

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6501454号  
(P6501454)

(45) 発行日 平成31年4月17日(2019.4.17)

(24) 登録日 平成31年3月29日(2019.3.29)

(51) Int.Cl.	F I					
<b>G09G</b>	<b>5/24</b>	<b>(2006.01)</b>	G09G	5/24	630S	
<b>G09G</b>	<b>5/22</b>	<b>(2006.01)</b>	G09G	5/22	630G	
<b>G09G</b>	<b>5/26</b>	<b>(2006.01)</b>	G09G	5/24	610Z	
<b>B41J</b>	<b>5/30</b>	<b>(2006.01)</b>	G09G	5/26	Z	
<b>G06T</b>	<b>11/60</b>	<b>(2006.01)</b>	B41J	5/30	Z	

請求項の数 7 (全 16 頁) 最終頁に続く

(21) 出願番号 特願2014-89308 (P2014-89308)  
 (22) 出願日 平成26年4月23日(2014.4.23)  
 (65) 公開番号 特開2015-206986 (P2015-206986A)  
 (43) 公開日 平成27年11月19日(2015.11.19)  
 審査請求日 平成29年4月20日(2017.4.20)

(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 100126240  
 弁理士 阿部 琢磨  
 (74) 代理人 100124442  
 弁理士 黒岩 創吾  
 (72) 発明者 河野 匡彦  
 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 橋本 直明

最終頁に続く

(54) 【発明の名称】 情報処理装置、その制御方法、プログラム、及び記憶媒体

(57) 【特許請求の範囲】

【請求項1】

複数のフォントを用いて、アイコンを含む文字列を描画可能な情報処理装置であって、  
 描画処理の対象となる文字列に対応する情報を取得する文字列取得手段と、  
 前記複数のフォントのうち、少なくとも1つの特定のフォントに対して個別に設定され  
 た第1のフォントスタイルを示す情報と、前記文字列のうちアイコンを描画するための指  
 定されるフォントであって、前記特定のフォントに個別に設定された第1のフォントス  
 タイルが適用されないフォントであるアイコンフォントに適用される第2のフォントス  
 タイルを示す情報とを取得する描画情報取得手段と、  
 前記文字列取得手段によって取得された情報に基づいて、前記描画処理の対象となる文  
 字列に含まれる各文字及びアイコンについて、前記複数のフォントのうち、描画に用いる  
 フォントを特定する特定手段と、  
 前記特定手段によって特定されたフォントが前記少なくとも1つの特定のフォントであ  
 る場合に、前記特定されたフォントに対して前記第1のフォントスタイルを反映して前記  
描画処理の対象となる文字列に含まれる文字を描画し、前記特定手段によって特定された  
 フォントが前記アイコンフォントである場合に、前記第2のフォントスタイルを反映して  
 前記描画処理の対象となる文字列に含まれるアイコンを描画する描画手段と、を備えるこ  
 とを特徴とする情報処理装置。

【請求項2】

前記フォントスタイルは、ノーマル、長体、平体、斜体のいずれかでないし複数で構成さ

れることを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】

前記文字列取得手段は、前記文字列を示す情報として文字コードの配列を取得するとともに、前記文字列に対して指定されたフォントを示す情報を取得し、

前記特定手段は、前記取得された文字コードの配列を、前記取得されたフォントを示す情報に基づいて、グリフィンデックス列に変換し、

前記描画手段は、前記グリフィンデックス列に基づいて、前記描画処理の対象となる文字列のうち、同一のフォントで描画される部分毎に、描画処理を実行することを特徴とする請求項 1 または 2 に記載の情報処理装置。

【請求項 4】

前記特定手段は、前記文字列取得手段によって取得された配列に含まれる文字コードに対応する字形が、前記指定されたフォントのデータに存在しない場合、予め定められた優先順に従って、前記複数のフォントのうち、前記指定されたフォントとは異なるフォントを、前記文字コードに対応する文字を描画するフォントとして特定することを特徴とする請求項 3 に記載の情報処理装置。

【請求項 5】

複数のフォントを用いてアイコンを含む文字列を描画可能な情報処理装置の制御方法であって、

文字列取得手段により、描画処理の対象となる文字列に対応する情報を取得する文字列取得工程と、

描画情報取得手段により、前記複数のフォントのうち、少なくとも 1 つの特定のフォントに対して個別に設定された第 1 のフォントスタイルを示す情報と、前記文字列のうちアイコンを描画するための指定されるフォントであって、前記特定のフォントに個別に設定された第 1 のフォントスタイルが適用されないフォントであるアイコンフォントに適用される第 2 のフォントスタイルを示す情報とを取得する描画情報取得工程と、

特定手段により、前記文字列取得工程において取得された情報に基づいて、前記描画処理の対象となる文字列に含まれる各文字及びアイコンについて、前記複数のフォントのうち、描画に用いるフォントを特定する特定工程と、

描画手段により、前記特定工程において特定されたフォントが前記少なくとも 1 つの特定のフォントである場合に、前記特定されたフォントに対して前記第 1 のフォントスタイルを反映して前記描画処理の対象となる文字列に含まれる文字を描画し、前記特定手段によって特定されたフォントが前記アイコンフォントである場合に、前記第 2 のフォントスタイルを反映して前記描画処理の対象となる文字列に含まれるアイコンを描画する描画工程と、

を備えることを特徴とする情報処理装置の制御方法。

【請求項 6】

コンピュータに読み取らせ実行させることによって、前記コンピュータに請求項 5 に記載の制御方法を実行させるプログラム。

【請求項 7】

請求項 6 に記載のプログラムが記憶されたコンピュータが読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はフォントを用いて文字を描画する処理に関する。

【背景技術】

【0002】

従来、指示された文字列を、フォントを用いて描画する技術が知られている。ここでフォントとは、レイアウトの基準となる情報と文字コードに対応するインデックス、インデックスに関連付けられた字形のデータと字形のレイアウトに関する情報を持つデータを差す。従って、指示された文字列の字形をどのようにレイアウトするか、上部・下部にどの

10

20

30

40

50

程度空白を持たせるか、などのデザインは個々のフォントのデザインに従って決定される。

【0003】

同一のフォントを有さない複数の装置では、同じ文字列を、同一の見た目となるように描画することが困難となる。特許文献1では、搭載するフォントが異なるPCとプリンタで、描画結果の見た目の整合性をとるため、PCで決定されたフォントをプリンタが持たない場合には、プリンタ上のフォントのサイズをPCで決定されたフォントに近づくようにスケールする。

【先行技術文献】

【特許文献】

10

【0004】

【特許文献1】特開2012-190461号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

単一のフォントが全ての文字コードに対して字形の情報を持っているとは限らない。従って、任意の入力文字列を単一のフォントで描画しようとしても、全ての文字を描画できない場合がある。このような場合に対応するため、フォントリスト技術が知られている。ここでいうフォントリスト技術では、あるフォントに対して、当該フォントが描画できない文字を代わりに描画させる他の複数のフォントが優先順のリスト状で関連付けられている。つまり、指定されたフォントが特定の文字コードに対応する字形をフォントが含まない場合、そのフォントに関連付けられた他のフォントを優先度の順番で選択し、該当する文字コードの字形を検索し、見つかった字形を用いて描画処理を実行する。

20

【0006】

文字列を描画させる際には、文字サイズを指示したり字形を斜体にしたりなどの見た目の装飾に関する指示を、ある程度まとまった文字列毎に指示することが広く行われている。しかしながら、従来技術では、フォントリストなどにより、複数のフォントが混在した文字列を描画する場合、複数のフォント同士の見た目の整合性を管理することができず、描画結果の視認性が低下するという課題があった。

【0007】

30

そこで本発明では、複数のフォントが混在した文字列のフォント同士の見た目の整合性を管理することを目的とする。

【課題を解決するための手段】

【0008】

本発明は、上記課題を鑑みてなされたものであり、複数のフォントを用いてアイコンを含む文字列を描画可能な情報処理装置であって、描画処理の対象となる文字列に対応する情報を取得する文字列取得手段と、前記複数のフォントのうち、少なくとも1つの特定のフォントに対して個別に設定された第1のフォントスタイルを示す情報と、前記文字列のうちアイコンを描画するための指定されるフォントであって、前記特定のフォントに個別に設定された第1のフォントスタイルが適用されないフォントであるアイコンフォントに適用される第2のフォントスタイルを示す情報とを取得する描画情報取得手段と、前記文字列取得手段によって取得された情報に基づいて、前記描画処理の対象となる文字列に含まれる各文字及びアイコンについて、前記複数のフォントのうち、描画に用いるフォントを特定する特定手段と、前記特定手段によって特定されたフォントが前記少なくとも1つの特定のフォントである場合に、前記特定されたフォントに対して前記第1のフォントスタイルを反映して前記描画処理の対象となる文字列に含まれる文字を描画し、前記特定手段によって特定されたフォントが前記アイコンフォントである場合に、前記第2のフォントスタイルを反映して前記描画処理の対象となる文字列に含まれるアイコンを描画する描画手段と、を備える。

40

【発明の効果】

50

## 【0009】

本発明により、複数のフォントが混在した文字列のフォント同士の見た目の整合性を管理することができる。

## 【図面の簡単な説明】

## 【0010】

【図1】実施例1における情報処理装置を含むシステムの構成の一例を示す模式図。

【図2】実施例1の情報処理装置の処理の流れの一例を示すフローチャート。

【図3】変形例の情報処理装置の処理の流れの一例を示すフローチャート。

【図4】描画設定情報としてフォントスタイルを設定した場合の例を示す図。

【図5】描画設定情報としてベースラインを設定した場合の例を示す図。

【図6】描画設定情報としてフォントサイズあるいはサイズ比率を設定した場合の例を示す図。

## 【発明を実施するための形態】

## 【0011】

以下、本発明の実施形態について、図面を参照して説明する。なお、以下の実施の形態は特許請求の範囲に関する本発明を限定するものではなく、また、本実施の形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。なお、同一な構成については、同じ符号を付して説明する。

## 【0012】

## &lt;実施例1&gt;

本実施例では、複数のフォントが混在した文字列に対して、フォントスタイル、フォントサイズ、レイアウト等見た目を変化させるための情報が設定された場合に、フォント毎に、設定された情報を反映するか否かを管理する例を説明する。

## 【0013】

図1(a)は本実施例に係る情報処理装置100の概要を示した構成図である。CPU(Central Processing Unit)101は、システム制御部であり、システムバス108を介して、接続された各要素を制御する。ROM(Read-Only Memory)102は、CPUの制御プログラムや、情報処理装置100が描画可能なフォントのデータなど各種固定データを格納するものである。RAM(Random Access Memory)103は、SRAM、DRAM等で構成され、プログラム制御変数等を格納するためのものである。また、各種設定パラメータ、各種ワーク用バッファもRAM103に格納されるものである。記憶装置104はハードディスク等で構成され、文書や画像などのファイルデータを格納するためのものである。入力部105は、キーボード、タッチパネル等で構成され、操作者が各種入力操作を行うためのものである。表示部106は、LCD、LED等で操作者に表示通知するためのものである。通信インターフェース107は、ネットワークに接続するためのインターフェースである。接続方法としてはLAN、USB等がある。同図に示した構成は、本発明の実施例に係る情報処理装置の一部を示したものであり、装置によっては、この他、スキャナ部、プリンタ部、モデム、スピーカ等が装置の構成に加わる場合がある。また、各構成要素は、必ずしも装置内で接続される構成に限定されず、各種インターフェースを介して接続された外部装置であっても構わない。

## 【0014】

図1(b)は、情報処理装置100の機能構成の一例を示すブロック図である。各機能部は、CPU101が、ROM102に格納されたプログラムをRAM103に展開し、後述する各フローチャートに従った処理を実行することで実現されている。ただし、本発明は、これらの機能部をハードウェアで実現する情報処理装置によっても同様に実現可能である。以下、各要素について説明する。

## 【0015】

文字列取得部111は、入力部105を介してユーザ操作により入力された指示に従ったり、構造化文書を読み込んで解析したりすることで、表示部106が表示する表示画面

10

20

30

40

50

を構成する要素から、描画対象となる文字列を取得する。また、本実施例では、取得した文字列を描画するために指定されたフォントを指定する情報を取得する。本実施形態では、表示画面の構成要素として、描画すべき文字列が複数存在する場合でも、1つ1つの要素ごとに文字列の情報を取得し、描画処理を行うことを繰り返す。

#### 【0016】

描画情報取得部112は、入力部105を介してユーザ操作により入力された指示に従ったり、構造化文書を読み込んで解析したりすることで、描画する文字列に対して、見た目の変化を指示するために設定された描画設定情報を取得する。さらに、本実施例では、情報処理装置100が描画可能なフォントのうち、少なくとも1つの特定のフォントに対して、個別に設定された描画設定情報を取得する。なお、本実施例において、フォントに個別に設定される描画設定情報は、表示画面の設計段階で予め定められた情報であって、個々の文字列の描画に当該フォントが使用されるか否かに関わらず、設定に従って取得される。ただし、情報処理装置100の操作者によって任意に変更可能な、言語環境などに従って定められてもよい。本実施例で扱う描画設定情報は、フォントスタイル、飾り文字、フォントカラー、フォントサイズ等、文字の見た目を変化させるための情報のうち、必要に応じて設計者が設定したものである。

#### 【0017】

特定部113は、文字列取得部111が取得した文字列の各文字について、描画に用いるフォントを特定する。そして、文字列を文字コードの配列から、各文字の描画に用いるフォントに応じたインデックスの配列に変換する。本実施例では、取得された文字列の全ての文字が、文字列取得部111に取得された情報で指示された通りのフォントで描画できない場合は、フォントリスト技術を用いて、当該文字を描画するフォントを選択する。つまり、本実施例では、ある文字について指定されたフォントが、当該文字に対応する字形を持たない場合、フォントリスト技術を用いて、優先順に他のフォントのデータを参照し、該当する文字の字形を有する他のフォントを特定する。従って、1つの文字列であっても、複数のフォントの字形が混在するように描画される場合がある。このとき、同じ文字であっても、フォントによって、対応する字形のインデックスは一致するとは限らない。例えば文字「A」に対応する字形のインデックスが、あるフォントでは10となり、別のフォントでは100となることがある。そのため、字形のインデックスは用いられるフォントの情報と合わせて管理される。以下、生成した字形のインデックスの配列をグリフィンデックス列、グリフィンデックス列とそれに対応するフォント情報を合わせてグリフィンデックス列情報と表記する。

#### 【0018】

描画部114は、変換された配列情報に基づいて、文字列のうち、同一のフォントで描画される部分毎に、描画設定情報を反映した描画を繰り返し行うことで、文字列全体の描画処理を実行する。特に本実施例では、描画に用いるフォントに個別に描画設定情報が設定されている文字（部分配列）は、文字列全体に設定されたフォントに依存しない描画設定情報を反映せずに、フォントに個別に設定された描画設定情報を反映して描画する。一方で、描画に用いるフォントに個別に描画設定情報が設定されていない文字（部分配列）は、文字列全体に設定されたフォントに依存しない描画設定情報を反映して描画する。つまり、描画対象の文字が、描画情報取得部112によって、個別に設定された描画設定情報が取得された少なくとも1つの特定のフォントであるか否かに応じて、描画の際に反映する描画設定情報を変える。判定部115は、グリフィンデックス列情報と描画情報取得部112が取得した情報とに基づいて、描画対象とされている部分配列に、当該フォントに個別に描画設定情報が設定されているか否かを判定し、描画部114の処理を切り替えさせる。

#### 【0019】

保持部116は、描画部114が、個別に描画設定情報が設定されたフォントの描画を行っている間、文字列全体に設定されたフォントに依存しない描画設定情報文字列全体に設定されたフォントに依存しない描画設定情報を保持する。

## 【 0 0 2 0 】

以下、図2のフローチャートに沿って実施例の動作を説明する。本実施例では、表示部106に表示される画面において画面遷移、または更新が実行されるトリガーとなるイベントが発生したことに応じて、文字列の描画処理が呼び出される。図2のフローチャートは、遷移後（更新後）の画面を表示するために描画すべき文字列の情報が、所定の単位毎（例えば、個々のGUI部品に描画される文字列毎）にアプリケーション等から通知される度に実行される。

## 【 0 0 2 1 】

ステップS201では、描画情報取得部112が、情報処理装置100が描画可能なフォントのうち少なくとも1つの特定のフォントに対して、見た目の変化を指示するために設定された描画設定情報を取得する。本実施例では、描画設定情報の一例として、フォントスタイルの指示を取得する。フォントスタイルとは、フォントを使って文字を描画する際に指示することが可能な、見た目の変形をいう。第1実施例の情報処理装置100では、フォントスタイルとして、通常の（見た目の変化が無い状態の）フォントより字形の幅を細くする長体、通常のフォントより字形の幅を太くする平体、字形を傾ける斜体などを指示することが可能である。フォントスタイルは、数値や名称を示す文字列といった定数の指定を受けることで決定される。予め、ROMなどの記憶手段において、通常の字形をもったフォントに対して、フォントスタイルに合わせて変形された字形をもったフォントが対応付けられていることがある。このような1以上のフォントのまとまりはフォントファミリーと呼ばれる。指示されたフォントスタイルに対応するフォントがフォントファミリーにある場合、そのフォントが描画に使用される。一方で、フォントスタイルに対応するフォントがフォントファミリーに存在しない場合には、フォントスタイルに応じた変形処理を、通常のフォントの字形に対して行い描画することもある。例えば平体のフォントスタイルが決定された場合、通常の字形を水平方向に拡大もしくは垂直方向に縮小して描画を行う。例えば長体のフォントスタイルが決定された場合、通常の字形を水平方向に縮小もしくは垂直方向に拡大して描画を行う。拡大率・縮小率は描画環境に応じて定められる。拡大率・縮小率はレイアウトや、文字列を入力するユーザの決定に応じて動的に変更してもよい。一例として、実施例1の情報処理装置100では、長体のフォントスタイルが決定された場合、水平方向の縮小率が0.5で描画される。

## 【 0 0 2 2 】

なお、描画設定情報として取得されるのは、フォントスタイルに限らない。例えば、文字のベースラインの位置やフォントのサイズのように、字形をレイアウトする基準を絶対値で決定する情報も、ステップS201において、描画設定情報として取得される。

## 【 0 0 2 3 】

ステップS202では、描画情報取得部112が、描画すべき表示画面において設定されている、アイコンに依存しない描画設定情報を取得する。そして、取得した描画設定情報を、描画コンテキストに設定する。

## 【 0 0 2 4 】

ステップS203では、文字列取得部111が、描画する文字列の各文字を示す情報と、指定されたフォントを示す情報を取得する。具体的には、文字コードの配列と、それらを描画するために指定された1以上のフォントを示す情報を取得する。

## 【 0 0 2 5 】

ステップS204では、特定部113が、取得した文字列の各文字を描画するフォントを特定し、文字列の文字コードの配列から、特定されたフォントのデータに格納された字形のインデックスの配列に変換する。具体的には、ROM102に記憶された複数のフォントの情報のうち、ステップS203で情報を取得したフォントの情報を参照して、取得した文字列の各文字の文字コードに対応するインデックスを探索する。文字コードに対応するインデックスがある場合、指定されたフォントを、描画に用いるフォントとして特定し、グリフィンデックス列情報を生成する。文字コードに対応するインデックスが存在しない場合、フォントリスト技術を用いて、優先順に他のフォントのデータを参照し、該当

10

20

30

40

50

する文字コードについての字形を有する他のフォントからインデックスを得る。これにより、全ての文字について、いずれかのフォントを特定し、特定されたフォントに応じたインデックスからなるグリフィンデックス列を生成することができる。いずれのフォントのデータにおいても字形が見つからない場合は、当該文字は、ミッシンググリフとして、不明な文字用にフォントで定義された字形のインデックスを使用して、グリフィンデックス列が生成される。なお本実施例では、指定されたフォントの代替として用いるフォントを決定する方法として、フォントリスト技術を用いる例を説明したが、これに限らない。例えば、予め、特定の文字コードに対しては、特定のフォントから字形を取得するように設定しておくこともできる。なお、描画する文字列が定型文である場合、文字列取得部 111 は、文字列を文字コードの配列ではなく、数値や文字列で表現される定数として、取得する場合もある。その場合も、ROM 102 に記憶された情報を参照して、取得した定数に対応するグリフィンデックス列情報を生成する。ただし、参照先は ROM 102 に限らず、記憶装置 104 が保持するフォントを参照してもよいし、通信インターフェース 107 経由で、外部装置からフォントデータを取得し、RAM 103 に展開して参照してもよい。

10

**【0026】**

また、描画環境がアラビア語などの右から左に描画する文字列が含まれる言語向けである場合や、受け取った文字列の一部にアラビア語が含まれるなどの場合には、文字列の論理的な順序と、描画する順序が異なる場合がある。こうした場合、特定部 113 は、Unicode Standard Annex #9 (<http://www.unicode.org/reports/tr9/tr9-29.html>) の仕様に従い、まず文字を、描画する順序に並び替える。そして、指定されたフォントの情報を参照して、並び替え後の文字列をグリフィンデックス列情報へ変換する。なお本実施例では、横書きの文字列は、より左側に配置される文字から順に描画し、縦書きの文字列は、より上側に配置される文字から順に描画を行う。

20

**【0027】**

ステップ S205 では、描画部 114 が、グリフィンデックス列から、描画順に沿って、同一のフォントで描画される部分配列を選択し、描画処理の処理対象とする。

**【0028】**

ステップ S206 では、判定部 115 が、グリフィンデックス列情報に基づいて、選択した部分配列に相当する文字を描画するフォントを特定し、特定したフォントに対して個別に描画設定情報が決定されているかどうかを判定する。選択した部分配列に相当する文字を描画するフォントに、個別に描画設定情報が決定されている場合（ステップ S206 で YES）、ステップ S207 に進む。一方、選択した部分配列に相当する文字を描画するフォントに、個別に描画設定情報が決定されていない場合（ステップ S206 で NO）は、ステップ S209 に進む。ステップ S207 において、描画部 114 は、描画コンテキストにセットされた描画設定情報を、保持部 116 に保持させる。つまり、フォントに個別に設定された描画設定情報を用いて描画をする間、フォントに依存せず文字列全体に対して設定された描画設定情報の情報を退避させる。

30

**【0029】**

ステップ S208 において、描画部 114 は、選択した部分配列に相当する文字を描画するフォントに個別に設定された描画設定情報を取得し、描画コンテキストの描画設定情報を更新する。ステップ S206 ~ ステップ S208 の処理により、フォントに個別に描画設定情報が設定されている場合は当該描画設定情報が、フォントに個別に描画設定情報が設定されていない場合は文字列全体に設定された描画設定情報が、描画コンテキストに設定される。

40

**【0030】**

ステップ S209 では、描画部 114 が、描画コンテキストに設定された描画設定情報を反映するように、部分配列に対応する文字列を描画する。本実施例では、フォントスタイルに応じた変形や変化が施された文字が描画されることとなる。

50

## 【0031】

ステップS210では、判定部115が再び、選択されている部分配列に対応する文字を描画するフォントに、個別に設定された描画設定情報があるか否かを判定する。設定されていると判定された場合（ステップS210でYES）、ステップS211に進む。一方、設定されていないと判定された場合（ステップS210でNO）、ステップS212に進む。ステップS211では、描画部114が、保持部116に保持されていた描画設定情報を読み出して、描画コンテキストに設定する。なお、ステップS210では、描画部114が、保持部116に描画設定情報が保持されているか、すなわち、ステップS207において退避させた描画設定情報があるかを判定しても構わない。この場合は、ステップS211において、描画コンテキストの設定を更新した段階で、保持部116が保持していた情報を破棄（初期化）する処理を加える。

10

## 【0032】

そして、ステップS212では、描画部114が、グリフィンデックス列情報に基づいて、文字列全体の描画が終了したかを判定する。終了していないと判定される場合（ステップS212でNO）は、ステップS205に戻り、描画順に沿って順次部分配列を選択する。一方、終了したと判定された場合（ステップS212でYES）は、処理を終了する。

## 【0033】

ここで、図4を参照して、本実施例の処理に従って描画される文字列の具体例を説明する。以下では、GUI（Graphical User Interface）のための表示画面が描画される場面を想定し、図2のフローチャートが1回実行されることで、1つのGUI部品に対応する文字列の描画が行われるとする。例えば、GUI部品である1つのボタンに、「Quality」という英単語と、扇形の図形のアイコンとが描画される例を説明する。

20

## 【0034】

まず、従来のように、複数のフォントが混在することを考慮せず、文字列フォントスタイルの適応した場合の描画結果の例を示す。文字列を構成する全ての文字を、「ノーマル」のフォントスタイルで描画した場合、描画結果は図4(c)のようになる。例えば、GUI画面の設計者が、ボタンオブジェクトをデザインするにあたり、文字列を描画する幅をより狭くしたいと考え、この文字列に対して「長体」のフォントスタイルを設定したとする。この指示に従い、全ての文字を長体のフォントスタイルで描画した場合、通常 of 字形に対して水平方向に0.5の縮小率が適用され、描画結果は図4(a)のようになる。この時、複数のフォントが混在することを考慮しなければ、扇形の図形のアイコンに対しても「長体」のフォントスタイルが適応され、扇形が歪んでしまう。多くの場合、絵文字など図形は、フォントスタイルを適応すると、異なる図形との識別能力が低下してしまうなど、視認性が低下する。しかしながら、アイコンの部分のフォントスタイルを後から解除したり、英単語の部分とアイコン部分とを別の文字列として、それぞれにフォントスタイルを設定したり、するのは設計者にとって煩わしい作業となる。

30

## 【0035】

そこで、本実施例では、予めアイコンフォント自体に描画設定情報を設定することで、アイコンを含む文字列に対してどのようなフォントスタイルが指示された場面であっても、アイコン自体の見た目が常に同一となることを可能とする。例えば、アイコンフォントに対して個別に「ノーマル」（変形なし）を示すフォントスタイルを設定する。

40

## 【0036】

上述した具体例に、本実施例を適応した場合に図2のフローチャートの各ステップで実行される処理を説明する。まず、画面遷移により、GUI画面の描画が指示されたことに応じて、画面の設計情報が解析され、該当するボタンオブジェクトの文字列の情報が通知される。まず、ステップS201において、描画情報取得部112によって、アイコンフォントに対して設定された「ノーマル」を示すフォントスタイルが取得される。なお、ここでは個別の設定が行われるフォントはアイコンフォントのみとするが、他にも個別設定

50

がなされたフォントの情報を取得しても構わない。次に、ステップS202において、文字列全体に対して設定された「長体」のフォントスタイルが取得され、描画コンテキストに設定される。次に、ステップS203において、文字列取得部111が、英語表記の描画用にデザインされたフォントでの描画が指定された「Quality」という文字列と、アイコンフォントでの描画が指定された扇形の図形アイコンの文字列の文字コードを取得する。なお、以下、英語表記の描画用にデザインされたフォントを単に英語フォントという。次に、ステップS204において、特定部113が、取得された文字コードの配列を、グリフィンデックス列に変換する。ここで配列は、英語フォントのグリフィンデックス列情報と、アイコンフォントのグリフィンデックス列情報とで構成される。

#### 【0037】

次にステップS205では、描画部114により、描画順に従い、英語フォントのグリフィンデックス列情報（「Quality」部分の文字列）が処理対象の部分配列として選択される。英語フォントには個別の描画設定情報は設定されていないので、ステップS209において、描画部114により、「長体」のフォントスタイルを反映した描画が実行される。描画コンテキストの更新はなく、文字列の描画が終了していないことから、再びステップS205で、描画部114により、アイコンフォントのグリフィンデックス列情報（扇形の図形部分の文字列）が処理対象の部分配列として選択される。ここでアイコンフォントには、個別に「ノーマル」のフォントスタイルが設定されている。従って、ステップS207において、「長体」のフォントスタイルの情報が保持部116に退避され、ステップS208において、描画コンテキストの描画設定情報が「ノーマル」に更新される。ステップS209では、描画部114により、「ノーマル」のフォントスタイルを反映した描画が実行される。ステップS211において、退避されていた「長体」のフォントスタイルが再び描画コンテキストに設定され、取得された文字列全体の描画が終了したことから、ステップS212において、YESと判定され、一連の処理が終了する。以上の処理による描画結果は図4(b)のようになる。個々の文字列に対して設定される描画設定情報によらず、アイコンフォントで描画される文字は、常にノーマルのフォントスタイルで描画することが可能になる。

#### 【0038】

次に、図5を参照して、描画設定情報として文字のベースラインを設定し、複数のフォントが混在する文字列の見た目の整合性を管理する具体例を説明する。ベースラインとは、文字列を構成する字形を並べる際に、配置の基準とする線のことである。ベースラインに対して、字形をどのように配置するかは、通常、フォントに応じて異なるデザインがなされている。例えば日本語フォントで横書きの文字列を描画する場合は、漢字がベースラインより上にくるようにデザインされることが多い。一方で、中国語フォントでは、漢字の下側の一部が、ベースラインを越えるようにデザインされていることがある。このとき、図5のように、日本語フォントが指定された「東京」という文字列と、中国語フォントが指定された「香港」の文字列を描画することを想定する。従来のように、日本語フォントと中国語フォントのグリフィンデックス列が混在することを考慮しなければ、描画結果は図5(a)のようになる。図5(a)では、各フォントについて、ベースラインと文字との位置関係は、デザインされた条件に従っているが、一見すれば、漢字が上下にずれ、見た目の整合性が損なわれたように見えてしまう。

#### 【0039】

そこで本実施例では、予め、中国語フォントに対して、個別に設定する描画設定情報として、ベースラインの位置を設定決定することで、このような課題を解決できる。すなわち、中国語フォントの漢字がベースラインを上をにずらすように指示する情報を設定する。これにより、描画結果は図5(b)のようになる。これにより、日本語フォントと中国語フォントが混在する文字列であっても、見た目の整合性を損なわずに描画することができる。漢字の並びが揃って見えるので、閲覧者にとっては視認性が低下せず、読みやすい文字列を描画することができる。なお、ベースラインをずらす量は、設計者が予め任意に決定した値を、取得することができる。さらには、隣接して描画される2つのフォントのデ

10

20

30

40

50

ータを解析することにより、自動的に、特定のフォントのベースラインの位置を算出することもできる。例えばフォントのデータから、字形を配置する際に、ベースラインより下に出る部分の長さを取得し、その分だけ上にずらしてもよい。また、フォントに含まれる字形データを全て取得し、字形が全てベースラインより上に来るようにベースラインのずらす量の値を算出することもできる。このように、算出により、特定のフォントに個別に設定される描画設定情報を取得することで、設計者の作業を簡略化させることができる。

#### 【0040】

次に、図6(a)及び(b)を参照して、描画設定情報として文字の描画サイズを設定し、複数のフォントが混在する文字列の見た目の整合性を管理する具体例を説明する。描画部114は、字形データに対して、指定を受けたフォントサイズに応じた拡縮を行い描画する。拡縮の基準としては、フォントが持つunitsPerEMやmax-heightの値が使用される。unitsPerEMはデザインの基準となる数値、max-heightはフォントに含まれる字形の最大高さの値である。例えば字形データがベクトルデータで、サイズ20で描画される場合、ベクトルデータがもつ値をunitsPerEMで割り、20を掛けた結果が実際の描画に使用する値になる。unitsPerEMの代わりにmax-heightで割ってもよい。ただし、unitsPerEMに対してどのように字形をデザインするかはフォントのデザイナーに任されている。例えば英語表記を描画する為のフォントでは、アルファベットの大文字の高さがunitsPerEMに近くなるようにデザインされる。一方で、アラビア語表記を描画する為のフォントでは、一般的な文字の上側に装飾文字と呼ばれる部分が存在する場合があるために、多くの字形の高さはunitsPerEMの半分程度になることもある。ここで従来のように、文字列に複数のフォントが混在することを考慮せずに、文字列全体に一律の文字サイズを設定した場合を例示する。英語とアラビア語の表記をどちらも含む文字列を、同じフォントサイズの「18」で描画した結果は図6(a)のようになり、英語に対してアラビア語の見た目が小さく見えてしまう。これに対し、本実施例では、予め、アラビア語フォントに個別に描画設定情報としてフォントサイズを「32」と設定しておくことが可能となる。その上で本実施例を適応した場合の描画結果は図6(b)のようになり、英語フォントに対するアラビア語フォントの見た目のバランスを向上させることができる。ここで、一般的に使用頻度がより高い英語表記では、アルファベットには様々な装飾を加えて描画することが求められる。従って、比較的使用頻度が低いアラビア語フォントに対して、個別の設定を行うことで、使用頻度の高い言語のフォントに対する自由度を確保した上で、より効率的な画面設計が可能となる。ただし、英語フォントに対して個別の設定を行うことももちろん可能である。

#### 【0041】

さらに、上記の例で個別に設定したフォントサイズ「32」は、画面の設計者が任意に決定することを想定したが、フォントや字形の情報から自動的に算出し、算出結果を描画設定情報として決定してもよい。例えば、英語フォントのunitsPerEMとアラビア語フォントのmax-heightが同じになるように、字形データを比較して、設定すべきフォントサイズを求めてもよい。また、フォントが持つベースラインに対する高さのひとつであるascendのunitsPerEMに対する比率が同じになるようにフォントサイズを算出してもよい。また、英語フォントとアラビア語フォントで同じ文字に対する字形がないか検索し、見つかった場合はその字形の外接矩形が同じ大きさになるようにフォントサイズを算出してもよい。

#### 【0042】

以上説明したように、本実施例によれば、複数のフォントが混在する文字列を描画させる場合に、文字の変化を指示する情報の設定作業の手間数を低減させつつ、複数のフォントの見た目の整合性を管理することができる。それにより、描画結果の文字列の視認性を低下させず、様々な装飾を施した文字列をデザインすることが可能となる。

#### 【0043】

<変形例>

10

20

30

40

50

実施例 1 では、描画設定情報を定数または絶対値で設定する方法について説明した。その変形例として、描画設定情報を相対値で決定する例について説明する。

【 0 0 4 4 】

変形例にかかる情報処理装置 1 0 0 のハードウェア構成、および機能構成は実施例 1 に準じるため詳細説明を省略する。ただし、変形例において、描画情報取得部 1 1 2 は、相対値を含む描画設定情報を取得する。また、判定部 1 1 5 は、保持部 1 1 6 に保持した描画設定情報を累積するか否かを判定し、描画部 1 1 4 の描画を制御する。

【 0 0 4 5 】

以下、図 3 のフローチャートに沿って実施例の動作を説明する。なお、図 2 のフローチャートと同じ処理を行うステップには同番号を付して適宜説明を省略し、実施例 1 と異なる点を中心に簡潔に説明する。

【 0 0 4 6 】

ステップ S 2 0 1 a において、描画情報取得部 1 1 2 は、特定のフォントに対して個別に設定された描画設定情報を取得する。ただし、変形例では、ステップ S 2 0 1 において取得する描画設定情報に、相対値が含まれる。例えば、アラビア語フォントについて個別に設定された描画設定情報として、デフォルトの英語フォントに設定されたフォントサイズに対するサイズ倍率を取得する。上述したように、英語フォントではアルファベットの大文字の高さが `units Per EM` に近くなるようにデザインされるのに対し、アラビア語フォントでは装飾文字が存在するため、多くの字形の高さは `units Per EM` の半分程度にデザインされている。そのため、英語フォントとアラビア語フォントを同じサイズで描画するとアラビア語が図 6 ( a ) のように小さく見えるので、例えば設計時に、アラビア語フォントのサイズ倍率を 2 . 0 に設定する。この場合、英語フォントに対して設定されたフォントサイズの値を 2 . 0 倍した値が、アラビア語フォントに個別に設定された描画設定情報としての、フォントサイズとなる。なお、ここでは一例として、浮動小数点数値を一つ決定するが、X 方向と Y 方向で異なる倍率を決定してもよい。

なお、一例としてサイズ倍率を 2 . 0 としたが、描画情報取得部 1 1 2 が、フォントや字形の情報から自動的に算出し、算出結果を相対値としてもよい。例えば、英語フォントとアラビア語フォントで同じ文字に対する字形がないか検索し、見つかった場合はその字形の外接矩形が同じ大きさになるようにサイズ倍率を算出する。また、フォントが持つベースラインに対する高さのひとつである `ascend` の `units Per EM` に対する比率が同じになるようにサイズ倍率を算出してもよい。

【 0 0 4 7 】

ステップ S 2 0 2 a では、描画情報取得部 1 1 2 が、描画すべき表示画面において設定されている、アイコンに依存しない描画設定情報を取得する。変形例では、ステップ S 2 0 2 a においても、取得される描画設定情報に、相対値が含まれる。そして、取得した描画設定情報と相対値とに基づいて定まるを、描画設定情報を描画コンテキストに設定する。例えば、サイズ倍率 1 . 0 が取得されたとすると、アラビア語以外の文字は、指定されたフォントサイズのまま描画される。

【 0 0 4 8 】

ステップ S 2 0 3 a では、文字列取得部 1 1 1 が、描画する文字列の各文字を示す情報と、指定されたフォントを示す情報、及び相対値で示された描画設定情報に対する基準値を取得する。具体的には、文字コードの配列と、それらを描画するために指定された 1 以上のフォントを示す情報、及び文字列に対して指定されたフォントサイズ等の絶対値を取得する。例えば、文字を入力したユーザが文字列に、英語フォントとフォントサイズ「 1 8 」を設定した場合、指定されたフォントは英語フォント、基準値は「 1 8 」となる。ステップ S 2 0 4 ~ ステップ S 2 0 7 の処理は実施例 1 と同内容であるため、詳細な説明を省略する。

【 0 0 4 9 】

変形例では、ステップ S 2 0 7 において、描画部 1 1 4 が、描画コンテキストにセットされた描画設定情報を、保持部 1 1 6 に保持させると、処理はステップ S 3 0 1 に進む。

10

20

30

40

50

## 【 0 0 5 0 】

ステップ S 3 0 1 では、判定部 1 1 5 が、描画設定情報を累積して用いるかを判定する。つまり、この時点で描画コンテキストに設定されている描画設定情報に含まれる相対値と、特定のフォントに個別に設定された相対値を累積した値を、描画に反映するか否かを判断する。本変形例では、表示画面の設計時に予め、累積して用いることを前提とした相対値が設定されているか、累積しないで用いる相対値が設定されているかを示す情報がフラグ等で用意されており、判定部 1 1 5 はそれに基づいて判定を行う。また、設計時の決定結果に限らず、情報処理装置 1 0 0 のユーザによる設定に応じて切り替えられても良い。累積すると判定された場合（ステップ S 3 0 1 で YES）には、ステップ S 3 0 2 に進み、累積しないと判定された場合（ステップ S 3 0 1 で NO）には、ステップ S 2 0 8 に進む。ステップ S 2 0 8 以降の処理は実施例 1 と共通するため説明を省略する。

10

## 【 0 0 5 1 】

ステップ S 3 0 2 に進んだ場合、描画部 1 1 4 は、描画コンテキストに設定されていた相対値と、選択されている部分配列に対応するフォントの個別の相対値とを累積する。なお、相対値が積によって累積されるか、和によって累積されるかは描画設定情報の種類や描画環境、デザインの方針、ユーザの指示などにより予め設定された方法に従う。ここでは一例として、アラビア語フォントを描画する場合には、ステップ S 2 0 2 a で取得されたサイズ倍率の 1 . 0 に対し、アラビア語フォントのサイズ倍率である 2 . 0 が積算され、累積値は 2 . 0 となるとする。ステップ S 3 0 3 において、描画部 1 1 4 は、累積した相対値を含む描画設定情報（選択されている部分配列に対応するフォントの描画設定情報）を描画コンテキストに設定する。そして、ステップ S 2 0 9 に進む。

20

## 【 0 0 5 2 】

ステップ S 2 0 9 では、描画部 1 1 4 によって、文字列に対して指定されたフォントサイズに対し、相対値が反映されたフォントサイズで、文字の描画が実行される。

## 【 0 0 5 3 】

ここまで説明している具体例では、英語フォントのサイズ倍率は 1 . 0 で、アラビア語フォントのサイズ倍率は 2 . 0 である。従って、例えば、ユーザが文字列に、英語フォントとフォントサイズ「 1 8 」が設定された場合、アラビア語フォントは 2 . 0 倍されたフォントサイズ「 3 6 」で描画される。描画した結果は図 6 ( c ) のようになり、従来例の図 6 ( a ) に対しては、英語フォントに対するアラビア語フォントの見た目のバランスが向上するという効果が得られる。以降の処理は実施例 1 と共通するため、詳細な説明を省略する。

30

## 【 0 0 5 4 】

以上説明したように、変形例においても、複数のフォントが混在する文字列を描画させる場合に、文字の装飾を指示する作業工数を低減させつつ、複数のフォントの見た目の整合性を管理することができる。それにより、描画結果の文字列の視認性を低下させず、様々な装飾を施した文字列をデザインすることが可能となる。

## 【 0 0 5 5 】

なお、上述した変形例では、フォントに依存しない描画設定情報も、相対値で示された情報を含むとしたが、これに限らない。例えば、フォントに依存しない描画設定情報は絶対値として取得され、その絶対値を、ステップ S 2 0 3 a で取得される基準値に相当する値として扱うことも可能である。

40

## 【 0 0 5 6 】

なお、描画設定情報を定数または絶対値で設定する実施例 1 と、描画設定情報に相対値を用いる実施例 1 の変形例の処理は、排他的に実施されることもできるが、組み合わせることもできる。例えば、描画設定情報のうち、フォントスタイルは絶対値、フォントサイズについては相対値で示された値に基づく処理を行う。また例えば、フォント毎に、絶対値あるいは相対値のいずれが適切な値で描画設定情報が設定されてもよい。

## 【 0 0 5 7 】

50

< その他の実施例 >

本発明の目的は前述した実施例の機能を実現するソフトウェアの文字描画プログラムコードを記録した記録媒体を、システムあるいは装置に供給してもよい。そのシステムあるいは装置のコンピュータ（またはCPUまたはMPU）が記録媒体に格納された文字描画プログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

この場合、記憶媒体から読み出された文字描画プログラムコード自体が前述した実施形態の機能を実現することとなり、その文字描画プログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0058】

文字描画プログラムコードを供給するための記憶媒体としては、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、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROM、DVDなどを用いることができる。

また、コンピュータが読み出した文字描画プログラムコードを実行することにより、前述した実施例の機能が実現されてもよい。更にその文字描画プログラムコードの指示に基づき、コンピュータ上で稼動しているOperating System（OS）などが実際の処理の一部または全部を行ってもよい。その処理によって前述した実施例の機能が実現される場合も含まれることは言うまでもない。

【0059】

さらに、記憶媒体から読み出された文字描画プログラムコードが、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。その後、その文字描画プログラムコードの指示に基づき、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い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【符号の説明】

【0060】

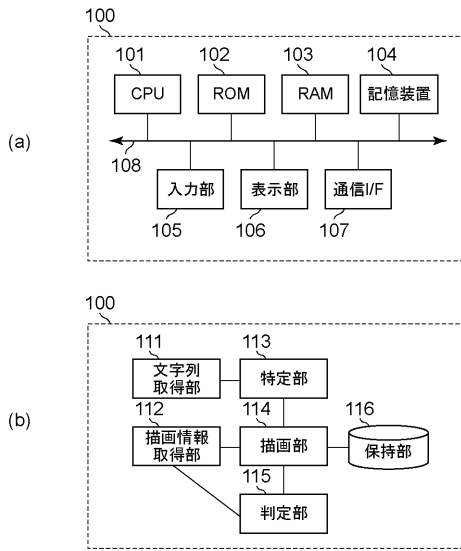
- 111 文字列取得部
- 112 描画情報取得部
- 113 特定部
- 114 描画部
- 115 判定部
- 116 保持部

10

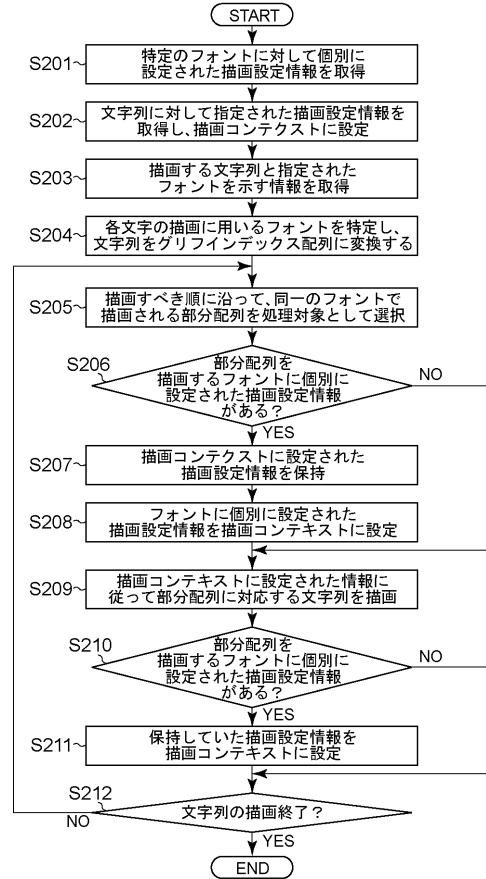
20

30

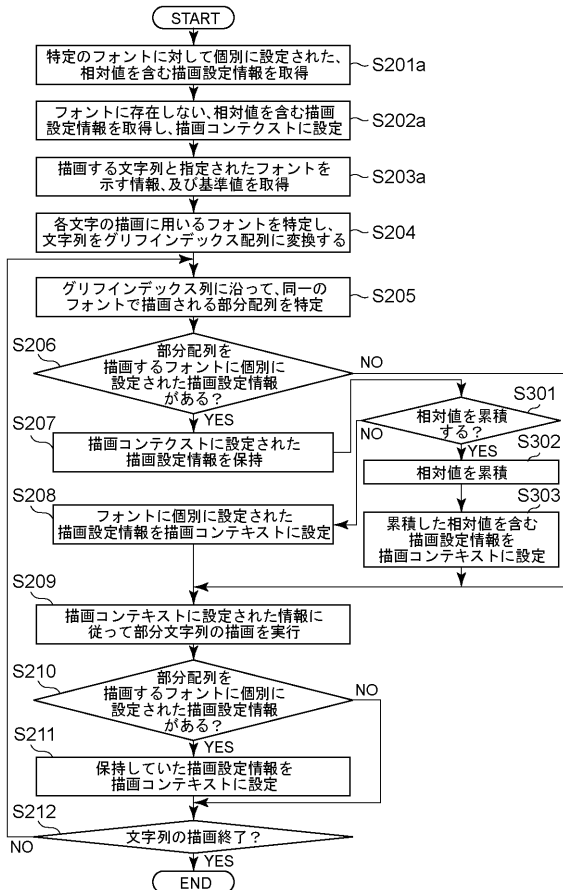
【図1】



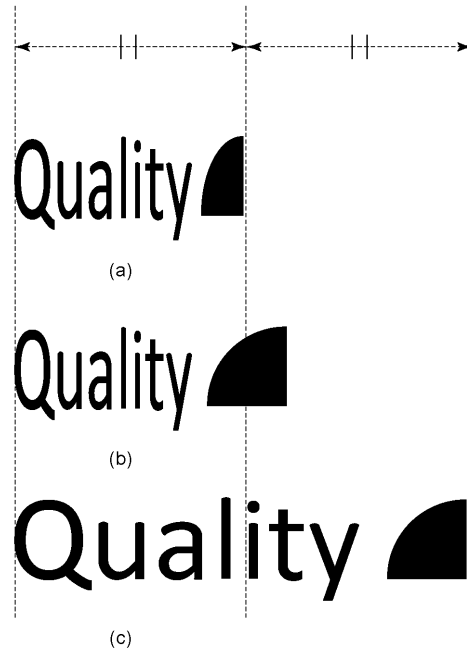
【図2】



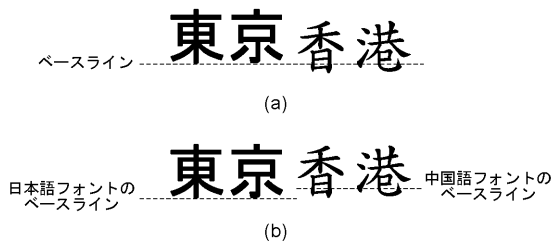
【図3】



【図4】



【図5】



【図6】



---

 フロントページの続き

(51)Int.Cl. F I  
 H 0 4 N 1/387 (2006.01) G 0 6 T 11/60 1 0 0 A  
 H 0 4 N 1/387

(56)参考文献 特開平05 - 177901 (JP, A)  
 特開平04 - 044160 (JP, A)  
 特開2007 - 240848 (JP, A)  
 特開平06 - 325035 (JP, A)  
 特開平07 - 191659 (JP, A)  
 特開平04 - 127270 (JP, A)  
 特開2004 - 126311 (JP, A)  
 特開2006 - 195767 (JP, A)  
 特開2011 - 073336 (JP, A)  
 特開平05 - 193085 (JP, A)  
 特開2007 - 245663 (JP, A)  
 特開平06 - 024084 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G 0 9 G 5 / 2 4  
 B 4 1 J 5 / 3 0  
 G 0 6 T 1 1 / 6 0  
 G 0 9 G 5 / 2 2  
 G 0 9 G 5 / 2 6  
 H 0 4 N 1 / 3 8 7