

40

【 0 0 6 7 】

更に、印刷時に縮小または拡大が伴う場合、従来は細線および文字については線の細りまたは太りが生じ、最悪のケースでは文字および細線が判読不能或いは消滅となるケース

50

が有った。しかし、本実施形態に係る発明を用いると、多重化した P D L を抽出出来れば良い事になる。

【 0 0 6 8 】

別のアプローチとして、従来は縮小または拡大コピー時には文字および細線も縮小または拡大が伴う為、視認性が劣化してしまうケースが有った。しかし本実施形態のように P D L を抽出できれば、TEXT描画命令に対して、ベクタデータを基にしてサイズ変更が可能となる為、以下の様なサービスが実現可能である。

- ・ 50 %縮小 + 文字サイズ200 %設定で、「文字サイズ維持縮小コピー」
- ・ 200 %拡大 + 文字サイズ50 %設定で、「文字サイズ維持拡大コピー」
- ・ 等倍 + 文字サイズ200 %設定で、「文字拡大コピー」
- ・ 等倍 + 文字サイズ50 %設定で、「文字縮小コピー」。

10

【 0 0 6 9 】

このように、ベクタデータである文字については変倍による劣化が生じにくく、また文字のみを対象として変倍することができるため、それを利用したサイズ変更が可能となる。同様に、コピーにおいて、LINE描画命令等、他の命令に対しても線の太さや線の種類・色の変更、塗りパターンの変更等を任意に行う事が出来る事となる。

【 0 0 7 0 】

以上の構成および手順により、画像データの元となった P D L データを、たとえばマトリクス型二次元コードへ符号化し、P D L データから生成したビットマップ画像データに多重化する。こうすることで、媒体上への画像の形成とその光学的な読み取りを経ても劣化しない P D L データを画像から復元でき、劣化のない画像を再生することができる。

20

【 0 0 7 1 】

(第二の実施形態)

第一の実施形態では、P D L ファイル全体をレンダリングされた B i t m a p 画像に埋め込む例を説明したが、本実施形態では、P D L ファイルサイズが大きい場合の画像劣化を避ける為に、P D L ファイルサイズを好適に低減する例を説明する。本実施形態では、P D L ファイルをそのまま多重化するのではなく、ファイルサイズを低減した多重化用 P D L を作成して多重化を行う。

【 0 0 7 2 】

図 9 は第二の実施形態における P D L 多重化印刷フローを説明する図である。図 9 中、ステップ s 9 0 1 ~ s 9 0 2 は図 2 中のステップ s 2 0 1 ~ s 2 0 2 と同じ処理なので説明を省略する。

30

【 0 0 7 3 】

ステップ s 9 0 3 では、P D L ファイルから画像ファイル（ビットマップ画像）を削除する事で、ファイルサイズを低減した多重化用 P D L ファイルを作成する。特に画像ファイルを削除する理由としては、第一の実施形態で行ったファイルサイズ試算の内訳に記載した様に、画像ファイルのファイルサイズが最も大きいからである。一方、描画命令部分は最もファイルサイズが小さく、かつ細線や文字の描画命令を含む為、優先的に多重化用 P D L ファイルに含める事が好ましい。中間的なサイズであるフォントデータについても、文字描画に必要な要素である為、多重化用 P D L ファイルに含める事が好ましい。

40

【 0 0 7 4 】

図 10 は第二の実施形態における多重化用 P D L 原稿例を説明する図である。図 10 中、オブジェクト 1 0 0 0 ~ 1 0 0 4 は、図 3 中のオブジェクト 3 0 0 ~ 3 0 4 と同じなので説明を省略する。しかし、図 10 中の領域 1 0 0 5 は図 3 中のビットマップ画像 3 0 5 と異なり、画像ファイルが存在しない。領域 1 0 0 5 は、元の P D L データではビットマップ画像が配置されていた領域を示している。この実現方法としては種々の方法を採用する事が可能であり、例えば、以下のような方法が考えられる。

- ・ 多重化用 P D L 中の IMAGE 描画命令を、

< IMAGE > 250, 350, 550, 825, "" </ IMAGE >

の様に「ファイル無し」状態で記述する。

50

・多重化用 P D L 中に、"PORTRAIT.jpg"ファイルを含めない様に構成し、IMAGE描画命令を実行不可能とする。

・多重化用 P D L 中のIMAGE描画命令事態を、

<IMAGE> 250, 350, 550, 825, "NULL.jpg" </IMAGE>

の様に記述し、別途"NULL.jpg"という「非常に画素数の小さい画像（例：8 × 8 画素の空白画像）」に置きかえる。

【0075】

本実施形態では、その中の、「ファイル無し」状態での記述方法を採用する。理由としては、スキャナを用いた「P D L 復号時」に明確に「ファイルを削除した」事を、多重化した P D L データを画像から復元した装置へと通知可能であるからである。

10

【0076】

図11は第二の実施形態における多重化用 P D L 生成フローを説明する図である。まず、ステップs1101でP D L ファイルを取得する。続いて、ステップs1102でP D L ファイルの解析を行う。これは第一の実施形態で用いたP D L ファイルに含まれた命令を1行目から順番に着目して解析する。

【0077】

ステップs1103で着目した命令がIMAGE描画命令であるか否かを判定する。判定結果がYesの場合にはステップs1104へと進み、判定結果がNoの場合にはステップs1105に進む。

20

【0078】

ステップs1104では、P D L 中の画像データを削除して着目した命令を多重化用 P D L に追加する。具体的には上述の様に、IMAGE描画命令中のファイル名を""に置き換えて、多重化用 P D L に追加する。

【0079】

ステップs1105では、P D L 中の描画命令をそのまま多重化用 P D L に追加する。

【0080】

ステップs1106では、全描画命令の処理が終了したか否かを判定し、判定結果がYesの場合はステップs1107に進む。判定結果がNoの場合にはステップs1102に進み処理を継続する。具体的な全描画命令の処理終了判断方法としては、ページの終了を示す</PAGE>タグに到達したかという方法を取る事が可能である。

30

【0081】

ステップs1107では生成した多重化用 P D L を保存する。本実施形態で生成した多重化用 P D L の例を以下に示す。

<PAGE=001>

```
<BOX> 50, 75, 200, 300, GRAY, STRIPE </BOX>
<BOX> 420, 75, 550, 300, BLACK, FULL </BOX>
<BOX> 50, 620, 200, 825, BLACK, NULL </BOX>
<TEXT> 50, 620, 200, 825, BLACK, STD-18, "ABCXYZ123" </BOX>
<LINE> 25, 25, 25, 700, BLACK, PLAIN, 3 </LINE>
<LINE> 25, 25, 450, 25, BLACK, PLAIN, 3 </LINE>
<IMAGE> 250, 350, 550, 825, "" </IMAGE>
</PAGE>
```

40

第一の実施形態におけるP D Lとの差異は、8行目の<IMAGE>描画命令中のファイル名が""となっている点である。

【0082】

図9に戻り、続くステップs904では、レンダリング画像にステップs903で生成した多重化用 P D L ファイルを多重化した画像を生成する。ここでは第一の実施形態に対して多重化すべき P D L ファイルサイズが低減しているので、多重化に伴う画像劣化が低減される事となる。

【0083】

50

続くステップ s 9 0 5 ~ s 9 0 6 の処理は図 2 中ステップ s 2 0 4 ~ s 2 0 5 と同じ処理なので説明を省略する。

【 0 0 8 4 】

こうして形成された画像データからは図 2 (b) の手順で P D L データを復元できる。ただしスキャンした画像から復元した P D L の I M A G E 描画命令には画像ファイルが含まれていない。そこで画像の生成時には、復元した P D L に含まれた I M A G E 描画命令の画像ファイルが空であれば、該当する領域の画像データをスキャンした画像データから切り出す。そしてそれを I M A G E 描画命令による描画対象の画像ファイルとして P D L データを再構成し、保存しておけばよい。

【 0 0 8 5 】

こうする事で、 P D L 復号側は、スキャンした画像中、 I M A G E 描画命令領域に対応した画像部分と合成する等の方法で文字および細線の形状を復元しつつ画像部も持つ P D L を生成する事が可能となる。また、フォントデータのデータ量が画像ファイルデータ量に次いで大きい。よって、例えば TEXT 命令で指定されているフォントが、 P C 5 0 若しくは M F P 4 0 で標準的に保持しているフォントである等、多重化しなくとも問題無い事が保障されている場合には、多重化 P D L から削除してもよい。

10

【 0 0 8 6 】

以上のようにして、 P D L データを画像に多重化することで、再生時に高品質の画像を再生できる。加えて、多重化する P D L データからサイズの大きなビットマップ画像データを除外することで、多重化するデータサイズを小さくできる。

20

【 0 0 8 7 】

(第三の実施形態)

第二の実施形態では、 P D L 中の描画命令種別に応じて多重化用 P D L ファイルに含むか否かを判断する例を説明したが、本実施形態では、描画命令等の実サイズに応じて決定する例を説明する。これまで、前提条件として、

I M A G E 描画命令 + 画像ファイルサイズ > 他の描画命令

という事で説明をして来たが、実際には画像ファイルであっても画素数が少ない為にファイルサイズが小さい場合もあり、逆に L I N E 描画命令で有ってもその数が非常に多くなればファイルサイズも大きくなる。

【 0 0 8 8 】

30

図 3 中のボックスオブジェクト 3 0 2 を例として説明する。ボックスオブジェクト 3 0 2 に対応する描画命令は、

< BOX > 420, 75, 550, 300, BLACK, FULL < /BOX >

と説明した。これに対して、例えば、

< BOX > 420, 75, 500, 300, BLACK, FULL < /BOX >

< BOX > 500, 75, 550, 300, BLACK, FULL < /BOX >

という様に、2つの B O X 描画命令で記述する事も可能である。どの描画命令を用いるかは P C 5 0 上で動作するアプリケーションの仕様に依存する為、例えば B O X 描画命令を用いず、全てを画素単位の L I N E 命令で記述する仕様のアプリケーションも存在し得る。

【 0 0 8 9 】

40

仮に各 1 画素単位で L I N E 命令を記述した場合、図 3 中のオブジェクト 3 0 2 は以下の様に、 $(550-420+1) \times (300-75+1) = 29,606$ 個の L I N E 描画命令によって構成される事となる。

< LINE > 420, 75, 420, 75, BLACK, FULL < /BOX >

< LINE > 421, 75, 421, 75, BLACK, FULL < /BOX >

...

< LINE > 550, 75, 550, 75, BLACK, FULL < /BOX >

< LINE > 420, 76, 420, 76, BLACK, FULL < /BOX >

...

< LINE > 550, 300, 550, 300, BLACK, FULL < /BOX >。

50

【 0 0 9 0 】

一つのLINE描画命令は43文字から構成されるため、1文字2Bytesとして1描画命令あたり86Bytesとなる。よって、29,606個の描画命令で合計 $86 \times 29,606 = 2,546,116$ Bytes 2.6Mbytesとなる。これは、第一の実施形態で用いたPDLファイルサイズ106,200Bytesよりも、更にはレンダリング画像データサイズ1,620,000Byte 1.6Mbytesよりも大きなサイズとなる。このように、仮に文字・線画の描画命令であっても、多重化用PDLに含めると画像劣化が大きくなり過ぎるケースも存在する。よって、本実施形態においては、PDLを所定サイズの領域単位のPDLに分割し、それぞれの領域のPDLサイズを所定の閾値サイズと比較し、閾値サイズ以下の領域のPDLを多重化PDLとする。こうすることで多重化するPDLデータのサイズを抑制する。

10

【 0 0 9 1 】

図12は第三の実施形態におけるタイル分割適応的PDL多重化印刷フローを説明する図である。ステップs1201及びs1202は図9中のステップs901及びs902と同じ処理なので説明を省略する。ステップs1203では、PDLを所定サイズの領域単位のPDLに分割する。

【 0 0 9 2 】

図13は第三の実施形態におけるタイル分割PDL原稿例を説明する図である。図13中、1300～1305は図3中の301～305と同じなので説明を省略する。本実施形態において、図13中原稿ページ1300は領域を 6×6 の36領域に分割されている。破線がその領域境界を示していて、原稿ページ1300の上側に記載されている数字0～5が水平方向の分割領域番号を、原稿ページ1300の左側に記載されている英字A～Fが垂直方向の分割領域番号を表す。各領域は「領域(英字)(数字)」と表す事とする。例えば左上の領域は「領域A0」であり、右上、左下、右下の領域はそれぞれ「領域A5」「領域F0」「領域F5」と表す。また、各領域のサイズは横100画素×縦150画素とする。この領域サイズは任意に設定可能であるが、領域サイズを小さくする程、領域数が大きくなってしまう。

20

【 0 0 9 3 】

図14は第三の実施形態におけるタイル分割多重化用PDL原稿例を説明する図である。図14中、1400～1404は図13中の1300～1304と同じなので説明を省略する。ハッチング領域1406は、多重化用PDLにおいて各分割領域のPDLデータサイズが閾値を超えるサイズと判断された領域を表す。ハッチング領域1406の内、領域C2～領域F5の矩形領域は図13中のIMAGE描画領域1305に対応する領域である。また、領域A4～領域B5の矩形領域は図13中のベタ部1302であり、上述通り、本実施形態では多くのLINE描画命令から構成されている。

30

【 0 0 9 4 】

この様に、データサイズの大きな領域を多重化用PDL生成対象から外す事で、多重化対象のデータサイズを小さく抑える事が出来、画像劣化度合いを低減する事が出来る。

【 0 0 9 5 】

図15は第三の実施形態における多重化用分割PDL生成フローを説明する図である。まず、ステップs1501でPDLを取得する。

40

【 0 0 9 6 】

ステップs1502で各タイル(領域A0～F5の36領域)のデータサイズを示す変数SIZE_XYを下式のように初期化(=0)する。

```

SIZE_A0 = 0
SIZE_A1 = 0
...
SIZE_F5 = 0.

```

【 0 0 9 7 】

ステップs1503で描画命令を解析する。図11中のステップs1102同様に、PDLをその1行目から順番に命令に着目して解析する。

50

【 0 0 9 8 】

ステップ s 1 5 0 4 で描画命令をタイルサイズに分割する。例えば、2行目の
`< BOX > 50, 75, 200, 300, GRAY, STRIPE </BOX>`

は、図 13 中の領域 1301 に相当する描画命令であり、領域 A0 ~ B1 の 4 領域に跨っている描画命令である。よって、各領域に対して上記命令を記述する必要がある。よって、ステップ s 1 5 0 5 で分割対象のタイル毎に描画命令を追加する。

【 0 0 9 9 】

2 行目の BOX 描画命令の分割が完了した状態で、各領域の多重化用 PDL は以下の様になっている。

```
< PAGE=001 > 10
< AREA=A0 >
  < BOX > 50, 75, 200, 300, GRAY, STRIPE </BOX>
</AREA>
< AREA=A1 >
  < BOX > 50, 75, 200, 300, GRAY, STRIPE </BOX>
</AREA>
< AREA=B0 >
  < BOX > 50, 75, 200, 300, GRAY, STRIPE </BOX>
</AREA>
< AREA=B1 >
  < BOX > 50, 75, 200, 300, GRAY, STRIPE </BOX>
</AREA >
</PAGE>.
```

【 0 1 0 0 】

ここで、各描画命令は領域を超える位置への描画は無視する事とする。そうする事で、「描画命令の分割 = 描画命令のコピー」として処理を簡略化している。

【 0 1 0 1 】

次に、ステップ s 1 5 0 6 で分割領域のタイル毎にデータサイズを加算する。

2 行目の BOX 描画命令は 43 文字から構成されているので、1 文字 2Bytes として各分割領域のデータサイズを 86Bytes 増加させる。この時点で、各領域のデータサイズは、

```
SIZE_A0 = 86
SIZE_A1 = 86
SIZE_A2 = 0
...
SIZE_B0 = 86
SIZE_B1 = 86
SIZE_B2 = 0
...
SIZE_F5 = 0
```

となる。以後、1 描画命令は概ね 86bytes として概算を行う。

【 0 1 0 2 】

ステップ s 1 5 0 7 で全描画命令の分割が終了したかを判定する。判定結果が Yes の場合にはステップ s 1 5 0 8 へ進む。判定結果が No の場合にはステップ s 1 5 0 3 に進み、次の行の描画命令の分割に進む。

【 0 1 0 3 】

3 行目からは上述の通り、大量の LINE 命令が繰り返される為、両機 A4 ~ B5 の 4 領域のデータサイズは非常に大きくなる。具体的には、

```
SIZE_A4 = 529,416 ( = 86 × (500-420+1) × (150-75+1) )
SIZE_A5 = 333,336 ( = 86 × (550-500+1) × (150-75+1) )
SIZE_B4 = 1,051,866 ( = 86 × (500-420+1) × (300-150+1) )
```

10

20

30

40

50

`SIZE_B5 = 662,286 (= 86 × (550-500+1) × (300-150+1))`

となる。

【0104】

ステップs1508から、各領域のデータサイズの判定を行う。ステップs1509では、判定対象領域のデータサイズが閾値を超えたか否かを判定する。判定結果がYesの場合にはステップs1510に進む。判定結果がNoの場合にはステップs1511に進む。閾値としては必要十分に小さい値を設定する事が好ましいが、本実施形態においては、描画命令35個分を目安として、3,010Bytes (= 86 × 35)とする。こうする事で、最大でも36領域で、108,360Bytesとなり、第一の実施形態で試算したPDLファイルサイズとほぼ同等となる。

10

【0105】

ステップs1510では、サイズが閾値を超過した領域について、その描画命令群をNull画像描画命令に差し替える。具体的には、第二の実施形態と同様に、

<IMAGE> 250, 350, 550, 825, "" </IMAGE>

の様な1命令とする。こうする事で、領域のデータサイズが閾値を超えた領域については、描画命令数1、サイズとしては86Bytes程度となる。

【0106】

ステップs1511では、全領域（本実施形態では36領域）の確認が終了したかを判定する。判定結果がYesの場合にはステップs1512に進む。判定結果がNoの場合にはステップs1508に戻り、次の領域のデータサイズ判定を行う。

20

【0107】

ステップs1512では、全領域の多重化用分割PDLを結合して、多重化用PDLを生成して保存する。

【0108】

この様にする事によって、領域内のPDLデータサイズが閾値を超えない多重化用PDLを生成する事が出来る。

【0109】

図12に戻り、ステップs1204では、ステップs1203で生成した多重化用PDLをステップs1202で生成したレンダリング画像に多重化して多重化画像を生成する。ステップs1204～s1206は図9中のステップs904～s906と処理内容が同じである為説明を省略する。

30

【0110】

このようにして生成された画像データが印刷され、スキャンされると、図2(b)の手順でそこから多重化されたPDLデータが復元できる。復元されたPDLデータからは、所定値を超えたデータ量の領域に関しては、その領域の指定された位置の描画命令は復元できるものの、その内容は削除されている。そこで復元したPDLデータと、スキャンした画像データとを保存しておいてよい。あるいは、復元したPDLの描画命令でその内容が空欄の命令については、IMAGE描画命令に置き換えてよい。この場合には、描画命令で指定された領域に書き込むデータとして、その領域に相当するスキャンした画像データの部分をファイル化してそれを指定してもよい。

40

【0111】

以上の様に、多重化用PDLのデータサイズを低減する事によって、多重化に伴う画像劣化を低減する事が出来る。また、本実施形態の方法であれば、描画命令の種類に因らず、描画命令サイズが大きな領域を多重化対象データから外して画像劣化を低減可能である。

【0112】

また、第二の実施形態同様に、サイズの大きな領域においては、

<IMAGE> 250, 350, 550, 825, "" </IMAGE>

の様に元々の領域に描画命令が存在していた事を明示しているので、PDL復号側はスキャンした画像中、IMAGE描画命令領域に対応した画像部分と合成する等の方法で文字・細線の形状を復元しつつ画像部も持つPDLを生成する事が可能となる。

50

【 0 1 1 3 】

また、本実施形態では P D L 中の描画命令数に注目してサイズ削減を行ったが、別途フオントデータサイズや画像データサイズにそれぞれ閾値を設けて多重化 P D L に含めるか否かを判定しても良い。

【 0 1 1 4 】

上記構成により、本実施形態によれば、画像データに多重化する P D L データのデータ量の上限を、各領域の多重化するか否かを定める多重化閾値の総和に抑えることができる。このため、このデータ量の上限を、1枚のシートに形成する画像に多重化できるデータ量としておけば、符号化した P D L データを1枚のシートに確実に多重化することが可能となる。加えて、上述した第一実施形態及び第二実施形態の利点を享受することもできる。

10

20

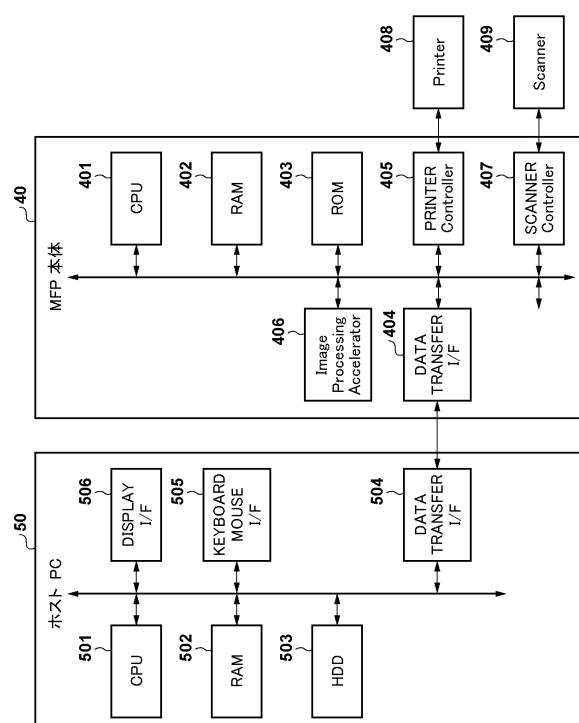
30

40

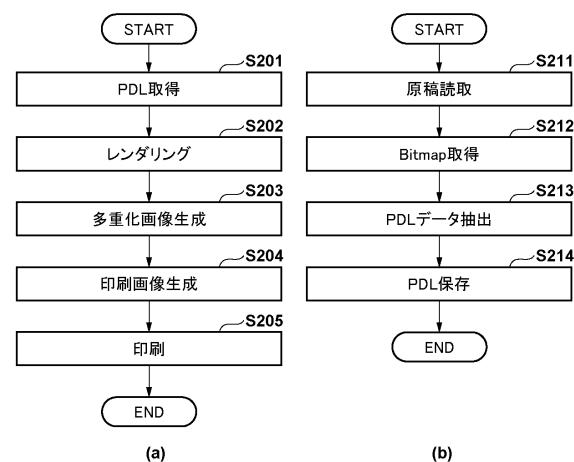
50

【図面】

【図 1】



【図 2】



10

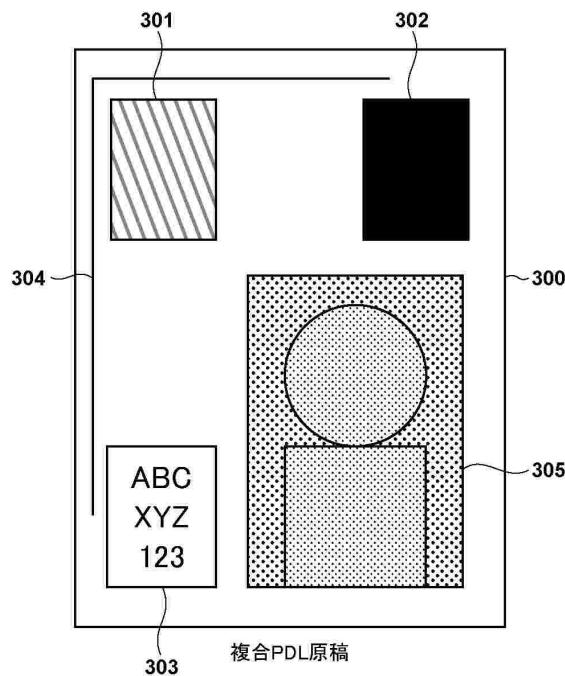
20

30

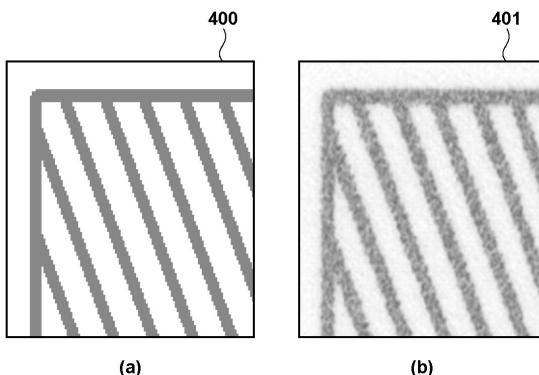
40

50

【図 3】



【図 4】



【図 5】

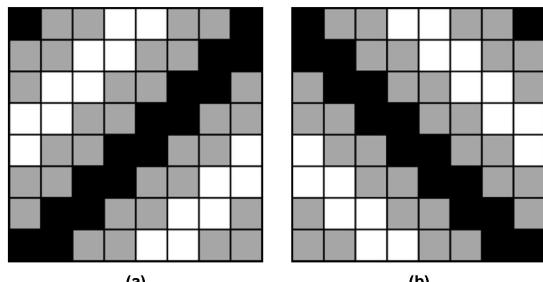
10	0	0	-10	-10	0	0	10
0	0	-10	-10	0	0	10	10
0	-10	-10	0	0	10	10	0
-10	-10	0	0	10	10	0	0
-10	0	0	10	10	0	0	-10
0	0	10	10	0	0	-10	-10
0	10	10	0	0	-10	-10	0
10	10	0	0	-10	-10	0	0

(a)

10	0	0	-10	-10	0	0	10
10	10	0	0	-10	-10	0	0
0	10	10	0	0	-10	-10	0
0	0	10	10	0	0	-10	-10
-10	0	0	10	10	0	0	-10
-10	0	0	0	10	10	0	0
0	0	10	10	0	0	-10	-10
0	10	10	0	0	-10	-10	0

(b)

【図 6】

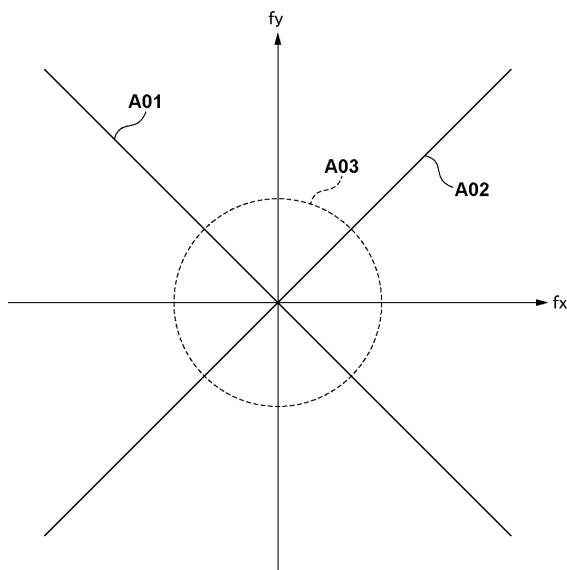


(a)

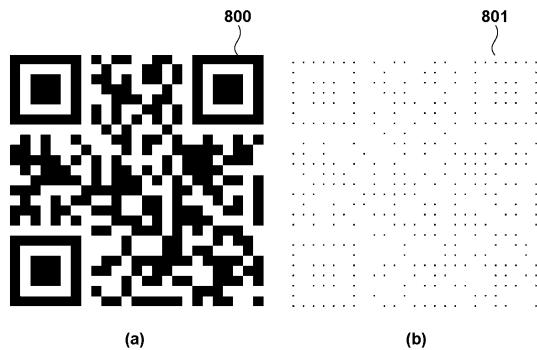
(b)

10

【図 7】



【図 8】



(a)

(b)

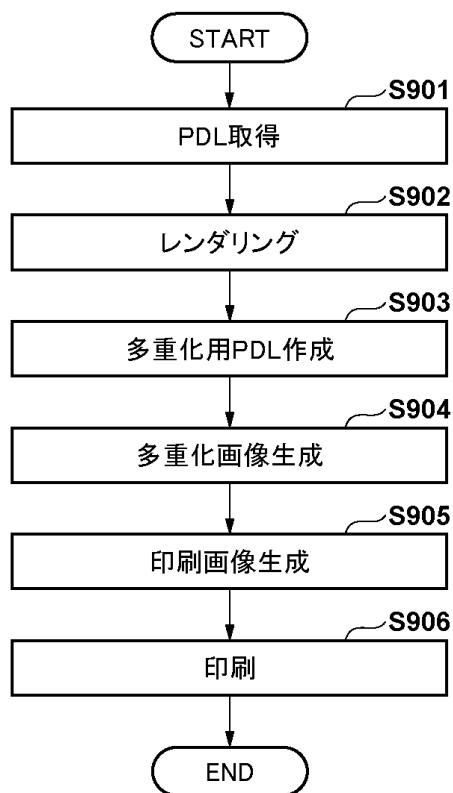
20

30

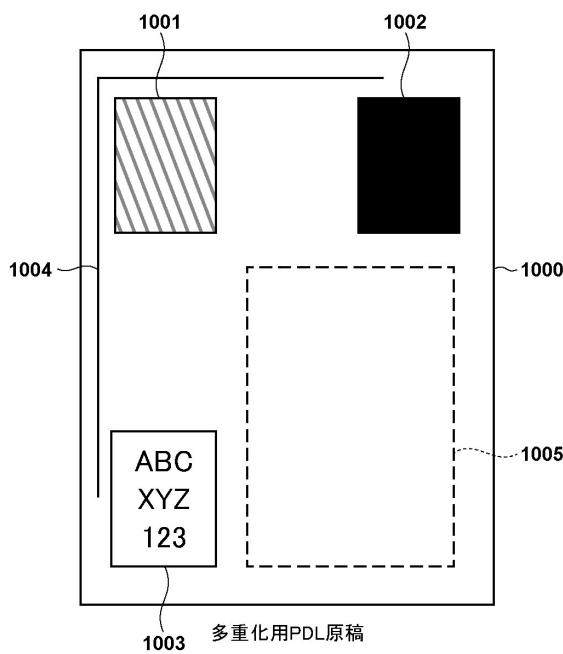
40

50

【図 9】



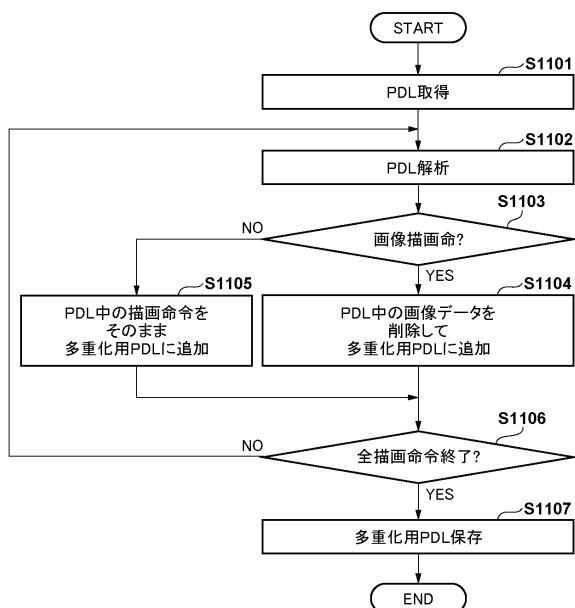
【図 10】



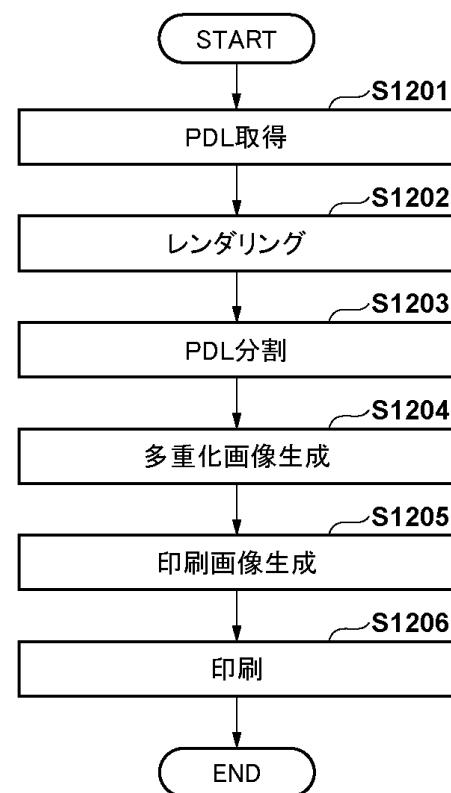
10

20

【図 11】



【図 12】

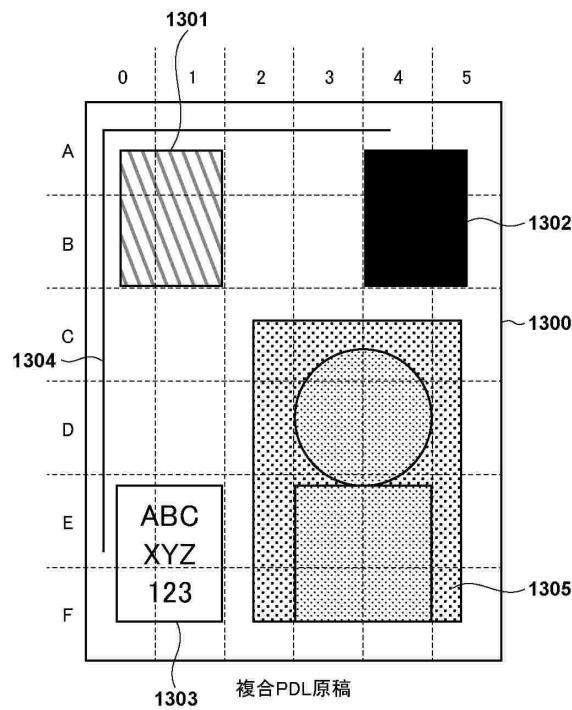


30

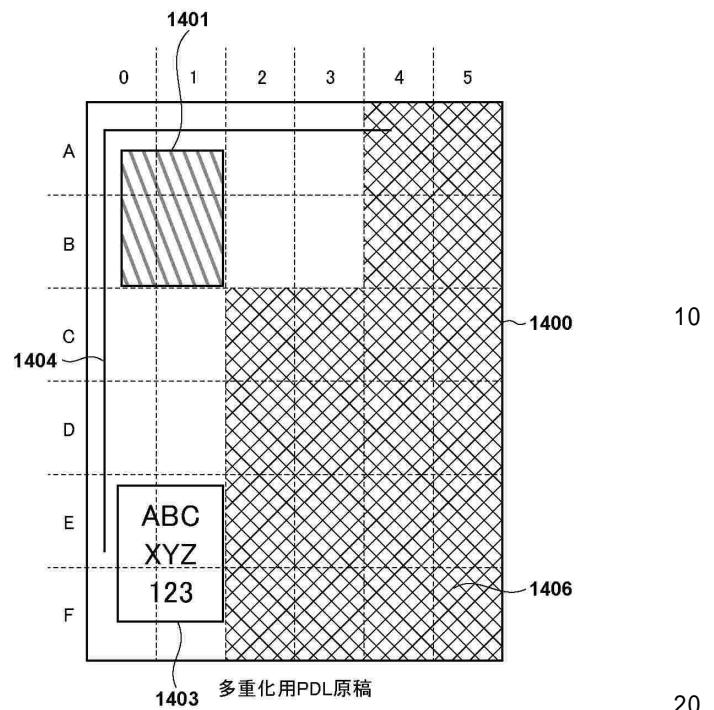
40

50

【図13】



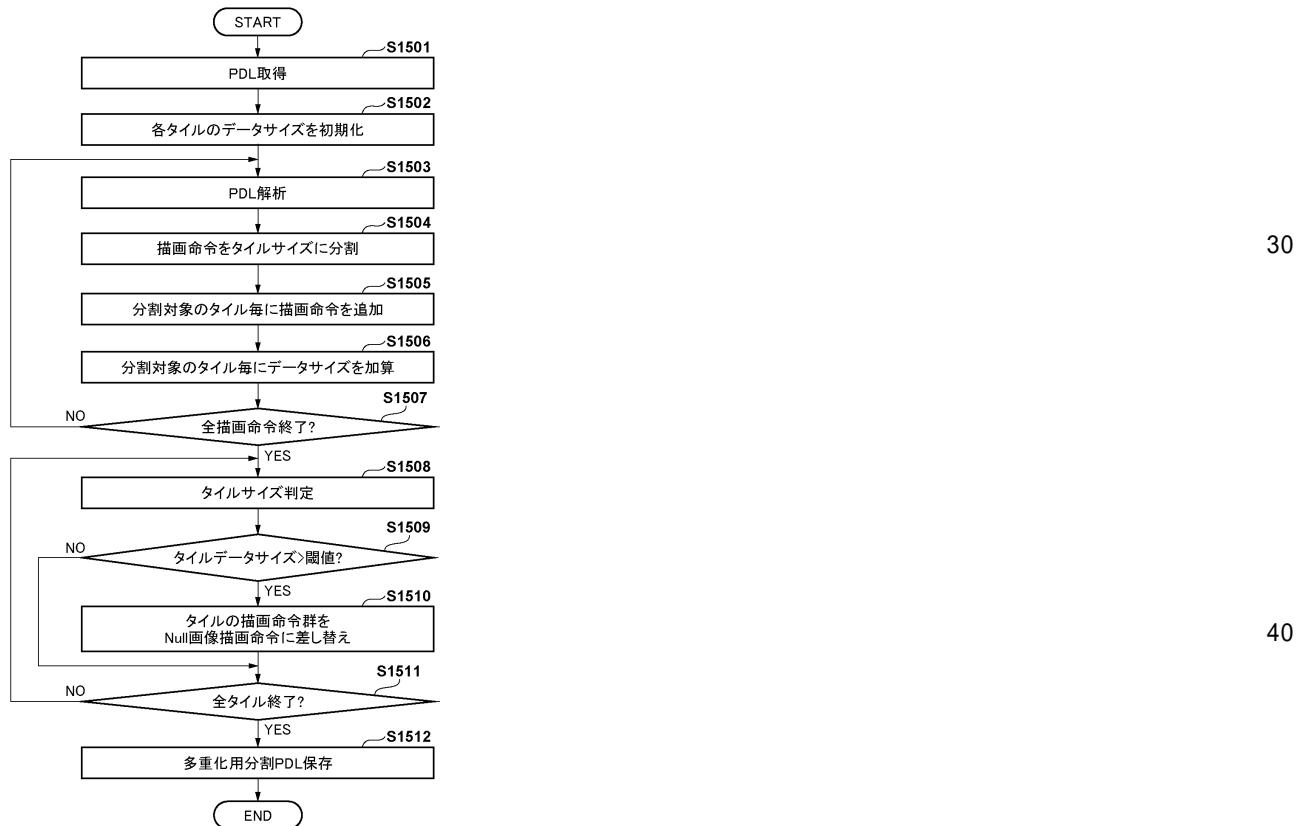
【図14】



10

20

【図15】



30

40

50

フロントページの続き

(56)参考文献 特開2009-199557(JP,A)

(58)調査した分野 (Int.Cl., DB名)

H04N 1/32

H04N 1/409

H04N 1/401

H04N 1/407