

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3832976号
(P3832976)

(45) 発行日 平成18年10月11日(2006.10.11)

(24) 登録日 平成18年7月28日(2006.7.28)

(51) Int. Cl.		F I		
G06F 12/00	(2006.01)	G06F 12/00	520J	
G06F 17/30	(2006.01)	G06F 17/30	170B	
G06T 1/00	(2006.01)	G06T 1/00	200D	

請求項の数 13 (全 20 頁)

(21) 出願番号	特願平10-245105	(73) 特許権者	000001007
(22) 出願日	平成10年8月31日(1998.8.31)		キヤノン株式会社
(65) 公開番号	特開2000-76120(P2000-76120A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成12年3月14日(2000.3.14)	(74) 代理人	100076428
審査請求日	平成16年5月28日(2004.5.28)		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	100093908
			弁理士 松本 研一
		(74) 代理人	100101306
			弁理士 丸山 幸雄

最終頁に続く

(54) 【発明の名称】 画像ファイル管理方法および装置

(57) 【特許請求の範囲】

【請求項1】

複数枚の画像を記憶するための画像ファイル管理方法であって、

前記複数枚の画像の各々の画像データを区切り情報を挟んで連続して記憶する画像データ領域と、前記各々の画像データの前記画像データ領域における格納位置を示す位置情報を含む管理情報を格納する管理情報領域とを含む画像ファイルに関して、前記画像データ領域中において対応する位置情報が存在しない画像データを無効な画像データとして検出する検出工程と、

前記検出工程によって検出された無効な画像データの前記画像データ領域における格納位置を当該無効な画像データに隣接する区切り情報に基づいて取得し、取得した格納位置を示す位置情報を生成する生成工程と、

前記生成工程で生成された位置情報を前記管理情報領域に記録して前記無効な画像データを復活させる復活工程とを備えることを特徴とする画像ファイル管理方法。

【請求項2】

前記位置情報は、前記画像データ領域における前記複数枚の画像の各々の先頭位置を示すことを特徴とする請求項1に記載の画像ファイル管理方法。

【請求項3】

前記位置情報は、前記画像データ領域における前記複数枚の画像の各々のデータ量を示すことを特徴とする請求項1に記載の画像ファイル管理方法。

【請求項4】

前記位置情報は、前記画像データ領域における前記複数枚の画像の各々の先頭位置とデータ量を示すことを特徴とする請求項 1 に記載の画像ファイル管理方法。

【請求項 5】

前記検出工程は、

ある画像データの先頭位置とデータ量から求まる次の画像データの先頭位置と、当該次の画像データに対応する位置情報が表す先頭位置とが一致しない場合に無効化された画像データがあると判断する判断工程を備え、

前記判断工程において無効化された画像データがあると判断された場合は、前記画像データ領域の、当該画像データの位置情報が示す先頭位置から前記次の画像データの位置情報が示す先頭位置までの間を調査して検出された区切り情報に基づいて無効な画像データを検出することを特徴とする請求項 4 に記載の画像ファイル管理方法。

10

【請求項 6】

前記管理情報は、前記画像データ領域に含まれる有効な画像データの数を示す画像数情報を含み、

前記復活手段によって復活した画像の数に基づいて前記画像数情報を更新する更新手段を更に備えることを特徴とする請求項 1 に記載の画像ファイル管理方法。

【請求項 7】

複数枚の画像を記憶するための画像ファイル管理装置であって、

前記複数枚の画像の各々の画像データを区切り情報を挟んで連続して記憶する画像データ領域と、前記各々の画像データの前記画像データ領域における格納位置を示す位置情報を含む管理情報を格納する管理情報領域とを含む画像ファイルを格納する格納手段と、

20

前記画像ファイルに関して、前記画像データ領域中において対応する位置情報が存在しない画像データを無効な画像データとして検出する検出手段と、

前記検出手段によって検出された無効な画像データの前記画像データ領域における格納位置を当該無効な画像データに隣接する区切り情報に基づいて取得し、取得した格納位置を示す位置情報を生成する生成手段と、

前記生成手段で生成された位置情報を前記管理情報領域に記録して前記無効な画像データを復活させる復活手段とを備えることを特徴とする画像ファイル管理装置。

【請求項 8】

前記位置情報は、前記画像データ領域における前記複数枚の画像の各々の先頭位置を示すことを特徴とする請求項 7 に記載の画像ファイル管理装置。

30

【請求項 9】

前記位置情報は、前記画像データ領域における前記複数枚の画像の各々のデータ量を示すことを特徴とする請求項 7 に記載の画像ファイル管理装置。

【請求項 10】

前記位置情報は、前記画像データ領域における前記複数枚の画像の各々の先頭位置とデータ量を示すことを特徴とする請求項 7 に記載の画像ファイル管理装置。

【請求項 11】

前記検出手段は、

ある画像データの先頭位置とデータ量から求まる次の画像データの先頭位置と、当該次の画像データに対応する位置情報が表す先頭位置とが一致しない場合に無効化された画像データがあると判断する判断手段を備え、

40

前記判断手段において無効化された画像データがあると判断された場合は、前記画像データ領域の、当該画像データの位置情報が示す先頭位置から前記次の画像データの位置情報が示す先頭位置までの間を調査して検出された区切り情報に基づいて無効な画像データを検出することを特徴とする請求項 10 に記載の画像ファイル管理装置。

【請求項 12】

前記管理情報は、前記画像データ領域に含まれる有効な画像データの数を示す画像数情報を含み、

前記復活手段によって復活した画像の数に基づいて前記画像数情報を更新する更新手段

50

を更に備えることを特徴とする請求項7に記載の画像ファイル管理装置。

【請求項13】

複数枚の画像を記憶するための画像ファイル管理処理をコンピュータに実行させるための制御プログラムを格納する記憶媒体であって、

前記画像ファイル管理処理が、

前記複数枚の画像の各々の画像データを区切り情報を挟んで連続して記憶する画像データ領域と、前記各々の画像データの前記画像データ領域における格納位置を示す位置情報を含む管理情報を格納する管理情報領域とを含む画像ファイルに関して、前記画像データ領域中において対応する位置情報が存在しない画像データを無効な画像データとして検出する検出工程と、

10

前記検出工程によって検出された無効な画像データの前記画像データ領域における格納位置を当該無効な画像データに隣接する区切り情報に基づいて取得し、取得した格納位置を示す位置情報を生成する生成工程と、

前記生成工程で生成された位置情報を前記管理情報領域に記録して前記無効な画像データを復活させる復活工程とを備えることを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は複数の画像から所望の画像を検索するために用いる画像データベースに好適な画像管理方法及び装置に関するものである。

20

【0002】

【従来の技術】

多数の画像データの中から所望の画像を検索するための画像データベースが多数提案されている。これらの多くは、

- ・ キーワードや撮影日時等の非画像情報を画像に関連付け、それを基に検索を行う方法、
- ・ 画像自体の特徴量（輝度・色差情報、画像周波数、ヒストグラムなど）を基に検索を行う方法、の2つに大別される。

【0003】

このいずれの方法においても、検索を行うための情報と画像データとは別々に管理されているのが一般的である。例えば、検索用のデータは1つのファイルやリレーショナルデータベースによって管理され、検索対象となる。そして、検索結果から該当する画像データのファイル名が得られ、そのファイル名によって画像データにアクセスし表示する。このような方式を採用するのは、画像データは一般にその容量が大きく、検索用データと分けて管理の方が効率がよいからである。

30

【0004】

個々の画像データはファイルシステム内で管理されるが、その管理の方法として次の2つの方法が考えられる。まず、第一の方法は、すべての画像データをひとつのディレクトリで管理する方法である。第二の方法は、画像データを複数枚数毎のいくつかのグループに分け、それぞれのグループ毎にディレクトリに分類して管理する方法である。たとえば、「動物」、「花」などに画像の内容で分類してディレクトリに分ける方法が考えられる。

40

【0005】

【発明が解決しようとする課題】

しかしながら、上記第一、および第二の方法どちらにおいても、検索キー等による検索の結果得られた複数枚数の画像を同時に表示しようとした場合に、その枚数が多いと、画像のアクセスに極端に時間がかかるようになる。

【0006】

また上記第一の方法の場合、画像の管理は容易であるが、枚数が極端に多くなると、ディレクトリ情報を得るだけでも膨大な時間を要するようになる。第二の方法では、常に、どの画像ファイルがどのディレクトリにあるかという対応関係を正しく維持する必要があり、ファイルの移動などの管理が煩雑になる。

50

【0007】

通常、画像データベースへの画像の登録は自動化されている場合が多い。たとえば、連続するフィルムからスキャナ装置により連続して画像を読み取り、これらをデータベースへ登録する、ビデオテープなどから動画を連続して静止画として取り込み、データベースへ登録するといった方式が実用化されている。

【0008】

しかしながら、写真を撮影する場合であれば、同じシーンを連続して撮影することもある。またビデオであれば、数秒間まったく同じ映像が連続する場合も多々ある。このいずれの場合でも、上記の如き自動的な登録を行えば、全く同じ、或いは非常に類似したシーンの画像が何枚も連続してデータベースに登録されてしまうことになる。また、たとえば映像と映像のあいだにコマーシャルのように、ユーザが必要としない画像がデータベースに登録されてしまう場合もある。

10

【0009】

このような画像はデータベースから削除することが望まれるが、上述したように、一般的な画像データベースでは、画像のアクセスに時間がかかること、ディレクトリ等の管理が複雑なことから、1枚の画像を削除するにも、多くの時間を必要としてしまう。

【0010】

本発明は、上記の問題に鑑みてなされたものであり、画像データへの高速アクセス及び画像データの簡易な管理を可能とするとともに、画像データベースからの画像データの削除を高速に行える画像記憶方法及び装置を提供することを目的とする。

20

【0011】

また、本発明の目的は、いったん削除した画像でも、再び必要となったときに、削除された画像を復活可能とすることを目的とする。

【0012】

【課題を解決するための手段】

上記の目的を達成するための本発明の一態様による画像ファイル管理方法は、たとえば以下の工程を備える。すなわち、

複数枚の画像を記憶するための画像ファイル管理方法であって、

前記複数枚の画像の各々の画像データを区切り情報を挟んで連続して記憶する画像データ領域と、前記各々の画像データの前記画像データ領域における格納位置を示す位置情報を含む管理情報を格納する管理情報領域とを含む画像ファイルに関して、前記画像データ領域中において対応する位置情報が存在しない画像データを無効な画像データとして検出する検出工程と、

30

前記検出工程によって検出された無効な画像データの前記画像データ領域における格納位置を当該無効な画像データに隣接する区切り情報に基づいて取得し、取得した格納位置を示す位置情報を生成する生成工程と、

前記生成工程で生成された位置情報を前記管理情報領域に記録して前記無効な画像データを復活させる復活工程とを備える。

【0013】

また、上記の目的を達成する本発明の他の態様による画像ファイル管理装置はたとえば以下の構成を備える。すなわち、

40

複数枚の画像を記憶するための画像ファイル管理装置であって、

前記複数枚の画像の各々の画像データを区切り情報を挟んで連続して記憶する画像データ領域と、前記各々の画像データの前記画像データ領域における格納位置を示す位置情報を含む管理情報を格納する管理情報領域とを含む画像ファイルを格納する格納手段と、

前記画像ファイルに関して、前記画像データ領域中において対応する位置情報が存在しない画像データを無効な画像データとして検出する検出手段と、

前記検出手段によって検出された無効な画像データの前記画像データ領域における格納位置を当該無効な画像データに隣接する区切り情報に基づいて取得し、取得した格納位置を示す位置情報を生成する生成手段と、

50

前記生成手段で生成された位置情報を前記管理情報領域に記録して前記無効な画像データを復活させる復活手段とを備える。

【0014】

【発明の実施の形態】

以下、添付の図面を参照して本発明の一の実施形態を説明する。

【0015】

<第1の実施形態>

図1は、本実施形態による画像記憶装置としてのコンピュータシステムの構成を示すブロック図である。図1において、101はCPUで、システム全体の制御を行なっている。102はキーボードで、102aのマウスとともにシステムへの操作入力に使用される。103は表示部であり、CRTや液晶等で構成されている。104はROM、105はRAMであり、システムの記憶装置を構成し、システムが実行するプログラムやシステムが利用するデータを記憶する。106はハードディスク装置、107はフロッピーディスク装置で、システムのファイルシステムに使用される外部記憶装置を構成している。108はプリンタであり、画像データに基づいて記録媒体上への可視画像の形成を行う。

10

【0016】

なお、以下で説明する画像ファイルの作成等の処理は、CPU101がROM104もしくはRAM105に格納された制御プログラムを実行することでなされるものである。また、以下の説明において形成される画像ファイルは、最終的にハードディスク106或いはフロッピーディスク107等の外部記憶装置に格納されるものとする。

20

【0017】

図2は第1の実施形態による画像記憶方式により作成される画像ファイルの概略構成図である。図2において、201は画像情報領域であり、画像の枚数、圧縮方式、縦横画素数、特徴量抽出方法などの、画像の読み出しや表示に必要な情報を記憶する領域である。202は画像データ領域であり、当該画像ファイルとして格納すべきすべての画像データを連続して記憶する領域である。203はサイズオフセット領域であり、複数の画像各々のデータ量を連続して記憶する領域である。204は特徴量データ領域であり、画像データ領域202に格納される複数の画像の各々の特徴量（輝度・色差情報、画像周波数、ヒストグラムなど）を連続して記憶する領域である。

【0018】

図3は図2に示した画像情報領域201の詳細なデータ構成例を示す図である。この例では、それぞれの領域が4バイトずつとられているが、扱う画像の数や大きさに応じて領域の大きさは変更してかまわない。

30

【0019】

領域301は本画像フォーマットの改訂番号を示すバージョン（Version）を格納する領域である。領域302は記録している画像の総数を示す画像数（Images）を格納する。領域303はどのような値を画像特徴量として使っているかを示すモード（Mode）を格納する。Modeに、例えば色差情報としてRGB値を用いている場合は値「0」、YUVを用いている場合は値「1」を入れることにより、画像特徴量の種別を表す。領域304は画像データ領域203に蓄積されている画像のフォーマットを示す情報（TileFormat）を格納する。TileFormatには、たとえば、JPEGでは値0、BMPでは値1、FlashPixでは値2がセットされる。

40

【0020】

領域305、306は画像データ領域203に蓄積されている画像の幅を示す情報（TileWidth）、および高さを示す情報（TileHeight）を格納する領域である。なお、これらは画素数によって表される。領域307には特徴量を算出する際の画面の分割方法を示す情報（SectionMode）が格納される。本例では、後述するように画像を6分割していれば値0、分割していなければ値-1がセットされる。

【0021】

領域308には画像データ領域202の先頭アドレスを示すポインタ（PointerToTile）

50

が、領域 3 0 9 にはサイズオフセット領域 2 0 3 の先頭アドレスを示すポインタ (PointerToSizeOfFDS) が、領域 3 1 0 は画像特徴量データ領域 2 0 2 の先頭アドレスを示すポインタ (PointerToData) が格納される。領域 3 1 1 は予備領域であり、 $N \times 4$ バイトの領域が確保されている。なお、本例では $N = 5$ として以下の説明を行う。

【0022】

例えば、画像情報領域 2 0 1 の直後からすきまをあけず画像データ領域 2 0 2 が続くのであれば、画像情報領域 2 0 1 が本例では 6 4 バイトなので、領域 3 0 8 の PointerToTile として、値「64」が格納される。なお、これら 3 0 1 から 3 1 1 までの情報の順番はこの例に限ったものではない。

【0023】

図 4 は図 2 の画像特徴量データ領域 2 0 4 におけるデータ構成の詳細を示す図である。図中 4 0 1、4 0 2、はそれぞれ複数枚ある画像中の 1 番目、2 番目の画像から算出された特徴量を示している。なお、画像特徴量の算出方法については後述する。図 4 において、 $R(0, 0) \sim B(2, 1)$ の計 1 8 個のデータが 1 枚の画像の特徴量をあらわしている。 $R(0, 0)$ 、 $G(0, 0)$ 、 $B(0, 0)$ はそれぞれ 1 枚の画像を 6 分割したうちの、左上角の領域 (図 9 により後述する) の RGB 値の平均値を示している。

【0024】

なお、図 4 において、NA は値に意味のないことを示している。本実施形態では、各分割領域ごとに、R、G、B の平均値を各 1 バイトで示し、区切りの良いように 4 バイトを 1 単位としているためである。別の方法として、NA 部分を削除し、詰めて並べてもよい。

【0025】

図 5 は図 2 の画像データ領域 2 0 2 におけるデータ構成の詳細を示す図である。図 5 では、画像圧縮方式として J P E G を使った場合を示している。したがってこの場合、画像情報領域 2 0 1 中の、画像フォーマットを示す情報が格納される領域 3 0 4 には、J P E G を示す情報が格納される。

【0026】

図 5 において 5 0 1 は画像データ中の 1 番目の画像の J P E G 圧縮データであり、5 0 2 は 2 番目の画像の J P E G 圧縮データである。図中、S O I、A P P 0、D H T、D Q T、S O F 0、S O S、E O I はマーカーと呼ばれる区切り記号である。S O I は J P E G データの開始、E O I はデータの終了、A P P 0 はアプリケーションにより任意に使用可能な領域、D H T はハフマンテーブル、D Q T は量子化テーブル、S O F 0 はベースライン J P E G 圧縮、S O S はハフマンコードを示している。1 つの画像の圧縮データは S O I と E O I で挟まれた部分となる。なお、J P E G については ITU-T WHITE BOOK デジタル静止画像圧縮符号化関連勧告集 (財団法人 新日本 ITU 協会発行) を参照のこと。また、図 5 の例では J P E G データを用いているが、B M P、FlashPix 等他の画像ファイルフォーマットであってもかまわない。

【0027】

図 6 は図 2 に示したサイズオフセット領域 2 0 3 のデータ構成の詳細を示す図である。図 6 において、5 1 0 1 は 1 番目の画像の開始位置 (5 1 0 1 a) およびデータ量 (5 1 0 1 b) を記憶する領域である。同様にして、領域 5 1 0 2、5 1 0 3 と順次画像データの各々の開始位置およびデータ量が記憶される。このようにして、サイズオフセット領域 2 0 3 には、 n 枚の画像データの開始位置 (オフセット) とデータ量 (サイズ) が格納される。

【0028】

次に、ハードディスク 1 0 6 あるいは、フロッピーディスク 1 0 7 上に、以上で説明した構成のフォーマットでデータを書き込み、画像ファイルを作成するための処理手順を説明する。なお、ここでは、本画像フォーマットの改定番号であるバージョン番号が 3、画像枚数が 1 0 0 枚、特徴量モードが RGB、画像フォーマットが J P E G、画像サイズが幅 $\times$ 高さ = 384×256 、特徴量抽出が分割モード (6 分割) である場合を例にあげて説明する。

10

20

30

40

50

【0029】

図7は第1の実施形態における画像ファイル生成処理の概略を示すフローチャートである。図7において、ステップS601では画像情報領域201にヘッダデータを書き込み、ステップS602では画像データの処理を行い、画像データ領域202に画像データの書き込みを行う。ステップS603では、再び画像情報領域201へヘッダデータの書き込みを行う。そして、ステップS604ではサイズオフセット領域、特徴量領域への書き込みを行う。

【0030】

以下、図7に示した各処理について更に詳細に説明を加える。

【0031】

10

図8は図7のステップS601のヘッダデータ書き込み処理を詳細に説明するフローチャートである。なお、本処理を実行するに際して、書き込み対象となる画像ファイルはすでにオープンされているものとする。

【0032】

ステップS701では画像情報領域201内の領域301にバージョン (Version) を示す値 (本例では「3」) を書き込む。また、ステップS702では、領域302に当該画像ファイルに格納すべき画像数 (本例では「100」) を書き込む。ステップS703では、領域303に画像特徴量のモードとして、0 (RGB) 或いは1 (YUV) のいずれかを書き込む (本例では、RGBを採用するので、値「0」が書き込まれる)。ステップS704では画像データ領域203に書き込む画像のフォーマットを表す画像フォーマット304に値0を書き込む。なお、フローチャートでタイルと記載されているのは、画像データ領域に格納される各画像のことである。ステップS705では領域305と領域306のそれぞれに、画像の幅 (TileWidth) (本例では、「384」) と画像の高さ (TileHeight) (本例では「256」) を書き込む。ステップS706では、領域307に、画像特徴量を算出する際の分割モードを示す値を書き込む。本実施形態では、分割しないモードの場合に-1、6分割の場合に0、8分割の場合に1が格納される。したがって本例では6分割が採用されるので、領域307に値0が書き込まれる。

20

【0033】

ステップS707では、当該画像ファイルに格納される特徴量データを一時的に記憶するための領域をメモリ (RAM105) 上に確保する。本実施形態では図4に示すごとく、各々の画像について24バイトの領域を使用する特徴量データが格納されるので、メモリ上に確保する領域は24バイト×画像数となる。なお、ここで、画像数はステップS402で領域302に設定された画像数 (Images) である。なお、ここで確保された領域は、D [Images] [24] という配列として、後述の図9のフローチャートで用いられる。

30

【0034】

ステップS708では各画像の開始位置とデータ量を一時的に記憶する領域をメモリ上に確保する。本実施形態では、図6に示すごとく、各々の画像について開始位置、データ量各4バイトの領域を使用するので、メモリ上に確保する領域は8バイト×画像数となる。ここで、画像数はステップS402で領域302に設定された画像数 (Images) である。なお、ここで確保された領域は、I S I Z E [Images] [2] という配列として、後述の図9のフローチャートで用いられる。

40

【0035】

ステップS709は画像データ領域202の先頭アドレスポインタを領域308 (PointerToTile) へ書き込む。本実施形態では、画像情報領域201は64バイトであり、その直後に画像データ領域が配置されるので、「64」が書き込まれることになる。

【0036】

図9は図7のステップS602における処理の詳細な手順を説明するフローチャートである。

【0037】

図9では複数の入力画像ファイルを逐一開き、各々の画像について、画像特徴量の算出、

50

圧縮処理、圧縮後の画像データ量の算出を行う。さらに、画像データは連続してファイル内に記憶されることから、画像データ領域202の開始位置に算出した画像データ量を累積加算していくことにより、各々の画像の開始位置を算出する。その後、圧縮データの書き込みを行い、1枚の入力画像の処理を終了しファイルを閉じる。この処理をすべての入力画像に対して行う。

【0038】

ステップS801では変数INUMに全画像数の値（領域301に格納されている値、本例では100）をセットする。ステップS802では変数iを0に初期化する。

【0039】

ステップS803では入力ファイル（i）をオープンする。ステップS804ではオープンした画像から特徴量を計算し、ステップS707で確保した一時記憶領域の配列D[i][0]～D[i][23]に記憶する。なお、特徴量の算出処理の詳細については後述する。 10

【0040】

ステップS805ではオープンしたファイルの画像データを圧縮する。圧縮後のデータの容量をSIZEとする。ステップS806では圧縮後のデータ容量SIZEをステップS708で確保した一時記憶領域の配列ISIZE[i][1]に記憶する。また、ISIZE[I-1][0]とISIZE[I-1][1]の和を計算することによりi番目の圧縮画像データのオフセット位置が得られるので、これをISIZE[i][0]に代入する。

【0041】

ステップS807では圧縮後のデータを出力ファイルへ書き込む。書き込む位置は、1枚目の画像は画像データ領域202の先頭アドレスを示すポインタ308から開始し、以後、次の画像はその直前の画像データの直後に連続して配置されるように書き込みを行う。なお、圧縮後のデータは図5で説明したフォーマットで書き込まれる。ステップS808では当該入力ファイル（i）をクローズする。ステップS809では変数iを値1だけ増加させる。ステップS810で、変数iをINUMと比較し、両者が等しくない場合はステップS803にもどり上述の処理を繰り返す。両者が等しければ、INUM個の全ての画像ファイルを処理したことになるので、本処理を終了する。 20

【0042】

次に、上述した画像特徴量の計算（ステップS804）について説明する。図10は本実施形態における特徴量算出時の画面分割を示す図である。図9に示されるように、対象となる画像の大きさは、水平方向にW画素、垂直方向にH画素である。本実施形態では、これを水平方向に3分割、垂直方向に2分割の計6分割し、左上から順に領域(0,0)、領域(1,0)、…領域(2,1)とする。そして、これら各領域のR、G、B値の平均値を算出し、計18個の数値をもって、画像の特徴量とする。 30

【0043】

図11は本実施形態による特徴量算出処理を説明するフローチャートである。まず、ステップS1001で変数kを値0で初期化し、ステップS1002で変数jを値0で初期化し、ステップS1003で変数iを値0で初期化する。

【0044】

次に、ステップS1004で、配列dのk番目の要素d(k)に、領域(i, j)のR値の平均値を代入する。また、d(k+1)にG値の平均値、d(k+2)にB値の平均値を代入する。なお、R、G、B値の平均値の算出方法は図12のフローチャートを用いて後述する。 40

【0045】

ステップS1005では、kを値3だけ増加させる。ステップS1006で、iを値1だけ増加させる。ステップS1007ではiを値2と比較し、2より多きければステップS1008へ進む。そうでなければステップS1004へ戻る。

【0046】

iが2よりも大きくなった場合は、当該分割行に対する処理が終了したことを表すので、次の分割行へ進むことになる。従って、ステップS1008で、jを値1だけ増加させる 50

。ステップS1009ではjを値1と比較する。jが1より多きければ、分割行の第2行目の処理を終えたこと、すなわち当該画面の全体の処理を終えたことを示すので、本処理を完了する。そうでなければ、新たな分割行について処理を行うためにステップS1003へ戻る。

【0047】

以上のような処理を完了すると、18個の要素をもつ配列d（）に、イラスト画像の画像特徴量が格納されることになる。

【0048】

なお、ここでは特徴量の算出のため、画像を6個の等面積の矩形領域に分割しているが、分割は矩形に限らずより複雑な形状でもよいし、分割数を増減させても良い。分割数を増減したときは、特徴量の要素数は18個でなく、それに応じて増減することは容易に理解され得る。

10

【0049】

次に、R、G、B値の平均値の算出方法について更に詳しく説明する。図12は、領域毎のR、G、B値の平均値算出方法を説明するフローチャートである。なお、画像データは、R(X,Y)、G(X,Y)、B(X,Y)の3つの配列に格納されているものとする。ただし、 $0 \leq X < W$ 、 $0 \leq Y < H$ であり、画像の左上隅を起点(0,0)とする。

【0050】

図12に示される処理では $X_0 \leq X < X_1$ 、 $Y_0 \leq Y < Y_1$ の部分領域の平均濃度を算出し、変数DR、DG、DBのそれぞれにR、G、Bの平均濃度値を入れて返す。

20

【0051】

ステップS804及び図11によって示した処理において、領域(i,j)に相当する領域は、

$$X_0 = W \times i / 3, X_1 = W \times (i + 1) / 3$$

$$Y_0 = H \times j / 2, Y_1 = H \times (j + 1) / 2$$

に対応するので、定数X0、X1、Y0、Y1を上記のように初期化してから図12フローチャートを実行する。

【0052】

まず、ステップS1101で変数DR、DG、DBを値0で初期化する。ステップS1102で変数Yを上記のY0で初期化する。同様に、ステップS1103で変数Xを上記のX0で初期化する。

30

【0053】

次に、ステップS1104で、DRにR(X,Y)を加える。同様にDGにG(X,Y)、DBにB(X,Y)を加える。そして、ステップS1105で変数Xを値1だけ増加させる。次に、ステップS1106で変数XとX1を比較し、等しければステップS1107へ、そうでなければS1104へ戻る。ステップS1107では変数Yを値1だけ増加させる。そして、ステップS1108で変数YとY1を比較し、等しければステップS1109へ、そうでなければステップS1103へ戻る。以上のステップS1103～ステップS1108の処理により、DR、DG、DBのそれぞれには、X0、X1、Y0、Y1で特定される領域内のR値の合計値、G値の合計値、B値の合計値が格納されることになる。

40

【0054】

次に、ステップS1109で、変数DR、DG、DBをそれぞれ $(X_1 - X_0) \times (Y_1 - Y_0)$ で除算する。これは、各変数に格納されている値を領域内の画素の数で割ること、すなわち平均値を取ることを表す。従って、ステップS1109の処理により、DR、DG、DBの内容は、領域内の画素濃度の総和を画素数で割った平均濃度となる。

【0055】

次に、図7のステップS603の処理を説明する。図13はステップS603におけるヘッダデータの際書き込み処理の詳細を示すフローチャートである。

【0056】

図13において、ステップS1201では最後の画像、すなわちこの例では100枚目の

50

画像の開始位置 (ISIZE[INUM-1][0]) とデータ量 (ISIZE[INUM-1][1]) を加えることで、画像データ領域 202 の直後に位置するサイズオフセット領域 203 の開始アドレスを算出する。算出した値を、画像情報領域 201 内のサイズオフセット領域 203 の先頭アドレスを示すポインタPointerToSize0FSとして領域 309 へ書き込む。

【0057】

また、ステップ S1202 ではサイズオフセット領域 203 の先頭アドレスポインタ 309 の値に当該サイズオフセット領域が占める容量、すなわち 8 バイト×画像数 (Images) を加算した値を、特徴量データ領域 204 の先頭アドレスを示すポインタPointerToDataとして、画像情報領域 201 内の領域 310 へ書き込む。

【0058】

次に、図 7 のステップ S604 における、サイズ領域・特徴量領域書込処理を図 14 のフローチャートを参照して説明する。

【0059】

ステップ S1301 ではメモリ上に確保され、ステップ S806 の処理によって所定のデータが格納された配列 I S I Z E [Images][2]を、サイズオフセット領域 203 へ書き込む。なお、このサイズオフセット領域 203 への書き込みの先頭アドレスは、領域 309 に格納されたPointerToSize0FSが示す位置となる。続いて、ステップ S1302 では、メモリ上に確保されステップ S804 の処理によって特徴量データが格納された配列 D [Images][24]を、特徴量データ領域 204 へ書き込む。なお、この特徴量データ領域 204 への書き込みの先頭アドレスは、領域 310 に格納されたPointerToDataが示す位置となる

【0060】

次に前記作成した画像ファイルから所望の画像データを読み出す処理の流れを説明する。図 15 は本実施形態による画像データの検索処理を説明するフローチャートである。

【0061】

ステップ S1501 では、画像情報領域から画像数 302、サイズオフセット領域へのポインタ 309、特徴量データ領域へのポインタ 310、画像データ領域へのポインタ 308 等を読み出す。ステップ S1502 ではサイズオフセット領域 203 の内容をメモリ (RAM105) 上の一時記憶領域に読み出す。ステップ S1503 では特徴量データ領域 204 の内容をメモリ (RAM105) の一時記憶領域に読み出す。ステップ S1504 では不図示の与えられた画像特徴量と、メモリに記憶された特徴量データ領域の内容とを逐一比較し、そのもっとも類似したものを取り出す。ステップ S1505 ではステップ S1504 で該当した番号の画像の先頭アドレスへのポインタを、ステップ S1502 でメモリに格納したサイズオフセット領域の内容を参照して獲得する。そして、獲得した画像の先頭アドレスへのポインタにより当該画像ファイルの画像データ領域から特定された画像を読み出し、表示する。

【0062】

さて、以上のような画像データファイルに不要な画像データが含まれていた場合には、それを当該画像データファイルから削除する必要がある。以下、不要な画像を削除する方法について説明する。

【0063】

図 16 は削除前の画像ファイルの状態の一例を示す図である。図 16 の画像ファイルでは 5 枚の画像が格納されている。画像情報領域 601 は図 2 の画像情報領域 201 に対応する。

【0064】

602 から 606 は画像データ領域 202 に格納された各画像データを示している。607 から 611 は、サイズオフセット領域 203 で、P0～P4 は画像データ 602～606 の開始位置の値、S0～S4 は画像データ 602～606 のデータ量を示している。612～616 は画像特徴量データ領域 204 で、画像データ 602～606 の各々の特徴量データを D0～D5 で示してある。

10

20

30

40

50

【0065】

図17は削除後の不要な画像データを含む画像ファイルの状態を説明する図である。この例では2番目の画像データを削除した後で、不要な2番目の画像データを含んだ様子を示している。このとき、画像情報領域701内の領域302の画像数(Images)は5から4に更新される。703は削除対象であるところの2番目の画像データであり、このように削除後も画像データ703はファイル中に存在している。707~711のサイズオフセット領域では2番目の画像領域の開始位置P1, データ量S1を示す項目が削除され、順次配列が詰められている。ただし、711にはもともと記憶されていた5番目の画像の開始位置P4とデータ量S4の値が残っている。

【0066】

同様に、712~716の特徴量データ領域でも、2番目の画像の特徴量D1は削除され、D0, D2, D3, D4が順につめられて記憶されている。そして、716にはもともとの値が残されている。

【0067】

図18は上記削除処理を説明するフローチャートである。

【0068】

ステップS1801では画像情報領域202の領域302から画像数(Images)を読み出してこれを変数inumへセットする。また、領域309のサイズオフセット領域へのポインタ(PointerToSize0FS)を読み出してこれを変数pSizeへセットする。更に、領域310の特徴量データ領域へのポインタ(PointerToData)を読み出してこれを変数pDataにセッ

【0069】

ステップS1802ではステップS1801で得られたpSizeの位置から、サイズオフセット領域の内容をメモリ(RAM105)上に設けた配列ISIZE[Images][2]へ、またpDataの位置から特徴量データ領域の内容をメモリ(RAM105)上の配列D[Images][24]へ読み込む。

【0070】

ステップS1803では変数iを値0で初期化する。また、ステップS1804では変数jを値0で初期化する。

【0071】

ステップS1805では変数i番目の画像が削除対象であるかどうかを判断する。この判断の根拠となるものはたとえば、キーボード102やマウス102aによってユーザから指示された内容である。削除対象の選択方法については、ここでは限定するものではない。ステップS1805で、第i番目の画像が削除対象であった場合はステップS1808に進み、削除対象でない場合はステップS1806へ進む。

【0072】

ステップS1806ではISIZE[i][2]の内容をISIZE2[j][2]にコピーするとともに、D[i][24]の内容をD2[j][24]へコピーする。ステップS1807では変数jを値1だけ増加する。そして、ステップS1808では変数iを値1だけ増加する。

【0073】

ステップS1809では変数iの値と変数inumの値を比較し、両者が等しい場合はステップS1810へすすみ、そうでない場合はステップS1805へ戻る。ステップS1810では変数jの値を画像情報領域201の領域302へ画像数(Images)として書き込む。そして、ステップS1811ではISIZE2[Images][2]の内容をサイズオフセット領域203へ書き込むとともに、特徴量データD2[Images][24]の内容を特徴量データ領域204へ書き込む。なお、領域309と領域310の内容(PointerToSize0FSとPointerToData)は変化しないので、各領域のi-j個の画像分のデータはそのまま残ることになる(図17の領域711、716)。

【0074】

以上のようにして、図17に示したように、画像ファイル内に存在する所望の画像データ

10

20

30

40

50

を無効化することができる。以下では、以上のような無効化処理を誤って実行してしまった場合等において、削除した画像データを再度有効化（復活）するための処理（以下、再生処理という）を説明する。上述したように、本実施形態の削除処理では画像データの形態は削除されず、サイズオフセット領域の情報、特徴量データ領域のみが削除されている。従って、再生処理においては、これらの情報を復活させればよい。

【0075】

上述の図17においては、2番目の画像データを削除した様子が示されている。この場合、画像情報領域201で示される画像総数（Images）は4であるが、実際に画像データ領域202には702～706までの5つの画像データが記憶されている。すなわち、703は削除対象となった2番目の画像データであり、このように削除後も画像データ703はファイル中に存在している。したがって、削除というよりは、無効化という方が適当である。従って、以下の図19に示される再生処理は、無効化された画像データを有効化するものということもできる。

10

【0076】

図19は本実施形態による、無効画像の再生処理の流れを示すフローチャートである。本再生処理では、サイズオフセット領域の各画像のオフセット位置、サイズ情報から順次次の画像の開始位置を算出する。こうして算出した次画像の開始位置と、サイズオフセット領域に格納されているオフセット位置によって示される次画像の開始位置情報とが等しければ、その間は画像が詰まっていると判断する。一方、算出された開始位置とサイズオフセット領域のオフセット情報で示される開始位置とが異なる場合、次の画像との間に、すくなくともひとつの無効画像データが存在することになる。

20

【0077】

たとえば、図17において領域709に格納されているオフセット情報P3とサイズS3から次の画像の開始位置が求まるが、これは領域710に格納されているオフセット情報P4が示す開始位置と等しく、この間に無効な画像データは存在しない。一方、領域707におけるオフセット情報P0とサイズS0から次画像の開始位置を求めると、図17のP1の位置が求まる。これに対して、次の画像のオフセット情報は、領域708に格納されているP2であるから、算出された開始位置とは一致しない。従って、この間に無効化された画像データが存在すると判断される。

【0078】

そして、この無効画像データ領域をスキャンし、J P E GデータであればE O I マーカーコードを検出することで、一つの無効画像データを抽出できる。そのときの画像のデータ量を、再登録することで、無効画像データを再生することができる。以下、図19のフローチャートに従って更に詳細に説明する。

30

【0079】

まず、ステップS1901では変数iに領域302（Images）の値を、変数pSizeに領域309（PointerToSize）の値を、変数pDataに領域310（PointerToData）の値を代入する。ステップS1902では配列ISIZEにサイズオフセット領域のサイズデータを読み込む。また配列IPOSにはサイズオフセット領域のオフセットデータを読み込む。更に、ステップS1903では変数iを値0で初期化し、ステップS1903では変数jを値0で初期化する。

40

【0080】

ステップS1905では変数PNEXTに、配列要素OPOS[j-1]とOSIZE[j-1]の和を代入する。なお、この処理において、OPOS[-1]にはPointerToTileの値が、OSIZE[-1]には0がセットされている。そして、ステップS1906において、変数PNEXTとIPOS[i]を比較する。ここで、両者が等しい場合は、算出された開始位置とオフセット情報による開始位置が等しいので、無効化された画像は存在しないと判断され、ステップS1909へ進む。一方、両者が等しくない場合は、算出された開始位置とオフセット情報による開始位置が不一致であり、無効化された画像データが存在すると判断され、ステップS1907へ進む。

【0081】

50

ステップ S 1 9 0 7 では、図 5 で説明した画像データの構造から、E O I マーカーを検出し、検出した位置を変数 TPOS、そのときに読み込んだ画像データのデータ量を変数 TSIZE に代入する。そして、ステップ S 1 9 0 8 において、配列要素 OSIZE[j] に変数 TSIZE の内容を、OPOS[j] に変数 TPOS の内容をそれぞれ代入する。こうして、オフセットサイズ領域に、無効化された画像データのオフセット位置とサイズが復活する。

【0082】

一方、無効化された画像が存在しないと判断された場合、ステップ S 1 9 0 9 において配列要素 OSIZE[j] に配列要素 ISIZE[i] の内容を、OPOS[j] に ISIZE[i] の内容を代入する。

【0083】

ステップ S 1 9 1 0 では変数 i を値 1 だけ増大する。ステップ S 1 9 1 1 では変数 j を値 1 だけ増大する。ステップ S 1 9 1 2 では、OPOS[j] と PointerToSize を比較し、どちらも等しい場合にはステップ S 1 9 1 3 へ進み、サイズオフセット領域に OPPOS[0] ～ OPPOS[j] と、OSIZE[0] ～ OSIZE[j] を書き込むとともに、Images を変数 j の内容で更新し、本処理を終了する。ステップ S 1 9 1 2 の条件を満足しない場合は、ステップ S 1 9 0 5 に戻り、上記処理を繰り返す。

【0084】

以上説明したように、画像ファイル中の無効画像データをスキャンする手段を設けたことにより、無効データを再登録することが可能になる。

【0085】

また、復活した画像データについて上述した特徴量算出を行ない、特徴量データ領域 2 0 4 の適切な位置に挿入する必要がある。すなわち、復活した画像データに対してステップ S 8 0 4 で説明した特徴量算出処理を行ない、得られた特徴量データ（配列 D[i][0] ～ D[i][23]）を特徴量データ領域 2 0 4 内の復活した画像データに対応する位置に挿入する。

【0086】

なお、上記実施形態では、サイズオフセット領域に各画像の開始位置を示すオフセット情報と各画像のサイズを示すサイズ情報が格納されているので、これらを用いてステップ S 1 9 0 5、S 1 9 0 6 のようにして無効化された画像の存在を検出している。しかしながら、本発明は、サイズオフセット領域 2 0 3 に各画像の開始位置を示すオフセット情報のみを記憶している場合や、各画像のサイズ情報のみを記憶している場合についても適用できることは明らかである。

【0087】

たとえば、サイズオフセット領域 2 0 3 にオフセット情報のみを記憶している場合は、各画像のサイズをオフセット位置の間隔によって得ることができる。ただし、たとえば図 1 7 に示すように削除を行なった場合、オフセット位置 P 0 の次に P 2 が記録されることになる。従って、ステップ S 1 9 0 5、S 1 9 0 6 で示した処理の代わりに、次のオフセット情報が示す位置まで画像データ領域をスキャンし、検出された S O I と E O I に基づいて無効化画像の存在を知ることができる。たとえば、オフセット位置 P 0 から P 2 をスキャンすれば、スキャンの最中のオフセット位置 P 1 のところで S O I と E O I が出現するので無効化画像が存在していることがわかる。

【0088】

また、オフセットサイズ領域にサイズ情報のみを記憶している場合は、各画像のオフセット位置を各画像のサイズを累積することで得ることができる。ただし、たとえば図 1 7 に示すように削除を行なった場合、次の画像データの位置 P 2 を正しく得られるように、領域 7 0 7 のサイズ情報を S 0 + S 1 とする必要がある。従って、ステップ S 1 9 0 5 と S 1 9 0 6 で示した処理の代わりに、サイズ情報の累積によって得られる位置から、対応する画像のサイズの全体にわたって画像データ領域をスキャンし、検出された S O I と E O I に基づいて無効化画像の存在を知る。たとえば、図 1 7 の場合、P 0 の位置から S 0 + S 1 の大きさの画像データ領域をスキャンすることになる。そして、この場合、スキャンの最中のオフセット位置 P 1（位置 P 0 からサイズ S 0 の位置）のところで S O I と E O

10

20

30

40

50

I が出現するので、無効化画像が存在していることがわかる。

【0089】

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0090】

また、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コードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読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。

10

【0091】

この場合、記憶媒体から読出された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になり、その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。

【0092】

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、フロッピーディスク、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などを用いることができる。

【0093】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）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。

20

【0094】

さらに、記憶媒体から読出された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込まれた後、その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。

30

【0095】

【発明の効果】

以上説明したように、本発明によれば、画像データへの高速アクセス及び画像データの簡易な管理が可能となるとともに、画像データベースからの画像データの削除を高速に行える。

【0096】

また、本発明によれば、いったん削除した画像でも、再び必要となったときには、削除された画像を復活させることが可能となる。

【0097】

【図面の簡単な説明】

40

【図1】本実施形態による画像記憶装置としてのコンピュータシステムの構成を示すブロック図である。

【図2】第1の実施形態による画像記憶方式により作成される画像ファイルの概略構成図である。

【図3】図2に示した画像情報領域201の詳細なデータ構成例を示す図である。

【図4】図2の画像特徴量データ領域204におけるデータ構成の詳細を示す図である。

【図5】図2の画像データ領域202におけるデータ構成の詳細を示す図である。

【図6】図2に示したサイズオフセット領域203のデータ構成の詳細を示す図である。

【図7】第1の実施形態における画像ファイル生成処理の概略を示すフローチャートである。

50

【図 8】図 7 のステップ S 6 0 1 のヘッダデータ書き込み処理を詳細に説明するフローチャートである。

【図 9】図 7 のステップ S 6 0 2 における処理の詳細な手順を説明するフローチャートである。

【図 1 0】本実施形態における特徴量算出時の画面分割を示す図である。

【図 1 1】本実施形態による特徴量算出処理を説明するフローチャートである。

【図 1 2】領域毎の R, G, B 値の平均値算出方法を説明するフローチャートである。

【図 1 3】ステップ S 6 0 3 におけるヘッダデータの際書き込み処理の詳細を示すフローチャートである。

【図 1 4】図 7 のステップ S 6 0 4 における、サイズ領域・特徴量領域書込処理を説明するフローチャートである。 10

【図 1 5】本実施形態による画像データの検索処理を説明するフローチャートである。

【図 1 6】画像削除前の画像ファイルの状態の一例を示す図である。

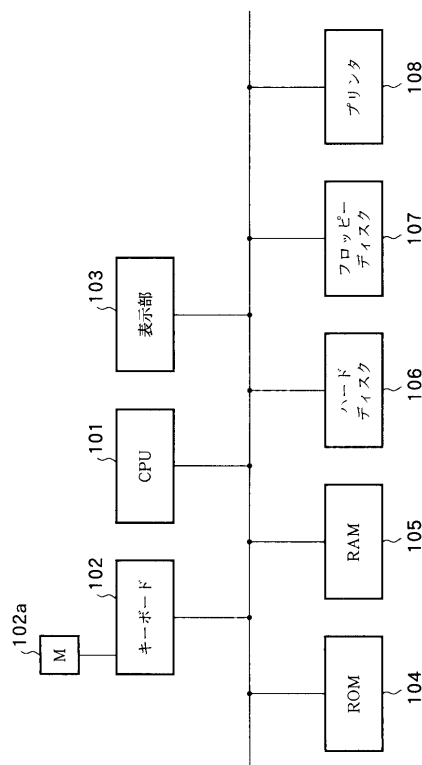
【図 1 7】画像削除後の不要な画像データを含む画像ファイルの状態を説明する図である。

。

【図 1 8】画像の削除処理を説明するフローチャートである。

【図 1 9】本実施形態による、無効画像の再生処理の流れを示すフローチャートである。

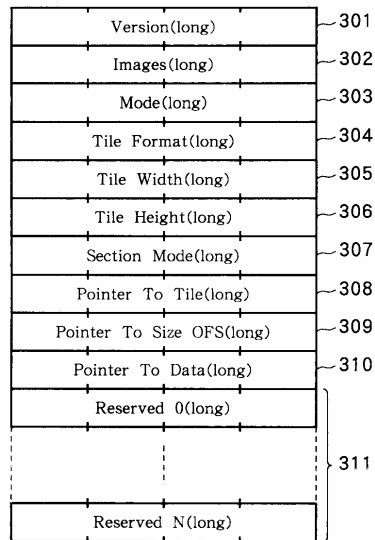
【図 1】



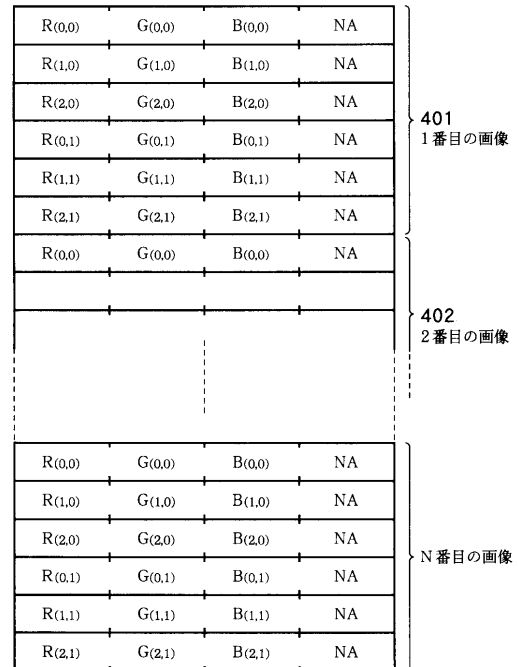
【図 2】



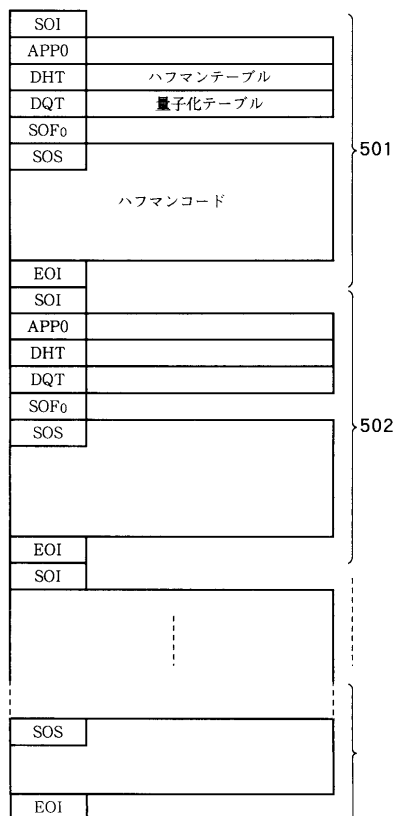
【図 3】



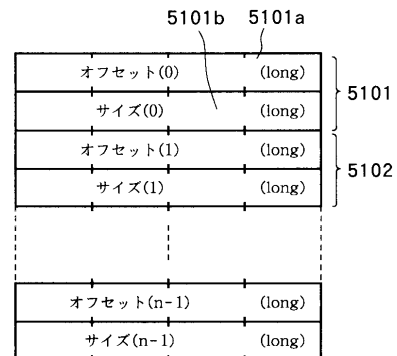
【図 4】



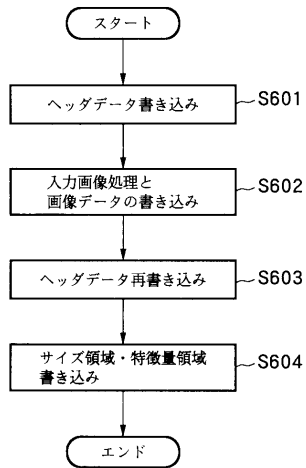
【図 5】



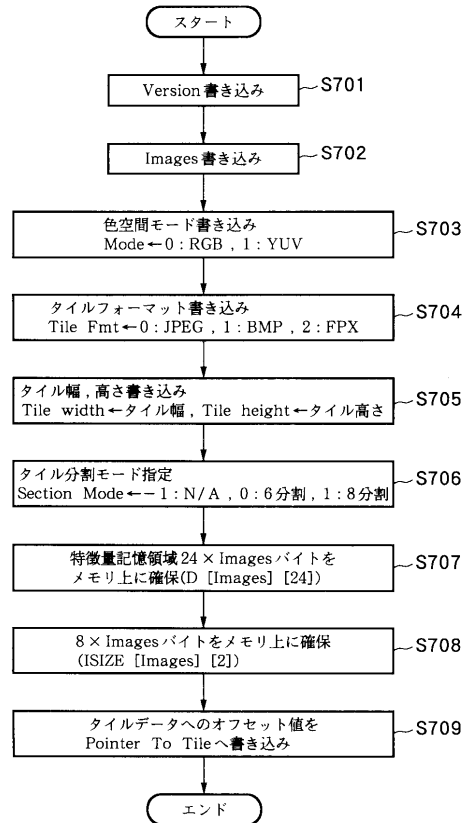
【図 6】



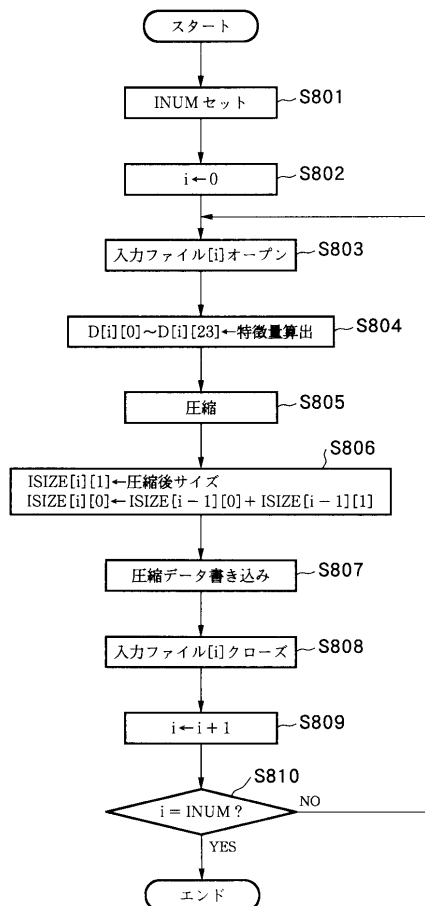
【図 7】



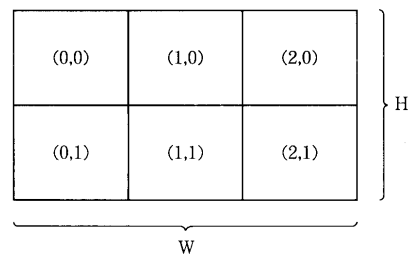
【図 8】



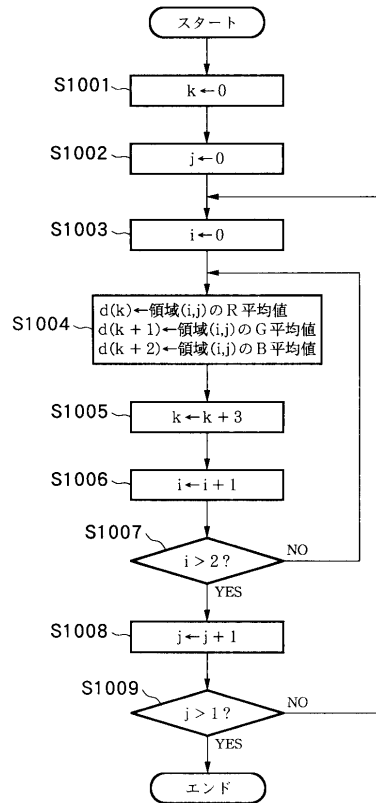
【図 9】



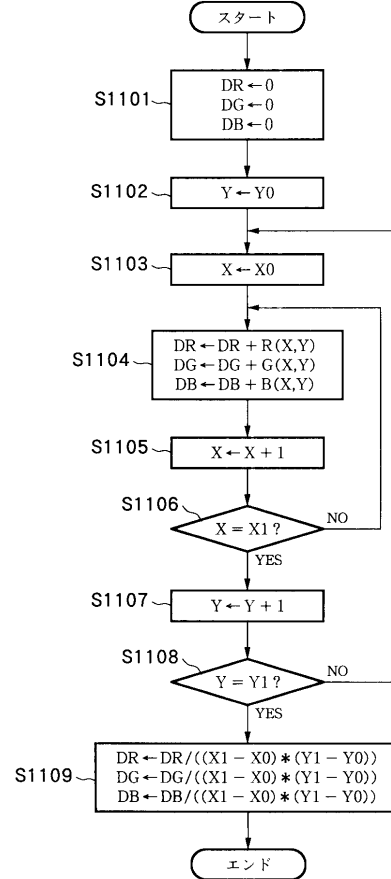
【図 10】



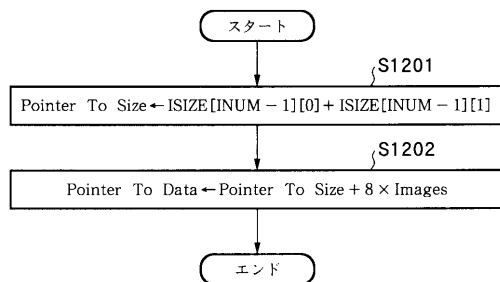
【図 1 1】



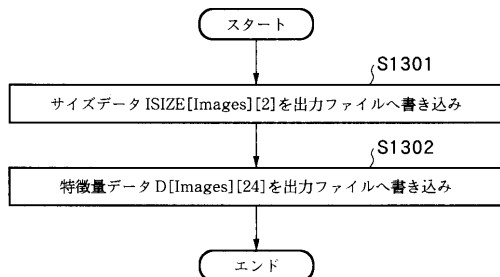
【図 1 2】



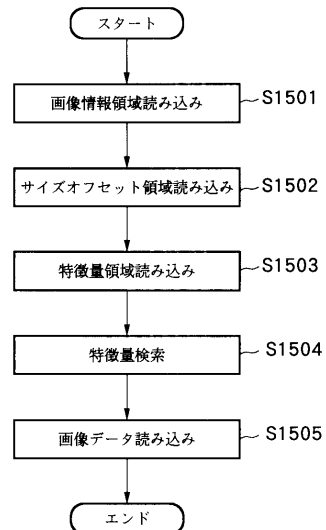
【図 1 3】



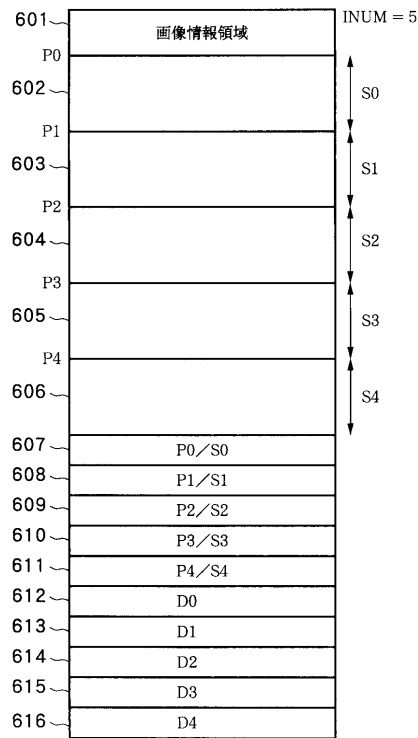
【図 1 4】



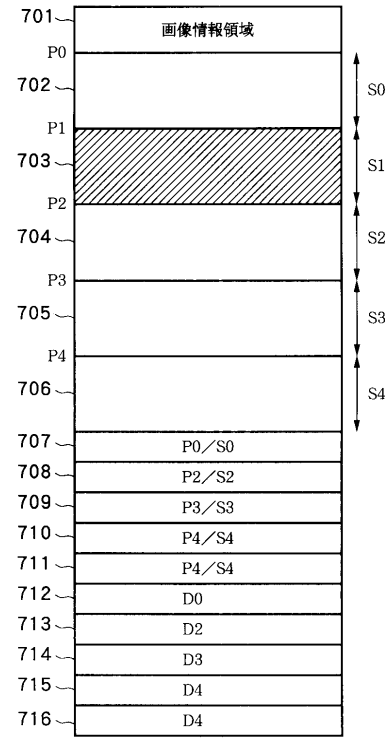
【図 1 5】



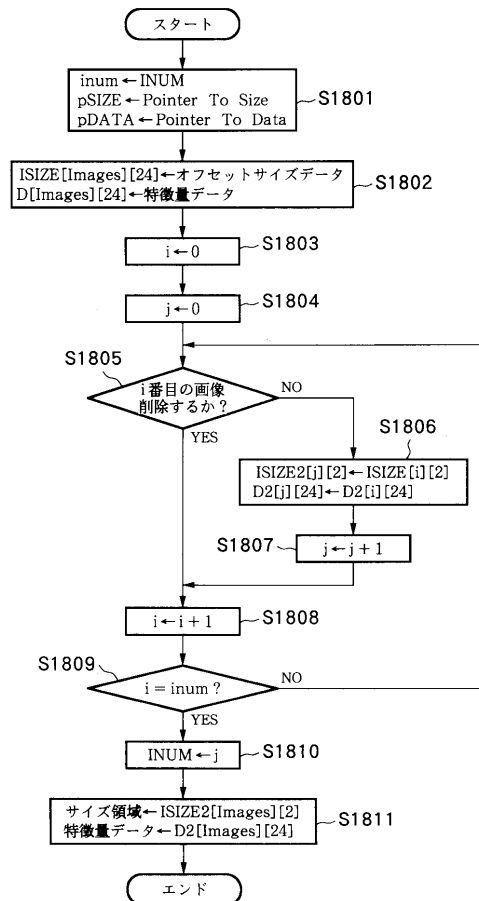
【図 16】



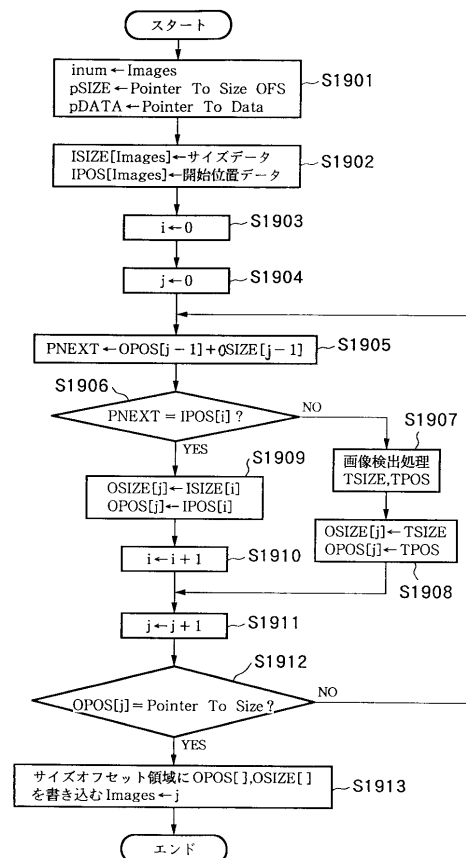
【図 17】



【図 18】



【図 19】



フロントページの続き

- (72)発明者 松本 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
- (72)発明者 草間 澄
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 山本 邦浩
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 桜井 茂行

- (56)参考文献 特開平04-323774 (JP, A)
特開平06-035780 (JP, A)
特開平02-156346 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 12/00

G06F 17/30

G06T 1/00