

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5180156号
(P5180156)

(45) 発行日 平成25年4月10日 (2013. 4. 10)

(24) 登録日 平成25年1月18日 (2013. 1. 18)

(51) Int. Cl.

F I

G 0 6 T 1/00 (2006. 01)

G 0 6 T 1/00 2 0 0 E

G 0 6 F 17/30 (2006. 01)

G 0 6 F 17/30 1 7 0 B

請求項の数 9 (全 40 頁)

(21) 出願番号 特願2009-153615 (P2009-153615)
 (22) 出願日 平成21年6月29日 (2009. 6. 29)
 (65) 公開番号 特開2010-9606 (P2010-9606A)
 (43) 公開日 平成22年1月14日 (2010. 1. 14)
 審査請求日 平成24年6月28日 (2012. 6. 28)
 (31) 優先権主張番号 12/163, 186
 (32) 優先日 平成20年6月27日 (2008. 6. 27)
 (33) 優先権主張国 米国 (US)

早期審査対象出願

(73) 特許権者 502096543
 パロ・アルト・リサーチ・センター・イン
 コーポレーテッド
 Palo Alto Research
 Center Incorporated
 アメリカ合衆国、カリフォルニア州 94
 304、パロ・アルト、コヨーテ・ヒル・
 ロード 3333
 (74) 代理人 100075258
 弁理士 吉田 研二
 (74) 代理人 100096976
 弁理士 石田 純
 (72) 発明者 ドロン クレッター
 アメリカ合衆国 カリフォルニア サン
 マテオ タートル ベイ プレイス 20
 最終頁に続く

(54) 【発明の名称】 局所化された2次元の視覚的指紋を使用してイメージコレクション内のピクチャイメージを見つ
 けるシステムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

少なくとも1つのプロセッサを用いて、イメージコレクションの内容を前処理して、前記イメージコレクション内のイメージごとに安定した再現可能キーポイントを識別するステップと、

少なくとも1つのプロセッサを用いて、前記キーポイントの局所グループから指紋情報を計算するステップと、

少なくとも1つのプロセッサを用いて、前記計算の結果の指紋情報をコンパクトな指紋データベースに格納するステップと、

少なくとも1つのプロセッサを用いて、前記コンパクトな指紋データベース内の入力指紋ごとに、候補指紋の組合せの列を生成するステップと、

少なくとも1つのプロセッサを用いて、指紋データベースに入力された指紋ごとの候補指紋の組合せの前記列を、ファンツリーおよびハッシュテーブルの少なくとも一方と、対応する指紋データ構造と、に格納するステップと、

少なくとも1つのプロセッサを用いて、キーポイントを識別することと、クエリイメージから指紋を計算することと、クエリ指紋を既存のファンツリー指紋データと照合することとによってリアルタイムイメージクエリを実行し、前記コレクション内の一致するターゲットイメージまたはターゲットイメージのセットを判定するステップと、

少なくとも1つのプロセッサを用いて、前記判定ステップに基づいて少なくとも1つのターゲットイメージを取り出すステップと、

10

20

前記取り出された少なくとも１つのターゲットイメージの表示、印刷、格納、または送信のうちの少なくとも１つを行うステップと、

を含み、

前記指紋情報の前記計算は、イメージ内で見つかる確率が非アンカキーポイントに比べて高いキーポイントであり、アンカキーポイントと呼ばれるイメージ内の最も強いキーポイントのサブグループと、加えて複数の局所的な非アンカキーポイントと、の組合せに基づく、

ことを特徴とする、イメージデータを格納し、インデクシングし、検索し、および／または取り出すイメージ管理方法。

【請求項２】

10

請求項１に記載の方法であって、

前記クエリ指紋の前記既存のファンツリー指紋データとの前記照合は、

クエリ指紋ごとに、オリジナルの指紋のイメージコレクションのサブセットを含む特定のファンツリー葉ノードリストにつながるファンツリーアドレスを計算することと、

その後、前記サブセットリストからの各オリジナル指紋を順に現在のクエリ指紋と比較することと、

前記クエリ指紋が前記取り出された指紋レコードのうちの１つと一致するたびに、そのイメージの一致する指紋の個数のカウントを増分することと、

指紋スコア分析モジュールによって、各イメージの一致する指紋の結果の累算されたカウントを調べることと、

20

最高スコア分析に基づいて、前記コレクション内の最もよく一致するイメージまたはイメージのセットを判定することと、を含み、

前記クエリ指紋は、ファンツリーのルートノードから、ファンツリーの間ノードを通して、ファンツリーの葉ノードに至る経路を特定するものである、
ことを特徴とする方法。

【請求項３】

請求項１に記載の方法であって、

前記アンカキーポイントは、前記クエリイメージおよびターゲットイメージ内で見つかる確率が前記非アンカキーポイントより高いことを特徴とする方法。

【請求項４】

30

請求項１に記載の方法であって、

各アンカキーポイントは、局所的な影響領域に関連付けられていることを特徴とする方法。

【請求項５】

請求項１に記載の方法であって、

前記指紋は、１つまたはそれ以上のアンカキーポイントの局所的近傍におけるキーポイントの組合せから形成され、非アンカキーポイントが所定の個数まで欠けていても有効な指紋の一致が得られることを特徴とする方法。

【請求項６】

請求項１に記載の方法であって、さらに、

40

イメージ回転による並び替えによって見つかる同値類の数を、`smallest least common quantization digit (SLCD)`法に基づいて減らすことを含む方法。

【請求項７】

請求項１に記載の方法であって、

指紋の確定には、非アンカキーポイントは使用せずアンカキーポイントのサブグループを使用して指紋の原点中心を確定することが含まれ、

ここで、前記指紋の原点中心は、現在のキーポイントに最も近いアンカキーポイントのサブセットから計算され、非アンカキーポイントが余分にある場合または欠けている場合であっても前記原典が安定なままであることを確実にするために、前記原点中心の計算に

50

は前記非アンカキーポイントを使用しないことを特徴とする方法。

【請求項 8】

クエリ文書のクエリイメージ内のキーポイントおよび潜在的ターゲット文書のコレクションの潜在的ターゲット文書イメージ内のキーポイントを含む、イメージ内のキーポイントを識別するように構成された識別モジュールと、

前記キーポイントから指紋情報を生成するように構成された指紋生成モジュールであって、前記指紋情報が、少なくとも 1 つのアンカキーポイントおよび所定の個数の非アンカキーポイントを含み、前記少なくとも 1 つのアンカキーポイントは前記所定の個数の非アンカキーポイントより強いキーポイントであり、イメージ内のより強いキーポイントとは、イメージ内で見つかる確率が非アンカキーポイントに比べて高いキーポイントである、指紋生成モジュールと、

10

前記クエリイメージの前記指紋情報を、潜在的ターゲット文書の前記コレクション内の前記潜在的ターゲット文書イメージの指紋情報と比較するように構成された比較モジュールと、

前記クエリイメージの前記指紋情報と前記潜在的ターゲット文書イメージの前記指紋情報との間の最良一致を判定するように構成された照合モジュールと、

前記照合モジュールにおける判定に基づいて少なくとも 1 つのターゲット文書イメージを取り出すように構成された取出しモジュールと、

前記取り出された少なくとも 1 つのターゲット文書イメージの表示、印刷、格納または送信を行う、電子ディスプレイ、プリンタ、メモristレージ、またはコンピュータネットワークのうちの少なくとも 1 つと、

20

を含むことを特徴とする、イメージデータの格納、インデクシング、検索、および / または取出しを行う、コンピュータ動作可能システム内のイメージ管理システム。

【請求項 9】

エラーを補正するためにノイズガードを用いることを特徴とする、請求項 8 に記載のシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本願は、ターゲットピクチャイメージの不完全なコピーであるクエリ (query) ピクチャイメージのイメージ (画像) に基づいて大きいイメージコレクション内のターゲットピクチャまたはターゲットピクチャセットのイメージを見つけるソフトウェアプログラムおよびアルゴリズムを組み込んだコンピュータ動作可能なシステムおよび方法を対象とする。

30

【背景技術】

【0002】

クエリイメージは、デジタルカメラ、携帯情報端末、文書スキャナ、テキストリーダー、ビデオカメラ、動画カメラ、コンピュータ、セル電話機カメラ、またはターゲットイメージのイメージ表現を生成することのできる他のデバイスのうちの少なくとも 1 つ (これらに限定はされない) によって取り込むことができる。

40

【0003】

ターゲットイメージを、モニタまたはコンピュータスクリーンに表示することができ、そのピクチャを、上のデバイスのうちの 1 つによって直接に撮影することができ、あるいは、ターゲットイメージを、まずプリンタまたは類似する出力デバイス上で印刷することができ、ピクチャを、ターゲットイメージの複写のために上のデバイスのうちの 1 つによって撮影することができる。代替案では、クエリイメージを、クエリイメージの格納された電子版から複写することができる。

【0004】

クエリイメージを取り込むのに使用される形およびデバイスに起因して、しばしば、取り込まれたクエリイメージは、ターゲットイメージと比較して、低解像度を有し、ぼけて

50

おり、回転および透視ビューイング条件によってひずみ、不均一な明るさを有する。

【0005】

イメージの対応を照合する通常の方法は、次のステップからなる。

【0006】

(1) 第1ステップでは、キーポイントが、角、接続、および/または明るいもしくは暗い斑点などのイメージ内の特有の位置について識別される。目標は、異なるビューイング条件、ノイズ、およびさまざまなイメージ劣化の下で、同一のキーポイントを信頼できる形で見つけることである。多数の既存の方法の中の1つの方法は、非特許文献1に記載の Scalable Invariant Feature Transform (SIFT) 法である。もう1つの方法は、非特許文献2に記載の PCA-SIFT 法である。両方の方法が、かなりの量の計算を必要とし、これが、大きいイメージコレクションに関する性能を制限する。

10

【0007】

(2) 第2ステップでは、「記述子」と呼ばれる特徴ベクトルを、すべてのキーポイントの局所近傍から計算する。記述子は、それに対応するキーポイントをイメージ内の他のすべてのキーポイントに対して高い確率で識別するために、非常に特徴的でなければならない。さらに、記述子は、ノイズ、キーポイント識別エラー（たとえば、欠けているキーポイントまたは余分なキーポイント）、カメラ対ターゲットイメージの配置、および一般的イメージ劣化に対して頑健 (robust) でなければならない。記述子をスケール不変および回転不変にするために、スケール正規化されたイメージ近傍を選択し、その主方位を判定し、適用して、イメージをキーポイントの周りで回転して、イメージを位置合せされた状態にする。類似する尤度の複数の可能な主方位がある場合には、複数の記述子を単一のキーポイントから生成してもよい。

20

【0008】

(3) 最後に、クエリ時に、クエリイメージの記述子ベクトルを、コレクション内のすべてのさまざまなイメージの記述子ベクトルと比較して、可能な一致を判定する。照合は、通常、L1 距離または L2 (ユークリッド) 距離など、2つの特徴ベクトルの間の距離尺度に基づく。多くの場合、2つのイメージについて正確に同一のキーポイント順序を入手することは、不可能である（たとえば、2つのイメージがお互いに関して任意に回転される場合）。したがって、記述子の追加のソートおよびインデクシングが適用されない限り、すべての可能な記述子の対の組合せを比較する必要がある。所望の感度に依存して、通常のイメージが、数千個のキーポイント（および記述子）を生じる場合がある。したがって、記述子次元数（特徴の個数）を最小にすることが望ましい。というのは、記述子次元数が、性能（距離を計算するのに要する時間）に直接に影響するからである。

30

【0009】

上述のように、キーポイント識別の技術の幅広い選択肢が、文献に既に存在する。とりわけ、ガウス微分、モーメント、複雑特徴 (complex feature)、ステラブル特徴 (steerable filter)、およびフェーズ特徴 (phase feature) を含む、さまざまな手法に基づくさらに幅広くさまざまな記述子が提案されてきた。非特許文献1で導入された特徴記述子の一つの特定のクラスは、正確さおよび速度に関してほとんどの他の記述子より性能が優れていることを実証された。このクラスの記述子（すなわち、SIFT 記述子）は、キーポイント近傍内の局所的な小スケールの特徴の分布に基づく。SIFT 記述子は、キーポイントの周囲の 4×4 グリッド内の8つの異なる方位で局所空間密度勾配のヒストグラムを計算し、その結果を 128 次元ベクトルに格納する。

40

【先行技術文献】

【非特許文献】

【0010】

【非特許文献1】ロウエ (D. G. Lowe), 「Distinctive Image Features from Scale-Invariant Keypoints」, International Journal of Computer Vision, 2004年, 第6

50

0 巻 (Vol. 60) , 第 2 号 (No. 2) , p . 9 1 - 1 1 0

【非特許文献 2】ケ (Ke) およびスクトハンカー (Sukthankar) , 「PCA-SIFT: A More Discriminative Representation For Local Image Descriptors」, Conference on Computer Vision and Pattern Recognition , 2 0 0 0 年 , p . 1 1 1 - 1 1 9

【発明の概要】

【発明が解決しようとする課題】

【0011】

本発明は、テキストおよび線描などの追加のタイプのイメージ内容を利用することでもできるが、大きいイメージコレクション内の類似するイメージの発見または照合を対象とする。自然なピクチャの発見は、潜在的に、コレクションのテキストイメージまたはライン
10 アートイメージの発見または照合より難しい問題である。というのは、そのようなピクチャの内容が、輝度/グレイスケール領域で連続的であり、したがって、頑健で信頼できる
キーポイントを識別することがはるかにむずかしいからである。

【0012】

すべての使用可能な方法の中で、SIFT記述子は、最も広く使用されていると思われる。SIFT記述子は、適度な個数のイメージの照合のために比較的高速に計算される別個の記述子を提供する。しかし、SIFT記述子の高い次元数は、大きいイメージコレクションを用いるリアルタイムの応用例での使用についてSIFT記述子を非実用的にする。
20

【課題を解決するための手段】

【0013】

本発明の一態様は、イメージコレクションの内容を前処理して、前記イメージコレクション内のイメージごとに安定した再現可能キーポイントを識別するステップと、前記キーポイントの局所グループから指紋情報を計算するステップと、前記計算の結果の指紋情報をコンパクトな指紋データベースに格納するステップと、前記コンパクトな指紋データベース内の入力指紋ごとに、候補指紋の組合せの列を生成するステップと、指紋データベースに入力された指紋ごとの候補指紋の組合せの前記列を、ファンツリーおよびハッシュテーブルの少なくとも一方と、対応する指紋データ構造と、に格納するステップと、キーポイントを識別することと、クエリイメージから指紋を計算することと、クエリ指紋を既存のファンツリー指紋データと照合することとによってリアルタイムイメージクエリを実行し、前記コレクション内の一致するターゲットイメージまたはターゲットイメージのセットを判定するステップと、前記判定ステップに基づいて少なくとも1つのターゲットイメージを取り出すステップと、前記取り出された少なくとも1つのターゲットイメージの表示、印刷、格納、または送信のうちの少なくとも1つを行うステップと、を含み、前記指紋情報の前記計算が、アンカキーポイントと呼ばれるイメージ内の最も強いキーポイントのサブグループと、加えて複数の局所的な非アンカキーポイントと、の組合せに基づく、ことを特徴とする、イメージデータを格納し、インデクシングし、検索し、および/または取り出すイメージ管理方法である。
30

【0014】

前記クエリ指紋の前記既存のファンツリー指紋データとの前記照合は、クエリ指紋ごとに、オリジナルの指紋のイメージコレクションのサブセットを含む特定のファンツリー葉ノードリストにつながるファンツリーアドレスを計算することと、その後、前記サブセットリストからの各オリジナル指紋を順に現在のクエリ指紋と比較することと、前記クエリ指紋が前記取り出された指紋レコードのうちの1つと一致するたびに、そのイメージの一致する指紋の個数のカウントを増分することと、指紋スコア分析モジュールによって、各イメージの一致する指紋の結果の累算されたカウントを調べることと、最高スコア分析に基づいて、前記コレクション内の最もよく一致するイメージまたはイメージのセットを判定することと、を含んでいてよい。
40

【0015】

また、前記アンカキーポイントは、前記クエリイメージおよびターゲットイメージ内で
50

見つかる確率が前記非アンカキーポイントより高い。

【0016】

本発明の他の一態様は、クエリ文書のクエリイメージ内のキーポイントおよび潜在的ターゲット文書のコレクションの潜在的ターゲット文書イメージ内のキーポイントを含む、イメージ内のキーポイントを識別するように構成された識別モジュールと、前記キーポイントから指紋情報を生成するように構成された指紋生成モジュールであって、前記指紋情報が、少なくとも1つのアンカキーポイントおよび所定の個数の非アンカキーポイントを含む、指紋生成モジュールと、前記クエリイメージの前記指紋情報を、潜在的ターゲット文書の前記コレクション内の前記潜在的ターゲット文書イメージの指紋情報と比較するように構成された比較モジュールと、前記クエリイメージの前記指紋情報と前記潜在的ターゲット文書イメージの前記指紋情報との間の最良一致を判定するように構成された照合モジュールと、前記照合モジュールにおける判定に基づいて少なくとも1つのターゲット文書イメージを取り出すように構成された取出しモジュールと、前記取り出された少なくとも1つのターゲット文書イメージの表示、印刷、格納または送信を行う、電子ディスプレイ、プリンタ、メモリストレージ、またはコンピュータネットワークのうちの少なくとも1つと、を含むことを特徴とする、イメージデータの格納、インデクシング、検索、および/または取出しを行う、コンピュータ動作可能システム内のイメージ管理システムである。

10

【図面の簡単な説明】

【0017】

20

【図1】本発明の実施形態の概念を実施できる環境を示す図である。

【図2A】本発明の実施形態によるシステムおよび方法を使用する安定したキーポイント識別動作の応用を表し、スクリーンに表示されたクエリカメライメージを示す図である。

【図2B】本発明の実施形態によるシステムおよび方法を使用する安定したキーポイント識別動作の応用を表し、スクリーンに表示されたクエリカメライメージのキーポイントの識別された位置を示す図である。

【図2C】本発明の実施形態によるシステムおよび方法を使用する安定したキーポイント識別動作の応用を表し、対応するターゲットイメージのキーポイントの識別された位置を示す図である。

【図2D】ターゲットイメージを表す図である。

30

【図3A】2D(2次元)視覚的指紋を用いた文書発見のための文書コレクションを準備する方法の概要を示す図である。

【図3B】2D視覚的指紋を用いた文書発見のための文書コレクションを準備する方法の概要を示す図である。

【図4】2D視覚的指紋を用いた文書発見の継続されるクエリ動作を示す図である。

【図5】文書コレクションの指紋データベースを生成する方法を示す図である。

【図6】ターゲット文書内の候補キーポイントを識別する方法を示す図である。

【図7】ターゲットイメージ内の候補アンカキーポイントを識別する方法を示す図である。

。

【図8A】指紋の構成のために三角形比を入手する方法を示す図である。

40

【図8B】図8Aの三角形比の量子化ステップを示す図である。

【図9】解凍された指紋データ内容を示す図である。

【図10】アンカキーポイントおよび非アンカキーポイントの組合せから圧縮された指紋データベースを生成する方法を示す図である。

【図11】指紋データベースからファンツリーを作成する方法を示す図である。

【図12】ファンツリー構造および対応する指紋データ内容を示す図である。

【図13】文書コレクション指紋データベースからファンツリー構造を作成するステップを示す流れ図である。

【図14】リアルタイムでクエリイメージを照合するステップを示す流れ図である。

【図15】クエリ候補キーポイントの識別を示す流れ図である。

50

【図 1 6】クエリ指紋化のステップを示す流れ図である。

【図 1 7】クエリ指紋の照合および文書ヒットのカウントのステップを示す流れ図である。

【発明を実施するための形態】

【0018】

本明細書で説明されるシステムおよび方法は、図 1 に示されたコンピュータネットワークのパラメータ内で働くことができる。コンピュータネットワーク 100 は、一連のワイヤ 102 からなるものとして行うことができ、ワイヤ 102 の多くは、分岐するかワイヤジャンクション 104 で第 3 のワイヤ 106 と接続されてもよく、独立型周辺デバイスに接続されるか、コンピュータ 108, 109 などの他のデバイスに接続するために周辺機器を通過してもよく、ここで、コンピュータは、周辺デバイスと考えることができる。ネットワークは、カラープリンタ 110 またはカラー以外のプリンタ 112、ならびに少なくともカラーレーザプリンタ 120, 122 または 1 台のカラー以外のレーザプリンタ 124 を組み込んでよい。ネットワークは、スキャナ 130、またはファクシミリ機 140、写真複写機 150、カラー写真複写機 152、あるいは組合せカラープリンタ/スキャナ/ファクシミリ機 154 を組み込むこともできる。ネットワークは、パーソナルコンピュータおよび/または独立型コンピュータ端末 160、あるいは独立型ハードドライブデータストレージ媒体 164 を含むこともできる。ネットワークは、無線ネットワーク送受信器 170 を含み、少なくとも 1 つのラップトップコンピュータ 172 または複数のラップトップコンピュータ 174 とインターフェースすることもできる。ネットワークは、インターネット、イントラネット、または他の通信ネットワークを含むがこれらに限定されない任意の形のネットワーク 180 に相互接続されてもよい。別の形のネットワークとのインターフェースの使用を介して、本システムおよび方法は、デジタルスチルカメラ 191、デジタルビデオカメラ 192、セル電話機 193、スキャナ 194、携帯情報端末 195、または文書インデクシングシステム 196 を含むがこれらに限定はされない複数の周辺データ取り込みデバイス 190 とインターフェースすることができる。本概念を、単一のデバイスを有するネットワークから数千個以上の接続されたデバイスを含むネットワークにいたる、上記のコンポーネントのさまざまな組合せを有するネットワーク内で実施できることを理解されたい。さらに、上記のコンポーネントのうちのさまざまなコンポーネントは、説明される概念を実施するのに有用である可能性がある複数の既知の構成のいずれかに配置されたメモリストレージエリアを有することができる。このストレージエリアは、RAM (Random Access Memory)、ROM (Read Only Memory)、フラッシュメモリ、または本願の概念を組み込んだソフトウェアを保持できる他のメモリタイプとすることができる。他のメモリストレージエリアを、複数のデータベースフォーマットのいずれかにさまざまなデジタルイメージを保持するように構成してもよい。

【0019】

コンピュータなどであるがこれに限定はされない図 1 のコンポーネントのうちのさらなるさまざまなコンポーネントは、そのコンポーネントにロードされるか他の形でそのコンポーネントによってアクセス可能なソフトウェアからの命令を処理するプロセッサを含む。プロセッサを有するコンポーネントのうちのさまざまなコンポーネントが、複数のプロセッサを有してもよく、これによって、命令の処理を複数のプロセッサの間で分割できることを理解されたい。代替案では、単一のプロセッサが、命令を分割するように動作でき、これによって、処理をマルチスレッド環境で行うことができる。

【0020】

本発明の実施形態では、既存の方法およびシステムで使用される記述子の必要を完全になくす代替の方法およびシステムが展開される。その代わりに、本発明の実施形態の方法およびシステムは、キーポイント位置から直接に計算される局所化された 2 次元指紋を使用する。指紋は、記述子よりはるかにコンパクトであり、対応するキーポイントの識別において同様に有効である。記述子とは異なって、指紋は、非常に大きいイメージコレクシ

10

20

30

40

50

ョンサイズまでたやすくスケーリングされる。さらに、指紋は、回転およびアフィン変換 (affine transformation) に対して不変になるように設計され、したがって、SIFTのように特殊な方位処理を必要としない。

【0021】

ピクチャコレクション内の類似するピクチャまたはピクチャセットを発見するための指紋の使用は、次の2つのステップからなる。

【0022】

1. 前処理ステージで、ターゲットピクチャのコーパスを分析して、各イメージから数百個の最も特徴的な指紋を抽出する。イメージの非常に大きいコーパスについて、低い確率で、所定の指紋が複数のターゲットピクチャ内で見つかる可能性がある。指紋は、高い次元 (約35) を有する。指紋は、ハッシュテーブルにハッシュ化され、このハッシュテーブルのエントリは、リンクされた指紋レコードのリストを含む。代替案では、ファンツリーを構成し、指紋をそれにインデクス付けすることができる。各指紋レコードは、特定の指紋のアイデンティティ、その値 (指紋ストリング)、およびその指紋を含むターゲットピクチャのアイデンティティを含む。指紋の値は、ハッシュテーブル衝突が許容される時に、同一のハッシュテーブルエントリにハッシュ化され得る異なる指紋の間で区別するために、ハッシュテーブル衝突の場合にのみ必要である。衝突は、メモリ内のハッシュテーブルサイズを制限した結果として発生する可能性がある。

【0023】

2. クエリ時に、指紋がクエリイメージから抽出される。順次、各指紋が、ファンツリーまたはハッシュテーブルの内容を検索するためのキーとして使用されて、ターゲットピクチャの関連する候補の指紋レコードが取り出される。そのような指紋レコードのそれぞれについて、潜在的なファンツリーまたはハッシュテーブルの衝突のゆえに、クエリの指紋ストリングが、ターゲットピクチャの指紋ストリングと比較される。特定のターゲットピクチャの指紋ストリングとの正確な一致がある場合には、そのターゲットイメージの投票カウントを増分する。多数の投票が、クエリピクチャと候補ターゲットピクチャとの間の正しい一致について生じ、少数の投票が、少数の指紋の偶然の一致から生じる不正な一致について生じると期待される。

【0024】

図2A～図2Dに、本発明の実施形態の方法およびシステムを用いて成功したイメージクエリの検索を示す。図2Aでは、クエリイメージが、カメラ、スキャナ、または他のそのようなデバイスによって取り込まれ、あるいは、図2Dのイメージなどのターゲットイメージの格納された電子版から複写される。図2Aにおける手による部分的な妨害は、クエリイメージの一部が他の物体によって部分的にさえぎられる状況を含めて、クエリイメージがある形で劣化する状況であっても本願の概念を達成できることを強調することを意図されたものである。結果のクエリキーポイント位置を、図2Bに示す。クエリ指紋のセットが、クエリイメージのキーポイント位置から計算され、イメージコレクションからの指紋のデータベースと照合される。図2Cに、ピクチャコレクション内のターゲットイメージについて見つかったキーポイントを示す。

【0025】

より具体的には、図2Aのイメージに対する本発明の実施形態のシステムおよび方法の動作によって得られるキーポイント位置を、図2Bに示す。3つのスケールで入手されたキーポイント位置が示され、最小スケールの位置は、「x」を用いてマークされ、中間スケールの位置は、「」を用いてマークされ、最大スケールの位置は、「o」を用いてマークされている。最小スケールの位置は、最小サイズのフィルタの使用によって識別され、中間スケールの位置は、最小サイズのフィルタのスケールアップ版 (スケールをより大きくしたバージョン) の使用によって識別され、最大スケールの位置は、最大サイズのフィルタの使用によって識別される。同様に、図2Dのターゲットイメージに対する本発明の実施形態のシステムおよび方法の動作によって得られるキーポイント位置を、図2Cに示す。したがって、図2Bおよび2Cは、本発明の実施形態のシステムおよび方法の動作

10

20

30

40

50

の結果を示す。

【 0 0 2 6 】

本発明の実施形態の方法は、図 3 A ~ 図 3 B および図 4 に概要を示された 3 つの処理ステップからなる。図 3 A ~ 図 3 B および図 4 が、本システムおよび方法の概要を提示し、これらの図で導入される概念が、後のセクションでより詳細に説明されることを了解されたい。

【 0 0 2 7 】

(a) 図 3 A の流れ図 3 0 0 では、ピクチャコレクションの内容が、安定した再現可能なキーポイントを識別するために前処理される。ターゲットピクチャのコレクション 3 1 0 内のピクチャごとに、キーポイントを識別し 3 2 0、キーポイントごとに、指紋化動作 3 3 0 を実行することによって指紋を計算する。結果の指紋データ情報 3 4 0 は、コンパクトな指紋データベース 3 5 0 に格納される。

10

【 0 0 2 8 】

一般に、テキストまたはラインアートの文書内よりも、連続トーンの画像内で安定したキーポイントを見つけることの方がむずかしいので、本発明の実施形態では、イメージ内の少数の最も強いキーポイント（アンカキーポイントと呼ばれる）と、加えて複数の他の（非アンカ）局所キーポイントとの組合せに基づく指紋化方法を詳細に述べる。その強さに起因して、アンカキーポイントは、クエリイメージとターゲットイメージとの両方で高い確率で見つかる可能性が高い。アンカポイントの使用は、近傍内の少数のより強いアンカキーポイントと共に、より弱い局所キーポイントと一緒に「グループ化する」ことによって、指紋組合せの総数をかなり減らすのを助ける。

20

【 0 0 2 9 】

(b) 次に、図 3 B に、指紋データベース 3 5 0 を編成するのに使用される流れ図 3 5 5 を示す。このプロセスでは、さまざまな指紋組合せ 3 6 0 を有する情報が、ファンツリーアドレッシング方法 3 7 0 に従ってファンツリーに編成される。指紋データベース 3 5 0 内の入力指紋ごとに、高速で効率的な指紋照合を容易にするために、候補指紋組合せ 3 6 0 の列が生成され、ファンツリー 3 7 0 および対応する指紋データ構造 3 8 0 に格納される。指紋組合せの列の目的は、クエリイメージのアフィン変換または透視変換などの可能な変換に起因して複数の（非アンカ）キーポイントが欠けることを可能にし、なおかつ、高い正確さで指紋を正しく検出できるようにすることである。

30

【 0 0 3 0 】

(c) 図 4 に、特定のクエリイメージ 4 1 0 に関するリアルタイムイメージクエリの実行 4 0 0 を示す。これは、特定のクエリイメージ 4 1 0 内のキーポイント位置を識別し（ 4 2 0 ）、クエリキーポイントの局所グループからのクエリキーポイントごとに指紋情報を計算し（ 4 3 0 ）、クエリ指紋 4 4 0 と既存のファンツリー指紋データ 4 8 0 を照合し、コレクション内で最もよく一致するイメージまたはイメージのセットを判定することで行われる。各クエリ指紋は、ファンツリーアドレスを計算すること（ 4 5 0 ）に用いられる。ファンツリーアドレスは、オリジナルのイメージコレクション指紋レコード 4 8 5 のサブセット（下位集合）を含む特定のファンツリーの葉のノードリストにつながる。サブセットリストからの各オリジナル指紋は、順次、指紋照合 4 6 0 のために現在のクエリと比較される。クエリ指紋が取り出された指紋レコードのうちの 1 つと一致するたびに、そのイメージの一致する指紋の数のカウントを増分する。さらに、指紋の強さおよびそれがターゲット指紋にどれほど一意に相関するかに基づいて、指紋の一致のそれぞれに対してスコアを割り当てる。最後に、指紋スコア分析モジュール 4 9 0 が、イメージごとに、一致する指紋 4 7 0 の結果の累算されたスコアおよびカウントを調べて、クエリイメージ 4 1 0 に最もよく一致する、コレクション内の最良一致イメージまたは最良一致イメージのセット 4 9 5 を判定する。総合スコアを、指紋一致の個数とその相対スコアとを組み合わせることによって、一致する文書ごとに計算する。応用例に応じて、指紋スコア分析モジュール 4 9 0 は、最高の総合スコアを有する単一の文書を選択してもよく、あるいは、その代わりに、所定の値より大きい総合スコアを有するすべての文書を選択してもよい。次

40

50

に、図 1 のコンポーネントのうちの 1 つによって、一致する文書セットを取り出し、プリントアウトし、電子的に送信し、表示し、および / または格納することができる。

【 0 0 3 1 】

I . a . イメージコレクション前処理

イメージコレクション前処理の流れ図 5 0 0 を、図 5 に示す。ターゲットイメージ 5 1 0 を、一時に 1 つずつ順次処理して、キーポイント識別に基づいてその視覚的指紋情報を抽出する。

【 0 0 3 2 】

キーポイント識別処理モジュール / ステップ 5 2 0 では、各ターゲットイメージ 5 1 0 を処理して、入力イメージ内容に基づいてキーポイントのセットを識別する。この方法の成功は、きれいなターゲットイメージ内と、ある一般的なひずみおよび変化する照明条件の下でのそのイメージの劣化した版とでほぼ同一のキーポイントを見つける能力に依存する。キーポイント識別モジュール 5 2 0 の出力は、キーポイントと、スケール、ページ上の位置、および相対強さなどのそれに関連する属性とのリストである。

【 0 0 3 3 】

次に、指紋構成処理ステップ 5 3 0 で、局所近傍内のキーポイントの小さいグループから指紋を計算する。局所化された指紋の使用は、特にカメライメージの配置の要件および全体的な照明の要件の緩和に関して、この方法の柔軟性および頑健性 (robustness) に大いに寄与する。結果の指紋は、指紋データベース 5 5 0 に格納され、指紋データベース 5 5 0 では、各イメージが、ターゲットイメージ 5 1 0 に対応する一意イメージ ID 5 4 0 を有する。

【 0 0 3 4 】

I . a . 1 イメージキーポイントの検出

キーポイント識別モジュール (たとえば、図 5 の 5 2 0) の目標は、ノイズと、シーン照明変動と、スキュー、ワープ、回転、並進、スケール、解像度の変化などの一般的ひずみと、を受ける劣化したイメージにおいてさえ繰り返して信頼できる形で見つけれられる可能性が高い、連続トーンピクチャイメージ内の安定したキーポイントを識別することである。

【 0 0 3 5 】

この方法は、複数のスケールでの入力イメージのスケール空間ピラミッド表現を構成する。各スケールで、特定のフィルタのセットが入力イメージに適用され、ここで、各連続するフィルタは、前のフィルタのスケールアップ版である。フィルタ出力は、スケールおよび空間の単一の関数 $D(x, y, s)$ を形成するために一緒に組み合わせられる。安定したキーポイント位置は、関数 $D(x, y, s)$ がスケール空間近傍内で局所ピーク値 (最大値または最小値のいずれか) を達成するピクセル位置として、各スケール内で識別される。

【 0 0 3 6 】

本願の一実施形態では、入力イメージ内で識別されるすべての使用可能なキーポイントが使用される。もう 1 つの実施形態では、使用可能なキーポイントが、相対強さと、キーポイントスケールおよび / または他のキーポイントへの近接などの他の考慮事項とによって、また、出力すべき最良のキーポイントのうちの固定された個数だけを選択することによって、ソートされる。さらに、第 3 の実施形態では、入力イメージ区域 5 1 0 を (オーバーラップするまたはオーバーラップしない) 領域のセットに下位分割することができ、複数の好ましいキーポイントが、ある領域が他のすべての領域に優位を占めるのを防ぐために、各領域から選択される。

【 0 0 3 7 】

最後に、キーポイント識別モジュール 5 2 0 の最後の処理ステップは、2 つのキーポイントが所定の公差レベル内で正確に同一の位置を有することがないことを保証するために、妥当性検査ステップを含む。この妥当性検査ステップは、すべての重複する位置のキーポイントを、この位置での最も強く最も望ましいキーポイント (その強さ、スケール、お

10

20

30

40

50

よび／または他のキーポイントへの近接に依存する)だけを保持し、他のすべての同一位置のキーポイントを除去することによって、除去する。キーポイント識別モジュール520の出力での残りのキーポイントのリストが、最終候補キーポイントリストになる。

【0038】

あるイメージ内で識別できるキーポイントの全体的な個数は、もちろん、入力イメージ内容と、キーポイントの強さのしきい値などのキーポイント識別パラメータの選択と、に依存する。通常の自然シーンイメージには、概して数百個から数千個までのキーポイントがあり得る。

【0039】

ここで図6に移ると、ターゲットイメージ610の文書キーポイント位置を検出するプロセス600が示されている。適応しきい値の設定、ブラーリング(blurring)、および連結成分コレクションのイメージ処理は、文献で周知である。最良の結果のためには、人間の視覚系の特性に従う知覚イメージ空間内でのキーポイント識別プロセスを実行することが望ましい。

【0040】

第1の適応しきい値モジュール620のバイナリ(二値)出力イメージ625は、CCサイズ推定モジュール630に送られる。ここで、用語「CC」は、連結成分(Connected Component)を表し、これは、同一極性を有するバイナリピクセルの最大限に接続されたサブグループである。2つのピクセルは、それらの間に同一極性のピクセルの経路が存在する場合に限って、同一の連結成分に含まれる。CCサイズ推定モジュール630の目的は、ターゲットイメージ610について、イメージごとに、後続のラジアルブラーモジュール640で適用されるブラーフィルタサイズなどのブラーパラメータ635を動的に推定することである。ブラーリングプロセスの目的は、頑健で信頼でき再現可能なキーポイント識別を提供することである。ブラーリングは、ごま塩ノイズなどのノイズを除去し、ページ上の小さい分離された特徴を除去するのも助ける。たとえば近似ガウシアン形状(approximated Gaussian shape)であるがこれに限定はされないブラーリングフィルタの形状は、望ましくないアーティファクトを導入するのを防ぐのに十分に滑らかでなければならない。

【0041】

正しい量のブラーリングを判定する際の1つの問題は、入力ページ上の通常のテキスト文字のサイズが事前にはわからないことである。本願の一実施形態は、10ポイントから14ポイントまでのRomanフォントなど、ある範囲の標準タイプフォーマットにまたがるよい妥協を提供する、固定され前もって事前に決定されたラジアルブラーパラメータ635を開示する。本願の第2の実施形態では、ラジアルブラーパラメータ635は、ユーザが文書ごとに適用すべき最適セッティングを指定できるように、選択可能にされる。これらの実施形態のどちらでも、CC推定モジュール630は使用されず、その出力は、選択された事前定義のラジアルブラーパラメータ635によって置換される。

【0042】

本願の第3の実施形態では、CC推定モジュール630は、入力ターゲットイメージ610内容を調べることによって最適ラジアルブラーパラメータを自動的に推定するのに使用される。この場合に、CC推定モジュール630は、適応しきい値620モジュールからのバイナリ出力625を使用する。CC推定モジュール630は、バイナリイメージ625を処理して、連結成分の要素を集め、連結成分の高さのヒストグラム化に進む。というのは、文字の高さは文字の幅より変化が少なく、ほとんどのロマンス語で文字の幅と比較して文字の高さの方がフォントサイズをよりよく示すからである。高さのヒストグラムは、発生の頻度に基づいて、適当なラジアルブラーパラメータを判定するのに使用される。このプロセス中に、CC推定モジュール630は、寸法が小さ過ぎる(たとえば、少数の分離されたピクセルからなる)または寸法が大き過ぎる(たとえば、長い線またはフレームにページの中で出会う時)時など、ノイズによって誘導される連結成分を除去するよう取り計らい、その結果、提供される推定値が、少数の異常に大きいまたは異常に小さい

10

20

30

40

50

連結成分要素によって偏ることがないようにする。さらに、ページは、フォントサイズの混合された寄せ集めを含む場合があるので、CC推定アルゴリズムは、混合されたフォントセットの場合であってもよい妥協を有する保守的な推定値を提供するように設計されている。

【0043】

本願の第4の実施形態では、CCサイズ推定モジュール630は、局所化されたサブイメージの領域内の入力イメージの内容を分析することと、領域ごとに（たとえば、主に小さいポイントのテキストを含む1つのイメージ区域からより大きいフォントを含む別の近くの区域に動的に移動する時）プラーリングパラメータを適応的に変更することとに基づいて、適応プラーリングパラメータを動的に提供することができる。

10

【0044】

ラジアルプラーモジュール640の出力は、グレイスケールイメージ645である。適応しきい値モジュール650は、ラジアルプラー640からのグレイスケールのイメージ出力645を、適応しきい値設定650によってバイナリ白黒出力655に変換する。

【0045】

第2適応しきい値モジュール650のバイナリ出力は、バイナリイメージ655であり、連結成分モジュール660に転送される。連結成分法は、当技術分野で周知であり、グラフの最大限に接続された成分のセットと考えることができる。

【0046】

本願の第1の実施形態では、使用可能な連結成分のすべてが使用される。第2の実施形態では、使用可能な連結成分が、たとえば、連結成分寸法の最適化、ピクセルカウント、アスペクト比、および/または他の連結成分への近接に重みを与えることによって、相対強さによってソートされ、連結成分のより小さいサブセットだけが出力される。第3の実施形態では、入力ターゲットイメージ610を、オーバーラップするまたはオーバーラップしない領域のセットに下位分割することができ、複数の連結成分を、ある領域が他のすべての領域に優位を占めるのを防ぐために、各領域から選択することができる。

20

【0047】

重心計算モジュール670は、連結成分モジュール660の出力での各連結成分の視覚的中心を判定する。連結成分ごとに、水平重心位置は、連結成分の各メンバピクセルの水平座標を一緒に合計し、その結果をメンバピクセルの総数で割ることによって計算される。垂直重心位置は、各メンバピクセルの垂直座標を一緒に合計し、メンバピクセルの個数で割ることによって、同様に計算される。合計は、効果的に、連結成分分析中にオンザフライで行うことができる。所定の連結成分のピクセルメンバだけが、すべての他の非メンバのピクセルの「穴」を無視して、その重心位置に寄与することに留意されたい。各連結成分の視覚的重心は、サブピクセル精度で計算される。というのは、多くの言語で、連結成分が、テキスト行内に配置される傾向があるからである。

30

【0048】

キーポイント識別フェーズの最後の処理ステップでは、重心計算モジュール670からの連結成分重心位置のリスト675が、重複除去モジュール680によって妥当性検査され、重複除去モジュール680は、キーポイントのリスト685を作る。この妥当性検査の目的は、2つの連結成分が所定の公差レベル内で同一の重心位置を有しないことを保証することである。ほぼ同一の重心位置を有する重複した連結成分が、除去される。

40

【0049】

重複除去モジュール680の出力での残りの連結成分重心のリストが、最終候補クエリキーポイントリスト695になる。候補キーポイント690の総数は、入力イメージ内容と、連結成分処理のタイプとに依存する。通常の機械印刷されたページについて、数百個のキーポイントがあり得る。

【0050】

I. a. 2. 指紋の構成

このセクションでは、キーポイントの局所グループから指紋を計算し、指紋データベ

50

ス（たとえば、図5の550）での効率的な格納のために指紋を圧縮するプロセスを説明する。指紋は、指紋データベースサイズ要件および格納要件を減らすためにパックされる。

【0051】

広範囲のノイズ、プューイング条件、およびイメージひずみにまたがって安定な、頑健な2D視覚的指紋の識別が、入力イメージ内で探される。さらに、システムが非常に大きいイメージコレクションサイズ（たとえば、数百万個のイメージを含む）を効率的に処理するようにスケールアップすることを可能にするために、指紋サイズを最小にすることが有用である。というのは、指紋データベースが、コレクション内の有効な指紋のすべてからなるからである。それと同時に、指紋は、それが表す個々のイメージシグネチャを高い

10

【0052】

指紋は、パーシスタント・レシオ（persistent ratio）と呼ばれる、量子化された変換不変の2Dの比の列として構成され、パーシスタント・レシオは、所定のキーポイントとその（N-1）個の最近傍キーポイントとの相対的な2D位置から導出される。したがって、各指紋は、処理対象のキーポイントの周囲の小さいイメージ近傍に局所化される。指紋の列は、処理対象のキーポイントとその（N-1）個の最も近いキーポイント近傍との間の相対的な2D配置だけに依存する。近傍の個数Nは、指紋の強さに影響する設計パラメータである。

【0053】

20

指紋化法は、連続トーンイメージの場合に拡張される。ある種の単語重心特徴がバイナリイメージ内の頑健なキーポイントを識別するのに利用される機械印刷された文書の場合とは異なって、連続トーン画像内で安定したキーポイントを見つけることは、かなりよりむずかしい。したがって、機械印刷された文書に見られるキーポイントと比較して、より高い可変性が連続トーンイメージに見られるキーポイントに関して期待される。

【0054】

連続トーンイメージに関する通常の問題は、2つのイメージの間のキーポイントの対での照合である。機械印刷された文書では、単語重心は、ノイズおよびイメージひずみに対して非常に頑健であり、両方のイメージ内に高い確率で現れる可能性が高い。対照的に、連続トーンイメージ内の顕著なスケール空間キーポイントの一部、特により弱いキーポイント

30

【0055】

連続トーンイメージに関する通常の状態は、キーポイントのほとんどを、クエリイメージとターゲットイメージとの両方で高い確率で共通して見つけることができるが、時々、ターゲットイメージ内の対応する一致するキーポイントを有しない、クエリイメージ内の少数の欠けているまたは余分なキーポイントがある場合があることである。欠けているキーポイントとは、ターゲットイメージには見られるがクエリイメージには見られないキーポイントである。余分なキーポイントとは、クエリイメージには見られるがターゲットイメージには見られないキーポイントである。

【0056】

40

クエリイメージとターゲットイメージとの間の、個数が可変の欠けているキーポイントまたは余分なキーポイントの存在は、指紋の照合結果をすばやく劣化させる可能性がある。複数の要因に起因して、余分なキーポイントまたは欠けているキーポイントは、複数の指紋に影響する可能性が高い。というのは、所定のキーポイントが、通常は複数の局所的な指紋組合せに用いられるからである。

【0057】

提示されるのは、イメージ内の少数の最も強いキーポイント（アンカキーポイントと呼ばれる）と、加えて他の（非アンカ）局所キーポイントとの特定の組合せに基づく新しい指紋化法である。その強さに起因して、アンカキーポイントは、クエリイメージとターゲットイメージとの両方で高い確率で見つかる可能性が高い。アンカポイントの使用は、近

50

傍内の少数のより強いアンカキーポイントと共により弱い局所キーポイントと一緒に「グループ化する」ことによって、指紋の組合せの総数をかなり減らすのを助ける。アンカキーポイントは、その相対的な強さおよび他のアンカキーポイントへの距離に基づいて、候補キーポイントのリストから選択される。このアルゴリズムは、距離に基づいて同一のまたはより小さい強さのキーポイントと一緒にグループ化し、グループ代表として最も強いグループキーポイントだけを保持するために、クラスタ化技法を使用する。このアルゴリズムは、グループにマージすべきさらなるキーポイントがなくなり、および/または前の反復からのグループ内距離の変化が所定のしきい値未満になるまで、反復される。さらに、最小距離クラスタ化パラメータが、イメージについて良いアンカキーポイントの範囲を得るのに使用される。次に、結果のグループ代表（すなわち、各グループ内の最も強いキーポイント）が、アンカキーポイントのセットとして選択される。

10

【0058】

指紋は、パーシスタント・レシオ（パーシスタント比，*persistent ratio*）と呼ばれる、量子化された変換不変の2Dの比の列として構成され、パーシスタント・レシオは、キーポイントのグループの相対的な2D位置から導出される。この実施形態では、欠けているキーポイントまたは余分なキーポイントに対して指紋を頑健にするために、指紋を構成するのに使用されるキーポイントの局所グループは、事前に決定される少ない個数のアンカキーポイントに加えて、局所キーポイントを含むように構成される。

【0059】

したがって、本発明の実施形態による指紋を形成するキーポイントのグループは、所定のキーポイント、少数のアンカキーポイント、および他の（非アンカ）局所最近傍キーポイントのセットからなる。指紋内のキーポイントの全体の個数およびアンカ対他のキーポイントの個数の比率は、指紋の強さおよび頑健性（*robustness*）に影響する2つのシステム設計パラメータである。

20

【0060】

アンカキーポイントおよび他の局所キーポイントの組合せから指紋を計算する特定の方法を、図7に示す。指紋は、キーポイント識別モジュールの出力での候補キーポイントごとに計算される。指紋がそれについて計算される現在のキーポイント位置（1）が、図7の中央に正方形パターンを用いて示されている。指紋は、（a）現在のキーポイント位置（1）、（b）現在のキーポイント位置に最も近い事前に決定された個数のアンカキーポイント（図7で星パターンによって示された2、3）、および（c）複数の最近傍非アンカキーポイント（図7で円パターンを用いて示された4から10まで）の2D位置から計算される。追加の非アンカキーポイント位置（図7の破線パターンの11から13までなど）は、現在のキーポイント位置（1）から事前に決定された距離より遠いので、指紋には含まれない。

30

【0061】

各指紋が、所定のキーポイント位置（1）に対して相対的に最も近いアンカキーポイントならびに非アンカキーポイントを選択することによって、処理対象のキーポイントの周囲の小さいイメージ近傍に局所化されることに留意されたい。本実施形態の特定の態様は、デジタルカメラ、セル電話機カメラ、または他のイメージ取り込みデバイスを用いるイメージのピクチャのスキャンまたは撮影のプロセス中に一般的に発生するある種のイメージひずみ（スキュー、ワープ、回転、並進、スケール、解像度の変化など）に対して指紋を頑健にすることにある。

40

【0062】

本発明の実施形態において、N個のキーポイントのグループの2D位置から指紋を計算する方法では、キーポイントの一部が、通常のキーポイントではなくアンカポイントになるようにさらに制約される。任意のアンカキーポイントおよび/または非アンカキーポイントの非同一直線上の組合せからなる $P = 4$ 個のキーポイント $\{A, B, C, D\}$ のグループごとに、三角形面積の比 (ABC / ACD) が、計算され、Q個のレベルに量子化される。Qの値は、2の2進べきになるように便利に選択することができる。量子化プロセス

50

を、 $Q = 8$ の場合について図 8 A ~ 図 8 B に示す。

【 0 0 6 3 】

図 8 A に示されているように、三角形 A B C 8 5 0 および第 2 の三角形 A C D 8 6 0 を含む、物体表面上の 4 つの非同一線上の点 { A , B , C , D } (すなわち、8 1 0、8 2 0、8 3 0、8 4 0) からなる任意の所定の多角形 8 0 0 について、三角形面積の比 (A B C / A C D) 8 7 0 は、任意のアフィン変換の下で一定のままになることが、当技術分野で周知である。したがって、 $P = 4$ 個の点だけが、三角形比 8 7 0 として示されている、1 つの三角形面積比を計算するのに必要である。このアフィン変換は、多くの実用的状況で、カメラ対平面物体ジオメトリを記述する受け入れられるモデルを提供することが示されている。

10

【 0 0 6 4 】

本願のもう 1 つの実施形態では、アフィンより高い変換次数がイメージモデルを記述するのに必要な状況について、その変換を、2 つの三角形比の積である単一のパーシスタント・レシオを計算するために $P = 5$ 個の点 (4 個ではなく) を使用する透視変換を処理するように拡張することができる。

【 0 0 6 5 】

したがって、単一の指紋は、時計回りの順番でソートされた N 個の最近近傍キーポイントのグループに関する量子化されたパーシスタント変換レシオの列からなる。指紋のサイズを小さく保つために、変換比 8 7 0 は、 Q 個のレベルに量子化される。一実施形態で、 Q の値は、2 のべき乗になるように便利に選択することができる。図 8 B では、量子化プロセスが、 $Q = 8$ の場合について図 8 A の三角形比 8 7 0 の量子化として図示されている。量子化の前の三角形比の有効範囲 (0 ,) は、図 8 B に示されているように ' 0 ' から ' 7 ' までのラベルを付けられた $Q = 8$ 個のインターバル (区間) に分割される (8 9 0)。インターバルの境界は、代表的な文書の大きいセットにわたって量子化ラベルがほぼ一様に分布するように経験的に決定される。入力される三角形比值 8 8 0 は、順次、増加するインターバル境界と比較されて、最初のより大きいまたは等しいインターバル境界が判定され、その時に、このプロセスは終了され、対応するラベルインターバルが、量子化された結果として割り当てられる。たとえば、1 . 0 より小さく前のインターバル境界より大きい 0 . 8 2 5 3 6 9 4 の三角形比の入力は、量子化された値 ' 3 ' を割り当てられる。

20

30

【 0 0 6 6 】

三角形面積比を使用することの利益は、その値がアフィン変換の下で一定 (したがって、パーシスタント (永続的)) のままになることである。アフィン変換は、多くの実用的状況で、カメラ対平面物体ジオメトリを記述する受け入れられるモデルを提供することが示されている。

【 0 0 6 7 】

指紋照合中に発生し得る 1 つの問題は、クエリイメージが、照合されるターゲットイメージに対して相対的に異なる方位である場合があることである。クエリイメージがターゲットイメージに対して相対的に回転される時に、結果として得られる指紋の列は、2 つの指紋が数字の同一のセットを共有する場合であっても、列内の数字の異なる順序に起因してターゲットの列と一致しない可能性が高い。2 つの指紋は、指紋の列全体ですべての対応する量子化された比の数字それぞれの間に 1 対 1 の一致がある場合に限って一致する。

40

【 0 0 6 8 】

所定の指紋の量子化された比の特定の列は、そのキーポイントの特定の順序付けに依存する。たとえば、クエリイメージが、基準オリジナルに対して相対的に回転される場合に、開始キーポイントが異なる可能性があり、これは、量子化された比の完全に異なる列につながる。可能な指紋一致を判定するために、2 つの指紋を、まず、対応するキーポイントの同一の列の順にしなければならない。

【 0 0 6 9 】

上の問題を克服する 1 つの手法は、イメージ回転の結果として潜在的に発生する可能性

50

本願の実施形態では、並び替えられた指紋の列が、オリジナルの指紋の列の代わりに使用される。順列を使用することの利益は、照合する必要がある指紋の列の候補の数かなり減ることである。オリジナルの指紋の列のすべての可能な循環する組合せを調べるのではなく、最小の最も数少ない数字の少数の順列だけが調査される。順列は、指紋データベースの準備の間に前もって生成され、格納される。クエリの時には、クエリの指紋の列の最初の順列だけが、照合に使用され、追加のクエリの順列をチェックする必要はない。というのは、すべての順列が前もって格納され、正しい順列が自動的に見つけれられるからである。

【0078】

本願の実施形態による順列の使用は、順列の数が、平均してオリジナルの指紋の列の長さよりはるかに小さいので、わずかなストレージ増加について大きい性能向上をもたらす。

【0079】

本願のもう1つの態様は、欠けているキーポイントまたは余分なキーポイントの場合の指紋の頑健性を高めるために、複数（通常は2つ以上）の非アンカ近傍キーポイントが、欠けていることを許容されることである。しかし、より強いアンカキーポイントは、指紋一致を得るために、クエリイメージとターゲットイメージとの両方に存在することを要求される。

【0080】

本願の一実施形態の下では、小さい局所化された近傍内の限られたアフィンひずみの下で、3つまでのキーポイントを除外することが許容される。もちろん、システムでの考慮事項に依存して、3つを超えるキーポイントを除外することができる。したがって、各所定のキーポイントは、一回に3つの非アンカキーポイントを除外することによって、複数の指紋を生じる。各指紋は、方位の半径方向順で残りのアンカキーポイントおよび非アンカキーポイントを系統的にウォークすることと、P個の点（アフィン変換では $P = 4$ 、透視変換では $P = 5$ ）のすべての可能な組合せについて量子化されたパーシスタント・レシオの列を記録することによって作成される。図9に、 $N_a = 2$ 個のアンカキーポイント、 $N = 8$ 、 $P = 4$ 、および $Q = 8$ の場合の単一のイメージの指紋の出力の例を示す。各キーポイントは、この場合では長さ35の複数の指紋の列を生じる。

【0081】

指紋化プロセス1000を、図10に詳細に示す。指紋化プロセス1000への入力は、入力イメージ1005の候補キーポイントのリスト1010である。

【0082】

図10での第1ステップは、アンカキーポイント識別モジュール1015が、上で述べたように、キーポイント強さ、スケール、位置、および/または他のキーポイントへの近接などのさまざまな判断基準に基づいて入力イメージ1005内で見つかる候補キーポイントの中でアンカキーポイントの適切なサブセットを選択することである。

【0083】

指紋のセットを、候補キーポイントリスト1010内のキーポイントごとに計算する。各現在の候補キーポイント K_p を、次キーポイント取得モジュール1020によって入力リストから順番に選択する。候補キーポイントごとに、最近アンカキーポイント発見モジュール1030が、所定のキーポイント K_p までの最も近い距離を有する N_a 個の最も近いアンカキーポイントを識別する。次に、最近近傍発見モジュール1040が、所定のキーポイント K_p までの最も近い距離を有する $(N - N_a - 1)$ 個の最近非アンカキーポイントを識別する。返されたキーポイントを、 K_p からの増加する距離によってソートする。キーポイントの総数 N は、指紋の「強さ」（特徴性）と、全体的なシステム性能（指紋あたりの計算の回数）と、結果のデータベースサイズ（指紋サイズ）との間のよいトレードオフをもたらすように選択される。一実施形態では、値 $N = 8$ 、12、および16が使用される。

【0084】

時計回り順でのソートモジュール1050は、ターゲットイメージとクエリイメージとの間の量子化された面積比の一貫した列を提供するために、増加する時計回りの方位で所定のキーポイントの $N - 1$ 個の最も近いアンカおよび非アンカのキーポイントのリストをソートする。指紋ごとに、時計回り順でのソートモジュールは、現在のキーポイントおよび N_a 個の最近アンカキーポイントを使用して、サブグループの原点中心を計算する。非アンカキーポイントは、いくつかの余分な非アンカキーポイントまたは欠けている非アンカキーポイントの場合であっても原点が安定したままになることを保証するために、サブグループの原点中心の計算では使用されない。

【0085】

現在の近傍キーポイントクラスタの原点中心を判定した後に、時計回り順でのソートモジュール1050は、増加する時計回り方位順でのキーポイントのソートに進む。ソートは、方位と距離との両方で行われる。主順序は、増加する時計回り方位順による。しかし、複数の点が、おおむね同一（事前定義の公差レベル以内）の方位を有する場合には、点を、ほぼ同一方位のすべての点について増加する距離によって下位順序付けする。原点中心の位置は、現在のキーポイント位置からの矢印によって指摘される（たとえば、図7の位置1）。

【0086】

N 個の最も近いアンカキーポイントおよび非アンカキーポイントの順序付けを所定のキーポイントについて確立したならば、指紋を生成することができる。指紋は、近傍内のサイズ $P = 4$ のキーポイントの連続するサブセットから形成される。 N 個のキーポイントの P 個のサブセットが、次サブグループ組合せモジュール1060によって、系統的な一貫した形で考慮される。それぞれについて、 P 個のサブセットのパーシスタント面積比を計算し、その面積比を整数にマッピングすることによって、整数を判定する。所定のキーポイントの指紋の長さは、そのような P 個のサブセットの総数である。これは、 N 個のキーポイントから一意の P 個のキーポイントを選択するための組合せの個数によって決定される。たとえば、 $N = 8$ かつ $P = 4$ の場合に、可能なサブグループの組合せの個数は、70個のパーシスタント・レシオである。

【0087】

N 個のキーポイントの一意のサブセットのそれぞれについて、圧縮量子化比モジュール1070は、単一のパーシスタント・レシオを計算し、これを事前定義のしきい値のセットを使用して量子化する。量子化レベルの個数 Q は、設計パラメータであり、たとえば、 $Q = 8$ または $Q = 16$ が使用される。量子化しきい値は、特定のカテゴリのピクチャの大きいコレクション内のパーシスタント・レシオの分布を調査することによって経験的に決定される。

【0088】

指紋データベース1085のサイズをさらに減らすために、圧縮量子化比モジュール1070は、複数の結果の量子化されたパーシスタント・レシオを1つの機械語に圧縮(pack)する。たとえば、 $N = 8$ 、 $P = 4$ 、および $Q = 8$ を用いると、指紋全体（70個の組合せの列）を、4つ未満の64ビット語に密に圧縮することができる。本願の一実施形態では、圧縮された指紋のサイズは、合計3つの64ビット語および3つの8ビットバイトを占める（複数の語および/またはバイトにまたがって部分的情報を分割する必要なし）。

【0089】

指紋を計算し圧縮するプロセスは、最後の組合せが最終組合せモジュール1080によって検出されるまで、一時に1パーシスタント・レシオずつ順番に継続する。現在の P 個のサブグループの組合せが、まだ最後の組合せではない場合には（1081）、最終組合せモジュール1080は、フローを次サブグループ組合せモジュール1060に戻ってルーティングして、次の P 個のサブグループを入手し、その量子化されたパーシスタント・レシオの計算に進み、これを圧縮する。このプロセスは、最後の P 個のサブグループ組合せが処理される1082まで継続される。その時に、結果として得られる圧縮された指紋

データ1083が、指紋データベース1085に書き込まれる。圧縮された指紋インスタンスのそれぞれが使用可能になる時に、一回に1つの指紋ずつ、指紋データを指紋データベース1085に順次書き込むことができることに留意されたい。

【0090】

最後に、指紋を書き込むプロセスは、最後のキーポイントが最終キーポイントモジュール1090によって検出されるまで、残りのキーポイントのすべてについて順次継続される。現在のキーポイント組合せが、まだ最後のキーポイントではない場合(1091のNO)には、最終キーポイントモジュール1090は、フローを次キーポイント取得モジュール1020に戻って、次のキーポイントを入手し、その圧縮された指紋を計算するステップおよびそれを指紋データベース1085に追加するステップの繰り返しに進む。指紋化プロセスは、最後のキーポイントの組合せ(最後の対応する指紋)が処理され、指紋データベース1085に追加されるまで、この形で継続される。最後のキーポイントに対処したならば、このプロセスは終了する(1095)。

【0091】

I. b. 高速照合のための指紋情報の準備

図11に移ると、指紋データベース1110からファンツリーを作成するプロセス1100が示されている。指紋データベース1110内の指紋は、メモリサイズおよびローディング時間を減らすために、圧縮フォーマットで格納される。これらを使用できる前に、指紋は、まず、解凍され、効率的な指紋照合を容易にするためにファンツリーデータ構造に従って一意の形で編成される。最初のイメージクエリの際に、指紋データベースを1回だけ準備することが必要であることに留意されたい。ファンツリーデータ構造(解凍された指紋の列を含む)は、メモリ内に保持し、その後、任意の個数のイメージクエリについて再利用することができる。

【0092】

最初に、キーポイント列除外モジュール1120が、複数の指紋非アンカキーポイント1130を除外することによって、複数の指紋キーポイントの組合せの候補を選択する。これは、局所近傍キーポイントの中の1つまたは複数の欠けているまたは余分な非アンカキーポイントを可能にする。この実施態様では、キーポイント除外モジュール1120は、固定された個数 N_e 個の非アンカキーポイントを除外する。 $N - N_a$ 個の非アンカキーポイントの局所近傍を用いると、これは、データベースエントリ(またはキーポイント)ごとに複数の指紋を生じる。

【0093】

指紋列モジュール1150は、キーポイントごとに指紋の候補の列を生成する。たとえば、 $N = 8$ について、キーポイント除外モジュール1120は、最初の N_e 個の非アンカキーポイントを除外することと、残りの非アンカキーポイントを選択することとによって、最初の指紋を生成させる。その後、キーポイント除外モジュールは、第2の指紋を作成するために、第1、第2、および第4の非アンカキーポイントを除外した、キーポイントの次の組合せを生成する。このプロセスは、すべての除外されるキーポイント組合せに出会うまで継続される。

【0094】

指紋列モジュール1150から出力される指紋データ1160は、データベースから取り出される現在のイメージの一意イメージID1140と一緒に、指紋レコード1180を形成する。指紋レコード1180は、ファンツリーモジュール1170によってアドレス付けされた対応するファンツリー葉ノードの位置に格納される。ファンツリー葉ノード情報は、対応する指紋データ構造1190内の指紋レコード1180のリンクリストとして格納される。実際のファンツリー葉ノード(実際の指紋に対応する)だけが、投入される。特定の葉ノードに最初に到着する指紋が、最初にその葉ノードに投入される。複数の指紋が、同一の葉ノードにもう一度(すなわち、同一のファンツリー経路に従って)たまたま到着する場合には、第2の指紋情報は、第2指紋情報をその葉ノード上の第1指紋情報にリンクすることによって、同一の葉ノードで追加される。

【 0 0 9 5 】

本願の実施形態で、指紋の列は、クエリイメージがターゲットイメージに対して相対的に回転される時であっても対応する指紋の照合を容易にし、取り込みプロセス中の透視ひずみおよび他のイメージ品質劣化に起因して非アンカキーポイントが欠けることを許容するために、前に説明した S L C D 法に従って並び替えられる。

【 0 0 9 6 】

本願の文脈では、並び替えられた指紋の列が、オリジナルの指紋列の代わりに使用される。順列を使用することの利益は、照合する必要がある候補指紋列の数がかなり減ることである。オリジナルの指紋の列のすべての可能な循環する組合せを調べるのではなく、最小の最も数少ない数字の少数の順列だけが調査される。順列は、指紋データベース準備中に前もって生成され、格納される。クエリ時には、クエリ指紋の列の最初の順列だけが、照合に使用される（すべての順列が前もって格納され、正しい順列が自動的に見つけれられるので）。

10

【 0 0 9 7 】

本願の実施形態による順列の使用は、順列の数が、平均してオリジナルの指紋の列の長さよりはるかに小さいので、わずかなストレージ増加で大きい性能向上をもたらす。

【 0 0 9 8 】

本願の一実施形態では、ターゲットイメージ位置に対する相対的なクエリイメージの最初の観察されるキーポイントの回転方位を確立するために、テストが実行される。このテストは、可能な最初の観察されるクエリキーポイントの組合せのそれぞれを評価することと、最良の列の一致を選択することとによって実行される。

20

【 0 0 9 9 】

本願の第 2 の好ましい実施形態では、N 個の可能な最初に観察されるクエリキーポイントの組合せは、前もって事前計算され、照合すべき追加の指紋の候補の列として格納される。クエリ時に、N 個の事前計算された組合せのうちの 1 つと一致するすべてのクエリ指紋が、一致と考えられる。誤ったクエリ指紋が余分な組合せのうちの 1 つと実際に一致する尤度は低い。この第 2 の実施形態の主な利益は、追加の候補の最初のキーポイントの指紋の組合せを含むために多少より大きいストレージを犠牲にして、第 1 の実施形態に対して相対的により高い照合性能を可能にし、クエリ計算負荷を下げることにある。

【 0 1 0 0 】

30

第 3 の好ましい実施形態では、Smallest Least Common Quantization Digit (S L C D) 法が使用される。S L C D 法は、最小の最も数少ない量子化数字を判定するために指紋の数字の列のヒストグラムを実行することと、最小の最も数少ない数字の位置ごとに指紋の列を並び替えることとに基づく。S L C D 法の利益は、すべての可能な指紋の列の数字の順列の空間全体における網羅的検索を必要とせず、よいバランスのとれた計算負荷で指紋あたりの順列の数を最小にしようとすることである。

【 0 1 0 1 】

並び替えられた指紋は、高速で効率的な指紋照合を容易にするために、独自の形で編成される。本願の一実施形態では、並び替えられた指紋の情報は、クエリイメージとピクチャコレクション内の指紋の候補との間の高速指紋照合のために、ハッシング方式を使用して編成される。本願の第 2 の実施形態では、並び替えられた指紋情報は、ファンツリー法に従って編成される。一般的なハッシング技法とは異なって、ファンツリーは、高次元空間での効率的な近傍指紋検索（たとえば、1 つまたは少数の数字の変化を除く同一の列を有する目標に近いが不十分な指紋）を可能にし、少なくとも、ハッシュテーブル手法と同様に高速である。図 1 2 に、高速に指紋をインデクス付けする方法として、ファンツリー構造および指紋内容の対応する編成 1 2 0 0 を示す。

40

【 0 1 0 2 】

ファンツリー構造および指紋内容の対応する編成 1 2 0 0 は、格納されたデータを編成するのに使用される。ファンツリーは、3 つのタイプのノードすなわち、(a) ルートノ

50

ード 1 2 1 0、(b) 中間レベルツリーノード (たとえば、1 2 2 0, 1 2 2 1, 1 2 3 0, 1 2 3 1, および 1 2 3 2)、および(c) 枝の終りの終端葉ノード (たとえば、1 2 4 0, 1 2 4 1, 1 2 4 2, 1 2 4 3, および 1 2 4 4) からなる。ファンツリーは、ルートノード 1 2 1 0 から始まる。ルートおよび中間レベルノードのそれぞれは、パーシスタント面積比の量子化レベルごとに 1 つの、N 個 (この場合には $N = 8$) までの子を持つ。したがって、指紋は、ルートからファンツリーの葉ノードまでの、たどられる枝によってトレースされる経路である。

【0103】

たとえば、図 12 の経路 (1 2 1 0 - 1 2 2 0 - 1 2 3 0 - 1 2 4 0) は、ルートノード 1 2 1 0 から特定の葉ノード 1 2 4 0 につながる、1 つの指紋に対応する。指紋のこの特定の表現は (この単純な例で)、目指す葉ノード 1 2 4 0 に達するのに 3 つの分岐コード (すなわち、「各ノードの左経路をとる」すなわち、(0) ラベルを付けられた分岐を 3 回) を使用する。

【0104】

ファンツリーの潜在的サイズは大きく、この場合には $8 \times 8^3 5$ であるが、実際のファンツリーは、分岐がデータベース内に実際に現れる指紋についてのみ任意の所定のノードで構成されるので、疎に構成される。ルートの近くでは、多くのノードが、8 個すべての分岐を含むが、2 ~ 3 レベル下では少数の分岐だけが展開される可能性がある。

【0105】

葉ノードは、外に出る分岐を有しないという点で、他のツリーノードタイプと異なる。その代わりに、各有効な葉ノードは、クエリ時にクエリ指紋と照合される、図 12 の要素 (1 2 5 0, 1 2 5 1, 1 2 5 2, 1 2 5 3, 1 2 5 4, 1 2 5 5, 1 2 6 0, 1 2 6 1, 1 2 6 2, および 1 2 6 5) を含むがこれらに限定はされない指紋レコードのリスト 1 2 7 0 を含むリンクリストデータ構造を指し示している。これらは、指紋データ内容 1 2 8 0 をも構成する。

【0106】

指紋データは、データベース内の関連する指紋レコードへのすばやいアクセスを容易にし、それと同時に、現在の処理対象のクエリ指紋に関して関連しない指紋レコードを調査する必要を除去するように効率的に編成される。

【0107】

ファンツリー手法は、ニアミス (near-miss) 指紋を取り扱う 2 つの方法を使用可能にする。ニアミス指紋とは、少数の数字の変化を除いてほぼ同一の整数の列を有する指紋である。上述の 2 つの方法とは、次の 2 つである。(1) ニアミス指紋の組合せを前もって計算し、ファンツリーに事前に格納することができる。(2) ニアミス指紋の組合せは、クエリ時の間にクエリ指紋の列から計算され得、検索時にファンツリー経路をバックトラックすることができる。第 1 の方法は、ニアミス指紋の組合せを格納するためのより大きいメモリを犠牲にして、計算のほとんどをオフラインで実行するという利益を有する。その一方で、第 2 の方法は、実際のクエリ時に追加計算を必要とするが、ストレージ用の余分のメモリを必要としない。

【0108】

ファンツリーは、効率的なシステム性能およびイメージの非常に大きいコレクション (たとえば、数百万個のピクチャを含む) までスケールアップする能力を可能にする。ファンツリーは、大きいイメージコレクションのサイズに特に適切である。実際に、よい指紋設計を用いると、ファンツリーを使用する検索の時間は、イメージコレクションサイズに伴ってゆっくり増加するのみである。それでも、小さいものから適度なまでのイメージコレクションサイズに関して、より単純なハッシング方式を使用することができる。

【0109】

I. b. 1. ピクチャコーパスに関するファンツリーの作成

指紋データベース (たとえば、図 11 の 1 1 9 0) 内の指紋は、メモリサイズおよびローディング時間を減らすために圧縮されたコンパクトなフォーマットで格納される。この

10

20

30

40

50

セクションでは、ピクチャコレクションから抽出されたコンパクトな指紋データベースからファンツリーをどのように作成するかを説明する。ファンツリーは、所定のピクチャコーパスについて1回だけ作成する必要がある。ファンツリーが作成された後に、ファンツリーデータ構造（指紋を含む）を、メモリ内に保存し、その後、任意の個数のイメージクエリについて再利用することができる。

【0110】

所定の指紋データベース1305からファンツリーを作成するプロセス1300を、図13に示す。このプロセスは、次ページモジュール1310を使用して一回に1ページのイメージずつ指紋データベース内容を読み取ることによって開始される。ページごとに、次の情報がデータベースから取り出される：（a）一意イメージID（指紋データベース作成時に一意イメージIDモジュール（たとえば、図11の1140）によって割り当てられる）、（b）現在のページ上の識別される指紋の個数、および（c）現在のページの指紋レコードの内容。

10

【0111】

一意イメージIDは、特定のページのイメージならびにそれが属する文書を一意に識別する整数内部表現である。イメージID表現は、内部のインデックス付けに便利に使用される（文書の従来のUniversal Resource Locator（URL）ストリングを使用するよりコンパクトであり、より少ないスペースを占めるので）。実際のピクチャ内容は、指紋データベースには絶対に格納されず、指紋情報だけが格納される。

【0112】

20

次指紋モジュール1320は、データベースから、現在のページに関する現在の圧縮された指紋エントリを取り出す。指紋解凍モジュール1330は、指紋内容を解凍し、進行して、キーポイント除外モジュール1340を使用して複数の非アンカキーポイントを除外することによって指紋の列を生成する。結果の指紋は、使用される方法に応じて、ハッシング構造またはファンツリー構造に格納される。

【0113】

ファンツリーの場合に、新しい指紋は、一回に1つの指紋ずつ、順次追加される。キーポイント除外モジュール1340の出力の候補指紋ごとに、順列ソートモジュール1350が、候補指紋の組合せを処理して、smallest least common quantization digit（SLCD）アルゴリズムまたは他の方法に従って指紋の列の順列を生成する。

30

【0114】

順列ソートモジュール1350の出力は、現在の処理対象の指紋の並び替えられた指紋量子化比の列のリストである。このリストは、smallest least common digitアルゴリズムの特性に従って、少なくとも1つの順列エントリを含むことが保証される。時々、並び替えられた指紋の列のリスト内に複数の順列がある場合がある。並べ替えられた指紋の列の数は、SLCD法の下では変化する可能性がある。順列を使用しない本願の実施形態では、順列ソートモジュール1350がバイパスされ、キーポイント除外モジュール1340からの並び替えられない指紋の列が、次置換モジュール1360の入力に直接に渡されることに留意されたい。

40

【0115】

次順列モジュール1360は、順列ソートモジュール1350の出力から現在の指紋の順列の列を取り出す。並び替えられた指紋のそれぞれが、順次登録される。ファンツリー順列登録モジュール1370が、現在の順列をファンツリーに登録し、対応するデータレコードを既存の指紋データ内容に追加する。登録処理は、最初からファンツリーを作成する処理と同等であり、新しい指紋の列を記述するのに必要なノードおよび分岐のみを追加することにより行われる。指紋の経路全体がファンツリー内に既に存在する場合には、ツリー構造は変更されず、現在の指紋レコードだけが、最後の以前のエントリにそれをリンクすることによって既存の指紋データ内容に追加される。

【0116】

50

その一方で、新しい指紋の経路が、まだ全体としてファンツリー内に存在しない場合には、欠けているツリーノードおよび分岐が、その経路を完成するために必要に応じてファンツリーに追加される。さらに、現在の指紋レコードが、既存の指紋データ内容に追加され、新たに形成される葉ノードにリンクされる。最終結果は、ファンツリーが現在の指紋を含むように拡張され、その内容レコードが、一回に1つの指紋の経路ずつ、指紋データ内容に追加されることである。

【0117】

ファンツリーの代わりにハッシング方式が使用されるときには、指紋レコードは、当技術分野で周知の形で大きいハッシュテーブルにハッシュ化される。

【0118】

ファンツリーまたはハッシング方式のいずれかに次の指紋の順列を追加するプロセスは、最後の順列が最終順列モジュール1375によって検出されるまで、一回に1つの順列ずつ順次継続される。現在の順列が、まだ最後ではない間(1376)は、最終順列モジュール1375は、フローを次順列モジュール1360に戻ってルーティングして、次の順列を入手し、その内容のファンツリー/ハッシュテーブルへの登録に進む。このプロセスは、最後の順列が処理されるまで(1377)、継続される。

【0119】

現在の処理対象の指紋のすべての順列を登録した後に、この手順は、最終除外キーポイントモジュール1380に移る。これが最後の候補指紋の組合せではない場合(1381)には、このプロセスは、キーポイント除外モジュール1340に戻って、次のキーポイントを除外された次の候補指紋の組合せを処理する。これは、除外されるキーポイントのすべての可能な組合せがファンツリーに登録され1382、その内容が指紋データ1305に追加されるまで継続される。

【0120】

解凍、キーポイントの除外、順列のソート、結果として得られるすべての順列のファンツリーまたはハッシング方式への登録のプロセスは、最後の指紋が最終指紋モジュール1385によって検出されるまで、一回に1つの指紋の組合せずつ順次継続される。現在の指紋が、まだ最後の指紋ではない間(1386)、最終指紋モジュール1385は、フローを次指紋モジュール1320に戻して、次の指紋を入手し、進行して、解凍し、置換をソートし、それらのすべてをファンツリーまたはハッシュテーブルに登録する。このプロセスは、現在の処理対象のページの最後の指紋が処理される(1387)まで継続される。

【0121】

現在の処理対象のページのすべての指紋が、上で説明した形で処理され、そのすべての順列がファンツリーに成功して登録され、指紋データ内容に追加されたならば、ページURL追加モジュール1390が、現在のページのURL(Universal Resource Locator)を取り出し、内部の一意イメージIDから真の文書URLへの逆マッピングを別々のデータ構造テーブルに追加する。一意イメージID番号は、完全なURLの文字列よりコンパクトなので、内部的に使用される。

【0122】

最後に、指紋データベース1305から指紋を取り出し、解凍し、ソートし、すべての順列をファンツリーおよび指紋データレコードに登録するためのページの処理は、最後のページが最終ページモジュール1395によって検出されるまで、一回に1ページずつ順次継続される。現在のページが、まだ最後のページではない場合には、最終ページモジュール1395は、フローを次ページモジュール1310に戻して、次のページを入手し、進行して、その指紋の順列の内容をファンツリーおよび関連する指紋データ内容に登録(1370)する。このプロセスは、最後のページが初期クエリの使用のために準備される(1397)まで継続され、その後、このプロセスは終了する(1398)。

【0123】

ファンツリーの作成のプロセス全体を、最初のイメージクエリの前にオフラインで行う

10

20

30

40

50

ことができる。指紋データベースが、本願の実施形態に従って準備されたならば、それを、後続のイメージクエリについて再利用することができる。いくつかのターゲットイメージの追加または除去の後など、基準イメージデータが変更された場合にのみ、ファンツリーを再作成する必要がある。

【0124】

I. c. ランタイムクエリ

クエリ指紋を照合するプロセス1400を、図14に示す。照合されるクエリイメージ1410が、入力で提示され、処理されて、キーポイント識別ステップ/モジュール1420によって候補キーポイント位置を識別される。1420でのキーポイント識別プロセスは、クエリイメージについて期待されるより低い品質に関連する顕著な相違を伴って、前に説明したピクチャコーパスに関するキーポイントを見つけるのに使用されるプロセスに類似する。

10

【0125】

クエリイメージ1410の候補キーポイントのリスト1425を処理して、相対的なキーポイントの強さ、スケール、位置、および他のキーポイントへの近接に基づいて適切なアンカキーポイントの小さいサブセットを選択する。その強さに起因して、アンカキーポイントは、クエリイメージとターゲットイメージとの両方で高い確率で見つけられる可能性が高い。

【0126】

キーポイントは、指紋化モジュール1430によって小さい局所近傍グループ内で処理されて、照合のための候補クエリ指紋が生成される。クエリ指紋は、キーポイント識別モジュールの出力での候補キーポイントごとに計算される。各クエリ指紋は、(a)現在のキーポイント位置、(b)現在のキーポイント位置に最も近い、事前に決定された個数のアンカキーポイント、および(c)複数の最近近傍非アンカキーポイントの2D位置、から計算される。クエリイメージに関する指紋化法は、ピクチャコーパスに関する指紋化方法に類似する。ただし、クエリイメージについて期待されるより低い品質(後述のノイズガード法を導入する)および異なるクエリ指紋順列方式に起因する顕著な例外を伴う。

20

【0127】

本願の一実施形態では、入力イメージ内で識別される使用可能なキーポイントのすべてが使用される。もう1つの実施形態では、使用可能なキーポイントは、相対的な強さと、キーポイントのスケール、位置、および/または他のキーポイントへの近接などの他の考慮事項とによって、ならびに固定された個数の出力すべき最良のキーポイントだけを選択することによってソートされる。第3の実施形態では、入力イメージ区域1410を(オーバーラップするまたはオーバーラップしない)領域のセットに下位分割することができ、複数の好ましいキーポイントを、1つの領域が他のすべての領域に優位を占めるのを防ぐために、各領域から選択することができる。さらに本願の第4の実施形態では、指紋化モジュール1430は、指紋の列内の数字または数字パターン出現の頻度に基づいて、最もよい指標となる候補のクエリ指紋だけを選択することができる(まれで異常な数字パターンが、よりよい指標となるものの)。

30

【0128】

指紋化モジュール1430の出力は、局所キーポイントのグループごとの候補クエリ指紋のリスト1435である。

40

【0129】

候補クエリ指紋1435のリストは、特定の応用例および所望の性能レベル(処理時間および正確さに関する)に応じて、すべての識別可能な指紋のすべてまたはサブセットのみを含むことができる。たとえば、最もよい指標となる指紋の小さいサブセットだけを、相対的な指紋の強さに基づいて選択することができる。代替案では、処理時間が主な関心事ではないときに、すべての使用可能な候補クエリ指紋を、よりよい正確さのために使用することができる。したがって、指紋化モジュール1430の出力での候補指紋の個数を、正確さと速度との間のトレードオフとして調整することができる。

50

【0130】

指紋化モジュール1430のクエリ指紋を、その後、ファンツリーアドレッシングモジュール1440によってファンツリー経路に変換し、ファンツリーアドレッシングモジュール1440は、各クエリ指紋の対応する葉ノードを判定する。葉ノード内容を、指紋データ構造1490内で検索して、特定の葉ノードの一致する指紋レコード1470のリストを取り出す。

【0131】

ファンツリーではなくハッシング方式が使用されるときには、クエリ指紋をアドレッシングモジュール1440によってハッシュアドレスに変換する。ハッシュアドレスは、ハッシュテーブルの内容から候補指紋レコード1470の類似するリストを検索し、取り出すのに使用される。

10

【0132】

指紋照合モジュール1450は、候補クエリ指紋1435の情報を、指紋レコードの葉ノードリストに含まれる各指紋エントリ1470と照合する。候補クエリ指紋1435と特定の指紋エントリとの間に正確な一致がある時に限って、このシステムは、一致する指紋レコード1470から対応するイメージIDおよび指紋ID組合せを取り出し、進行して、カウンタ配列における一意の<イメージID/指紋ID>の組合せについて、モジュール1450の指定された指紋のカウンタを初期化し(初回のみ)、増分する。このシステムは、(別々のデータ構造内で)指紋照合フェーズ中に少なくとも1つの一致する指紋に出会ったイメージIDのリストをも維持する。指紋照合1450は、すべての候補クエリ指紋1435をこの形で処理し終えるまで継続される。結果として得られる一致カウンタ結果1455の配列の内容は、次に、最終スコア分析およびコレクション内の最もよく一致するピクチャまたはピクチャのセットの判定のために、指紋分析フェーズ1460に転送される。

20

【0133】

指紋分析フェーズ1460は、一致カウンタ配列の内容を分析して、最終的な一致結果を判定する。特定の分析1460は、応用例の特定の照合目的に依存する。複数の可能な照合目的があり得る。本願の一実施形態では、目的は、ピクチャコレクション内の最良イメージ一致を見つけることである。この場合、最も多い指紋一致(最高スコア)を集めたイメージIDすなわち指紋一致ID1465が、よい候補である。さらに、個々のカウントを、事前に指定された最小個数未満の指紋一致を有するページ一致を除外するために、信頼数として使用することができる。カウントを、より均一の相対的尺度を提供するために1ページ上の指紋の総数によって正規化することもできる(イメージ内容に応じて、いくつかのページが他のページより多数の指紋を有する場合があるので)。

30

【0134】

本願のもう1つの実施形態では、目的は、事前に指定された指紋カウントの最小値(または上記と同様の信頼レベルの最小値)を満足するコレクション(またはそのサブセット)内のすべての一致するイメージのリストを返すこととすることができる。

【0135】

さらに本願のもう1つの実施形態では、オリジナルイメージ区域をサブイメージのグリッドに下位分割する(オーバーラップを伴ってまたは伴わずに)ことができ、指紋照合分析を、よい指紋照合が得られる区域の荒い密度プロファイルを提供するために各サブイメージ内で別々に実行することができる。後者の分析は、オリジナルイメージの一部が欠けているか、シーン内の別の物体によって部分的にさえぎられる状況に特に適切である。この場合に、共通のイメージ区域(欠けておらず、またはさえぎられていない)は、それでも、高い度合の一致する指紋カウントを有する。さらに本願のもう1つの実施形態では、イメージの区分け(上記で概要を示したサブイメージへの分割による)を使用して、他のイメージ区域に対するあるイメージ区域の重みを強調することができる。たとえば、イメージの中央(焦点を合わされた物体がある場合がある)付近の指紋一致により大きい重みを与え、外側周辺の指紋一致により少ない重みを与えることである。この手法は、フレー

40

50

ム境界アーティファクト（イメージ外側境界に沿った）を除去するのに特に有用である。たとえば、電子オリジナルおよびその印刷された版の取り込まれたイメージ（たとえば、スキャナまたはセル電話機カメラからの）など、異なるソースからのイメージを照合する時である。後者の技法は、イメージ境界の外側の人工的背景の導入（たとえば、紙の白またはブルースクリーンカラーに対するイメージの角など）に起因する候補キーポイントの形成を避けるのに効果的に使用することができる。

【 0 1 3 6 】

指紋分析モジュール 1 4 6 0 の出力は、所望の照合目的に応じて、単一のまたは複数の一致するイメージ ID 1 4 6 5 の最終リストである。文書 URL ルックアップモジュール 1 4 8 0 では、このリストが、指紋分析モジュール 1 4 6 0 の一意イメージ ID の内部整数表現から真の文書 URL (Universal Resource Locator) に変換され、この文書 URL は、1 つまたは複数の一致するイメージの位置を識別する。この変換は、文書 URL 検索モジュール 1 4 8 0 内で達成され、一致結果 1 4 8 5 が作られる。一実施形態で、指紋データベース 1 4 9 0 の生成時に一意イメージ ID モジュールによって作成される逆マッピングテーブルが、一意イメージ ID をインデックスとして使用して URL を検索するのに使用される。最後に、照合された結果 1 4 8 5 が出力 1 4 9 5 に渡される。出力 1 4 9 5 は、電子ディスプレイなどのディスプレイ、ハードコピー出力を提供するプリンタ、メモリストレージとすることができる。あるいは、結果 1 4 8 5 を他の互換デバイスに送ることができる。

【 0 1 3 7 】

I . c . 1 候補クエリキーポイントの識別

クエリピクチャイメージ内で安定したキーポイントを識別する方法は、図 6 に関する議論を含む前述の議論で述べたピクチャコーパスに関する安定したキーポイントの識別に使用されるプロセスに類似する。この方法は、クエリイメージにフィルタの特定のセットを適用することと、スケールおよび空間の単一の関数 $D(x, y, s)$ を形成するためにフィルタ出力を一緒に組み合わせることとによって、複数のスケールでクエリイメージのスケール空間ピラミッド表現を構成することに基づく。安定したキーポイント位置は、関数 $D(x, y, s)$ が局所スケール空間近傍内でピーク値（最大値または最小値のいずれか）を達成するピクセル位置として各スケール内で識別される。

【 0 1 3 8 】

図 1 5 のクエリ候補キーポイント識別流れ図 1 5 0 0 に示された、図 6 を用いる説明からの 1 つの相違は、イメージエンハンスメントモジュール 1 5 1 0 が、クエリイメージの潜在的により低い品質に対処するために、適応しきい値モジュール 6 2 0 の前に導入されることである。ある種の応用例で、クエリイメージは、低品質なセル電話機カメラを用いて、ターゲットイメージの印刷されたハードコピーまたはスクリーン版から取り込まれる場合がある。この状況で、クエリイメージは、低解像度、ブラー、ノイズ、変化する照明、反射、透視ビューイングひずみなどの複数の要因に起因して、かなりのイメージ劣化を示す場合がある。したがって、クエリキーポイント識別の前に入力イメージ品質を改善するために必要なイメージエンハンスメント (image enhancement) を適用するために、第 1 のイメージエンハンスメントモジュールを追加することが望ましい場合がある。特定のエンハンスメントおよびイメージエンハンスメントモジュールに対する変更は、応用例、クエリイメージの期待される品質、予想されるイメージひずみ、ならびに所望の正確さおよび性能目標に依存する。イメージエンハンスメントモジュール 1 5 1 0 を、リアルタイムでクエリイメージを照合するシステムおよび方法を表す図 1 4 のキーポイント識別モジュール 1 4 2 0 の前に導入することもできることをも理解されたい。

【 0 1 3 9 】

本願の一実施形態では、イメージエンハンスモジュールは、クエリイメージのシーンにまたがる変化するカメラの照明を推定するために背景推定モジュールを含む。背景推定は、あるピクセル区域で最も明るいピクセルレベルの推定と、クエリのシーンにまたがってゆっくり変化する照明をモデル化するための低次 2 次元多項式曲線のあてはめと、を含む

ことができる。多項式モデルを反転することによって、イメージにまたがる変化する背景照明の影響を、中和し、キーポイント識別プロセスへの干渉および潜在的に明るいまたは暗いキーポイント特徴の適度な識別の失敗を防ぐことができる。

【0140】

本願の他の実施形態では、イメージエンハンスモジュールは、キーポイント抽出の前にクエリイメージの品質をさらに改善するために、コントラスト強調、シャープニング (sharpening, 鮮鋭化)、およびノイズ除去などの他の既知のイメージ処理技法を含むがこれらに限定はされない。

【0141】

I. c. 2 クエリ指紋の計算

このセクションでは、クエリ指紋を識別し、ファンツリー構造内の関連するピクチャコレクションの指紋との照合のためにそれらのクエリ指紋を準備するプロセスを説明する。

【0142】

クエリ指紋化プロセスを、図16に示す。クエリイメージの指紋化方法は、図10に関して詳細に述べた、ピクチャコーパスの指紋化方法に類似する。したがって、説明の便宜のために、図10に対応するモジュールには、前と同一の符号を付す。しかし、クエリイメージについて期待されるより低い品質 (たとえば、ノイズガードモジュール1630の導入) および異なるクエリ指紋化順列方式 (たとえば、第1順列モジュール1640) に関連する複数の顕著な相違がある。

【0143】

第1の相違は、指紋データベース内にクエリ指紋を保存する必要がないので、クエリ指紋が圧縮されない (したがって、図10の圧縮量子化比モジュール1070は使用されない) ことである。その代わりに、クエリ指紋は、単に、一回に1つのクエリ指紋ずつ、処理され (たとえば、量子化比追加モジュール1620、ノイズガードモジュール1630、第1順列モジュール1640)、照合される (たとえば、指紋照合モジュール1650)。各クエリ指紋は、照合すべき関連する候補指紋のリストを取り出すために、ファンツリーまたはハッシュテーブル内で検索される。所定のクエリ指紋の照合プロセスが完了したならば、そのクエリ指紋は、破棄され、照合結果 (一致カウンタ配列に関する) だけが、あるクエリ指紋から次のクエリ指紋へ累算され続ける。

【0144】

第2の相違は、クエリ指紋化が、順列のソートを必要としないことである。というのは、すべての順列が、ピクチャコレクションの準備フェーズ中に前もって生成され、クエリ時には1つの順列すなわち最初に現れる順列 (第1順列モジュール1640) だけが使用されるからである。すべての可能な順列が、前もって準備され、ファンツリーまたはハッシュテーブル内に登録されているので、すべての有効な順列が、対応する一致を有しなければならない。

【0145】

第3の相違は、以下で説明するノイズガードの概念 (たとえば、1630) を、ファンツリー準備フェーズ中に、またはその代わりにクエリ時に適用できることである。ノイズガード法のクエリ時の適用は、余分なストレージメモリを必要としないという点でより経済的であり、ファンツリーは、計算のわずかな増加で、ニアミス指紋を扱う便利な形を提供する。

【0146】

もう1つの重要な相違は、クエリ照合の時間を最小化する必要に関係する。多くの応用例で、クエリ照合は、リアルタイムで行われるが、システムは、ある厳しい性能要件を満足するかそれを越えることを期待される。そのような状況では、一般に、実際のクエリ照合時間を最小にするために、潜在的な指紋組合せのソートおよび可能なすべてのものを前もって準備することにより多くの時間を割り当てる、アンバランスな前倒し (front-loaded) システムを有することが望ましい。たとえば、ファンツリーまたはハッシュテーブルの作成のオフラインステージ中に、本願の実施形態の方法は、欠けているキ

10

20

30

40

50

ーポイントの組合せを含むすべての可能な指紋の順列を前もって登録する。これらの組合せのうちのいずれか1つでも、指紋の一致を得るのに十分である。最初の一致（たとえば、1655）が観察されたならば、特定の指紋について追加の一致を探し続ける必要はない。

【0147】

むしろ、このプロセスは、すべての一致するターゲット指紋のターゲット文書の一致カウンタ配列更新1660を増分し、このプロセスは、最後のキーポイントに出会ったかどうかを判定する（1090）ために進行する（1665）。その代わりに、指紋照合モジュール1650が、一致を見つけない（1656）ときには、このプロセスは、最終除外モジュール1670に進み、ここで、このプロセスが、最後のキーポイントを除外しなかったと判定された（1675のNO）ならば、このプロセスは、キーポイント除外モジュール1610に移動し、新しいキーポイントを除外し、そして、このプロセスはさらに進む。キーポイントが最後に除外されたキーポイントではないと判定される（1676のYES）ときには、このプロセスは、最終キーポイントモジュール1090に継続して、すべてのクエリキーポイントに出会ったかどうかを判定する。追加のまだ未処理のクエリキーポイントがある（1091のNO）場合には、フローは、次キーポイント1020に戻って、次のキーポイントを処理し、そうではない（1092のYES）場合には、このプロセスは、最後のクエリキーポイントで終了する。全体的なクエリ時間は、よりありそうな一致をソートし、ファンツリーまたはハッシュテーブル内に最初に配置することによって最小化される。

【0148】

もう1つの相違は、照合すべき第1クエリ順列1640が、この実施形態による全体的照合時間を最小にするために、特定の指紋および他の欠けているキーポイントの組合せについてさらなる順列の検索を停止させることである。

【0149】

I . c . 3 . ノイズガード法

図16で使用されて示されている、ノイズガードモジュール/方法1630は、既存の方法に共通する問題を克服するために指紋の計算中に任意選択で適用することができ、優れた指紋照合性能につながるができる。

【0150】

ノイズガード1630は、パーシスタント・レシオ値が量子化の境界の近くに該当する時に、キーポイントの位置の小さい変化（たとえば、ノイズおよび/または有限の丸め誤差に起因する）が異なる量子化整数値につながるのを防ぐように設計される。ノイズガード法は、パーシスタント・レシオ値から最も近い量子化しきい値までの距離を監視する。この距離が、事前に指定された公差レベル未満である時に、ノイズガードモジュールは、いずれかの場合に指紋の一致が得られるように、可能な候補として両方の量子化列を生成する。

【0151】

指紋

「14603616376756326322477662652206667」

を検討されたい。

【0152】

第5パーシスタント・レシオの値が、1.0のインターバル境界のすぐ下の0.9999であり、したがって、図8Bに示されているように「3」の整数値に量子化されると仮定する。これは、コレクション内のターゲットイメージに関する。

【0153】

劣化したクエリイメージ内で、このパーシスタント・レシオの計算におけるノイズまたはある小さい丸め誤差の存在が、その値をインターバル境界のすぐ上、たとえば1.0001（0.9999からの0.0002という非常に小さい変化）に押し上げることが十分にあり得る。この値は、今は1.0より大きいので、「3」ではなく「4」に量子化さ

れ、結果のクエリ指紋は、もはや「1 4 6 0 4 6 1 6 3 7 6 7 5 6 3 2 6 3 2 2 4 7 7 6 6 2 6 5 2 2 0 6 6 6 7」に一致しなくなる。第5の数字での分岐が、ここで、完全に異なるファンツリー経路につながる。

【0154】

ノイズガード法は、量子化前の入ってくるパーシスタント・レシオを監視して、それらが整数境界の近くであるかどうかを調べる。十分に近いものに限って、両方の指紋の列が、ファンツリーに入力される。すなわち、単一の指紋の列ではなく、2つの指紋すなわち、ターゲットイメージの第5番目の数字に「3」を有する指紋および「4」を有する指紋が入力される。

【0155】

クエリイメージの時に、クエリ指紋が計算され、「3」または「4」のどちらであるかは問題ではない。というのは、いずれもファンツリー内に存在し、正しい一致が必ず見つかるからである。

【0156】

ノイズガード法の威力は、整数境界に近いときに限って組合せを追加することである。そうではなく、すべての数字について盲目的に追加を試みる場合には、極端に大きい数 $2^{\wedge}35$ の組合せが、指紋ごとに生成されるはずである。

【0157】

従来のハッシング方式が、そのような数字の変化に敏感であることをも指摘しなければならない。これは、従来のハッシング方式において、入力列をランダムに「ハッシュ化」し、2つのほぼ同一の数字の列が通常はお互いからはるかに離れ、他方の位置を知ることによって一方の位置を判定する効果的な形がないからである。それが、ハッシュテーブルではなくファンツリー法を使用することの1つの利益である。

【0158】

ノイズガードモジュールを使用することの1つの利益は、ノイズガードモジュールが、量子化のノイズ感度を下げることによって指紋化の頑健性を改善することである。しかし、これは、量子化がしきい値の近くになるたびに候補指紋の列の数が2倍になるので、追加計算という犠牲を払って達成される。幸運なことに、三角形比がしきい値の近くになる事象は、非常にまれであり（クエリイメージのノイズレベルに依存する）、したがって、影響は、平均して非常に大きくはない。さらに、全体的な性能レベルを維持するために、ノイズガードが各指紋で適用される回数に対して最大限度をセットする（量子化しきい値に近い多数の三角形比を有する指紋が、非常に特徴的ではないので）。この実施態様では、ノイズガードモジュールは、5回の限度を有して使用される。

【0159】

ノイズガード法は、ピクチャコレクションの前処理時にオフラインで、またはクエリ時にオンラインでのいずれかで適用することができる。ストレージの展望からは、コレクション全体について前もって行うよりも、クエリ時に追加のノイズの組合せを生成する方が経済的である。しかし、クエリ時のノイズガードの適用は、追加計算に起因してリアルタイム性能をわずかに低下させる可能性がある。ストレージとリアルタイム性能との間のトレードオフは、期待されるノイズレベルおよび量子化誤差の尤度に依存して最適化することができる。

【0160】

I. c. 4. クエリ指紋一致のカウント

このセクションでは、指紋照合を高速で効率的にするように設計された、図16の指紋照合モジュール1650のさらなる詳細を説明する。

【0161】

クエリ指紋照合モジュール1650は、照合結果を記憶するために、次の3つのタイプのデータ配列構造をメモリ内に含み、利用する。

【0162】

(a) Seen Docs 配列は、現在のクエリ指紋の照合の間にこれまでに会った文

10

20

30

40

50

書IDのリストを含む。SeenDocsリストは、新しいクエリ指紋の組合せのそれぞれの開始に伴ってクリアされる。

【0163】

(b) SeenFinger配列は、現在のクエリ指紋の照合の間にこれまでに会った指紋IDのリスト(文書IDごとの)を含む。SeenFinger配列は、新しいクエリイメージごとにクリアされる。

【0164】

(c) MatchCount配列は、現在のクエリ指紋の照合の間にこれまでに会った文書IDごとの一致する指紋のカウントのリストを含む。MatchCount配列は、新しいクエリイメージごとにクリアされる。

10

【0165】

本願の実施形態で、連続するイメージインデックス番号が、便利に上記のアレイ(b)および(c)のピクチャイメージに割り当てられることに留意されたい。イメージインデックスから一意イメージID番号への変換は、別々のデータ構造内で維持される。

【0166】

図16の指紋照合モジュール1650の詳細なブロック図を、図17に図1700として示す。指紋照合は、図16の第1順列モジュール1640の出力からのクエリ指紋の組合せの列から始まる。クエリ指紋の組合せは、上で説明したsmallest leas
t common quantization digit(SLCD)に従って並べ替
えられる。クエリ列取得モジュール1710は、使用可能なセットから、照合される現在
のクエリ指紋の組合せの列を入手する。

20

【0167】

ファンツリーアドレスモジュール1715は、現在のクエリ指紋の列を使用して、ファンツリー構造内の指紋経路をたどる。ファンツリー経路に沿ったいずれかの点で、現在のツリーノードが行き止まり(すなわち、投入されていないツリーノード)につながる場合には、現在の指紋の照合は、打ち切られ、このプロセスは、最終クエリモジュール1785に進み(1716)、照合すべき次のクエリ指紋の列に移る。空のツリーノードの場合には、現在のクエリ指紋の列が、量子化されたクエリ指紋の列をもはや一致しなくさせる可能性があるひずみおよびノイズによって激しく劣化される時に発生し得る。少数の指紋の組合せ(所定のページについて数千個のうちの)が照合に失敗する時には、空ノードの場合を単純に無視することができる。しかし、多数のクエリ指紋が、空のツリーノードにつながる挙動を示し始める場合には、それが、クエリイメージがコレクション内に一致するイメージを有しないことのよい表示である可能性がある。

30

【0168】

通常の場合で、ファンツリーアドレスモジュール1715は、ルートノードから始まり、以前に説明した葉ノードで終わる、ファンツリーを通る現在のクエリ指紋の経路をたどる。葉ノードリスト取得モジュール1720が、対応する葉ノードリストのアドレスを取り出し、この葉ノードリストは、照合すべき関連する指紋レコードのリストを含む。

【0169】

代替案で、ファンツリーではなくハッシュテーブルが使用される場合には、ハッシュアドレスが、クエリ指紋から計算され、ハッシュテーブル内容の検索と、照合すべき関連する指紋レコードのリストの取出しとに使用される。

40

【0170】

リストレコード取得モジュール1725は、関連する指紋レコードのリストから現在の指紋レコードを取り出す。列比較モジュール1730は、クエリ指紋の列1726を、関連する指紋リストからのターゲット指紋の列(すなわち、コレクション指紋)1727と比較する。正確一致モジュール1735は、この2つの指紋の列の間の正確な一致を探す。正確な一致とは、すなわち、対応する量子化比の対のそれぞれが同一の値を有することである。単一の不一致は、1つの量子化比の位置であっても、不一致(1736)を宣言するのに十分である。クエリ指紋の列と現在のターゲットエントリとの間の正確な一致が

50

見つからない場合には、フローは、最終リストエントリモジュール 1780 に進んで、次の葉ノードリストエントリに移る。

【0171】

しかし、クエリ指紋の列とターゲット指紋の列との間に正確な一致が見つかる場合(1737)には、ページID/指紋ID取得モジュール1740が、現在の一意イメージIDおよび指紋IDを現在の葉ノードリストレコードから取り出す。既出ページIDモジュール1745は、一意イメージIDをSeenDocs配列内で検索して、現在のピクチャに現在のクエリ指紋の照合の間に既に出会ったかどうかを判定する。既に出会っている場合には、ページインデックス取得モジュール1750が、現在のピクチャの対応するページインデックスを、一意イメージID番号を使用してマッピングテーブルから取り出す。そうでない場合には、現在のページは、まだ出会ったことがない新しいピクチャである。新規ページインデックスモジュール1755が、新しいページインデックス番号を現在のピクチャに割り当て、今これに出会っていることを示すために一意イメージID番号を使用してSeenDoc配列に新しいエントリを追加する。それと同時に、新規ページインデックスモジュール1755は、MatchCount配列の現在のページインデックスアドレスに新しいエントリを追加し、その値を0(このページの一致する指紋なし)に初期化もする。

10

【0172】

次に、既出指紋IDモジュール1760が、現在の葉ノードリストレコードから得られた現在の指紋IDに現在のクエリ指紋の照合の間に既に出会っているかどうかを調べるためにチェックする。指紋IDをチェックする理由は、上述のように、各指紋が、それぞれが除外されるキーポイントの異なる組合せを用いてファンツリーまたはハッシュテーブルに複数回入力されるからである。しかし、組合せのうちのいくつかは、ノイズおよびイメージひずみの存在に起因して、正確な一致で終わらない場合がある。したがって、いくつかのイメージが他のイメージより高いカウントを複数の指紋の組合せから得る状況を防ぐために、各指紋の指紋IDを記憶する必要がある。本願の実施形態では、この問題に対する解決策は、指紋IDをチェックすることと、一意の指紋IDごとに1回だけMatchCount配列を増分することとによる。

20

【0173】

既出指紋IDモジュール1760によって、現在の指紋IDに既に出会っている場合には、MatchCount配列を増分する必要はない。指紋エントリ更新モジュール1765は、単に、必要に応じてページインデックスおよび指紋IDを用いてSeenFinger配列を更新する。その一方で、見られた指紋IDモジュール1760によって、現在の指紋IDにまだ出会ったことがない場合には、新規指紋エントリモジュール1770が、新しいSeenFingerエントリを割り振り、その内容が現在のページインデックスの現在の指紋ID値(すなわち、ページID/指紋ID取得モジュール1740での)になるようにセットアップする。さらに、ページカウント+1モジュール1775が、現在のページインデックスのMatchCount配列の内容を1つだけ増分して、この特定の指紋IDに現在のクエリ照合の間に出会った初めての時を示す。

30

【0174】

現在の指紋レコードの照合が完了した後に、最終リストエントリモジュール1780は、まだ照合されていないエントリが取り出された指紋レコードリストにまだあるかどうかを調べるためにチェックする。現在の指紋レコードが、まだ最後の葉ノードエントリではない場合には、リストエントリモジュール1780は、フローをレコード取得モジュール1725に戻して、葉ノードリストから次の指紋レコードを入手し、進行して、それをクエリ指紋の列と照合する。対ごとの照合プロセスは、最後の指紋レコードが処理されるまで継続される。

40

【0175】

現在の葉ノードリスト内のすべての指紋レコードが、すべて取り出され、指紋が、クエリ指紋の列と照合された後に、この手順は、最終クエリモジュール1785に移る。これ

50

が、最後の候補クエリ指紋の列ではない場合には、このプロセスは、クエリ列取得モジュール 1710 に戻ってルーティングされて、次の候補指紋の組合せを処理し、進行して、これを関連する指紋の列それ自体のリストと照合する。これは、クエリ指紋の列のすべての可能な組合せが、すべて、関連するファンツリーまたはハッシュテーブル内容と照合されるまで継続され、その後、このプロセスは終了する (1790)。

【0176】

指紋照合プロセスの最終結果は、Match Count 配列の内容であり、これは、クエリイメージと一致する少なくとも 1 つの指紋を有するコレクション内の関連するピクチャごとの一致する指紋のカウントを含む。Match Count 配列内のエントリの個数は、クエリ照合プロセスの間に出会った新しいピクチャの個数と同一であり、通常、コレクション内のピクチャの総数の小さい部分にすぎない。ピクチャコレクションの大多数を含むすべての他のピクチャは、特定のイメージクエリ中に一度も見られない（それらが、クエリイメージと共通する指紋を有しないので）。Match Count 配列内のイメージエントリが、そのランニングページインデックスによってアドレス付けされることに留意されたい。しかし、ページインデックスは、クエリ照合プロセス中に作成されるマッピングテーブルを検索することによって、簡単に一意イメージ ID 番号に変換することができる。

10

【0177】

最後に、Match Count 配列内の照合結果が、最終スコアリング分析およびコレクション内の最良の一致するイメージまたはイメージのセットの判定のために、指紋分析モジュール（たとえば、図 14 の 1460）に転送される。

20

【0178】

I . c . 5 . 最終判断

クエリ指紋照合プロセス中に、Match Count 配列は、クエリイメージと一致する少なくとも 1 つの指紋を有する関連ピクチャのそれぞれの一致する指紋の個数を累算する。通常、コレクション内の一つの特定のイメージすなわち一致するピクチャは、多数の一致する指紋（大きいカウント）を有し、多数の他のイメージは、特にコレクションサイズが大きい時に、少数の偶然の一致する指紋を有する可能性がある。

【0179】

上述のように、Match Count 配列の結果の内容は、最終スコア分析およびコレクション内の最もよく一致するピクチャまたはピクチャのセットの判定のために、指紋分析モジュールに転送される。複数の可能な照合目的があり得る。目的が、ピクチャコレクション内で最良のイメージ一致を見つけることである場合には、最も多い指紋一致（最高スコア）を集めたイメージ ID が返される。さらに、一致カウントを、事前に指定された最小個数未満の指紋一致を有するページ一致を除外するために、信頼数として使用することができる（ひずみおよびノイズに起因する偶然の少数の迷いカウントを除外するために）。カウントを、より均一の尺度を提供するために 1 ページ上の指紋の総数によって正規化することもできる（イメージ内容に応じて、いくつかのページが他のページより多数の指紋を有する場合があるので）。

30

【0180】

代替案では、目的は、事前に指定された一致する指紋カウントの最小値（または上記と同様の信頼レベルの最小値）を満足するコレクション（またはそのサブセット）内のすべての一致するイメージのリストを返すこととすることができる。

40

【0181】

本願の一実施形態では、オリジナルイメージの区域をサブイメージのグリッドに下位分割する（オーバーラップを伴ってまたは伴わずに）ことができ、指紋照合の分析を、よい指紋照合が得られる区域の荒い密度プロファイルを提供するために各サブイメージ内で別々に実行することができる。後者の分析は、オリジナルイメージの一部が欠けているか、シーン内の他の物体によって部分的にさえぎられる状況に特に適切である。この場合に、クエリとオリジナルとの間の共通のイメージ区域（欠けておらず、またはさえぎられてい

50

ない)は、それでも、高い度合の一致する指紋カウントを有する。

【0182】

さらに本願のもう1つの実施形態では、イメージの区分け(上で概要を示したサブイメージへの分割による)を使用して、他のイメージ区域に対するあるイメージ区域の重みを強調することができる。たとえば、イメージの中央(焦点を合わされた物体がある場合がある)付近の指紋一致により大きい重みを与え、外側周辺の指紋一致により少ない重みを与えることである。この手法は、フレーム境界アーティファクト(イメージ外側境界に沿った)を除去するのに特に有用である。たとえば、電子オリジナルおよびピクチャの取り込まれたイメージ(たとえば、スキャナまたはセル電話機カメラからの)または印刷された版など、異なるソースからのイメージを照合する時であり、ここで、取り込まれたイメー

10

【0183】

図14の指紋分析モジュール1460の出力は、1つまたは複数の一致するイメージIDの最終リストである。次に、文書URL検索モジュール1480は、このリストを、内部一意イメージID整数表現から真のピクチャURL(Universal Resource Locator)に変換し、このピクチャURLは、一致するピクチャまたはピクチャセットの位置および対応するページ番号を識別する。最後に、一致するピクチャまたはピクチャセットが出力され(1490)、これは、一致するピクチャまたはピクチャセットが、ディスプレイに表示され、プリンタによって印刷され、メモリに格納され、あるいはある他の互換デバイスに送られることを意味する。

20

【0184】

SIFTおよびそれから派生する手法とは異なって、本方法およびシステムは、並列実施に特に適する。というのは、本方法およびシステムが、本質的に、(a)クエリイメージ内のキーポイントの識別および指紋の列の計算と、その後の(b)効率的なファンツリー検索およびクエリイメージの(多数の)独立な指紋の照合(指紋照合中の)という2つの連続するステップからなるからである。新規のキーポイント抽出方法を使用すると、キーポイント識別フェーズ(a)は、各ピラミッドレベルの計算が、他のピラミッドレベルと完全に独立であり、フィルタにかかわらず正確に同一の時間を要するので、並列実施に適する。対照的に、一連のガウシアンフィルタリングおよびダウンサンプリングを利用する従来のキーポイント検出技法の反復的性質は、簡単に並列実施には向かない。フェーズ(b)に関して、各ファンツリー検索を、完全に並列で行うことができる。たとえば、大きいファンツリーを、複数のプロセッサまたは独立の機械の間で分割することができる。クエリ指紋を一回に1つずつ順次処理するのではなく、すべてのクエリ指紋が、まず、そのファンツリーまたはハッシュテーブルアドレスを判定する(たとえば図17のモジュール1715による、アドレス検索だけを使用して)ために処理される。次に、ソートフェーズが続いて、ファンツリー/ハッシュテーブルアドレスによってクエリ指紋のセット全体を並べ替え、所定のアドレス範囲の指紋の各グループを、ファンツリー/ハッシュテーブルのその特定の部分を含む対応するプロセッサまたは機械に送り出す。したがって、複数のファンツリー/ハッシュテーブルルックアップを、それぞれ異なる機械またはプロセッサ内で、完全に並列に行うことができ、照合結果を記録し、1つの指定された機械に転送し、最後に1つの指定された機械で組み合わせることができる。したがって、本方法は、その全体において十分に並列なマルチスケールピクチャ照合方式を提供するという利益を有する。

30

40

【符号の説明】

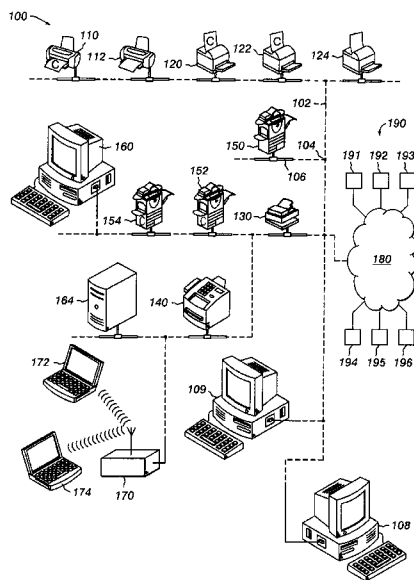
【0185】

100 コンピュータネットワーク、102, 106 ワイヤ、104 ワイヤジャン

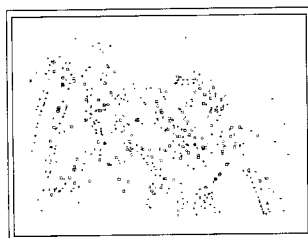
50

クシヨ、ン、108, 109 コンピュータ、110 カラープリンタ、112 カラー以外のプリンタ、120, 122 カラーレーザプリンタ、124 カラー以外のレーザプリンタ、130 スキャナ、140 ファクシミリ機、150 写真複写機、152 カラー写真複写機、154 組合せカラープリンタ/スキャナ/ファクシミリ機、160 独立型コンピュータ端末、164 独立型ハードドライブデータストレージ媒体、170 無線ネットワーク送受信器、172, 174 ラップトップコンピュータ、180 ネットワーク、190 周辺データ取り込みデバイス、191 デジタルスチルカメラ、192 デジタルビデオカメラ、193 セル電話機、194 スキャナ、195 携帯情報端末、196 文書インデクシングシステム。

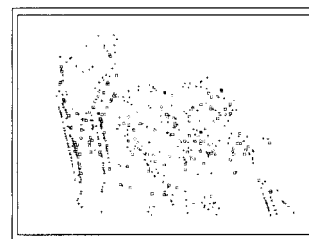
【図1】



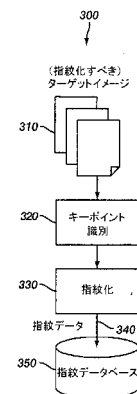
【図2B】



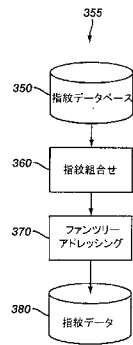
【図2C】



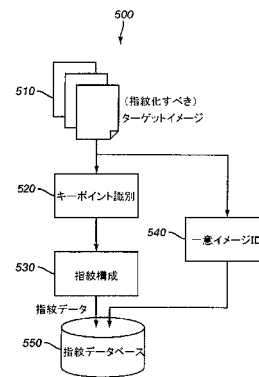
【図3A】



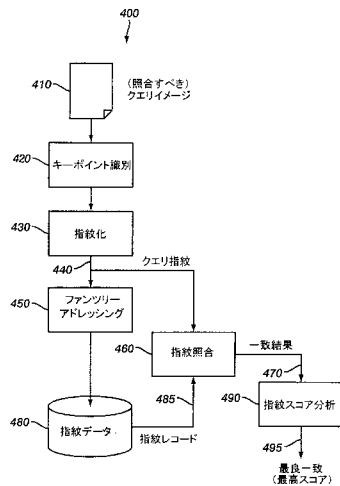
【図 3 B】



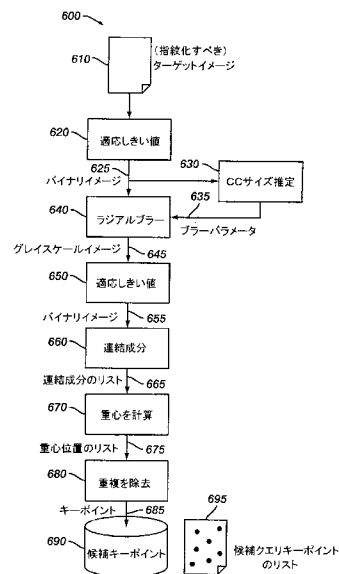
【図 5】



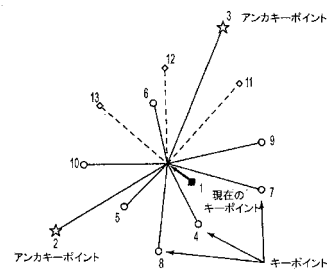
【図 4】



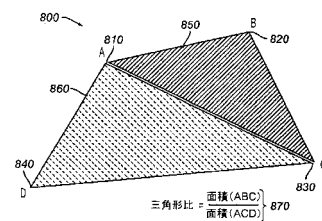
【図 6】



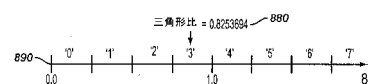
【図 7】



【図 8 A】



【図 8 B】



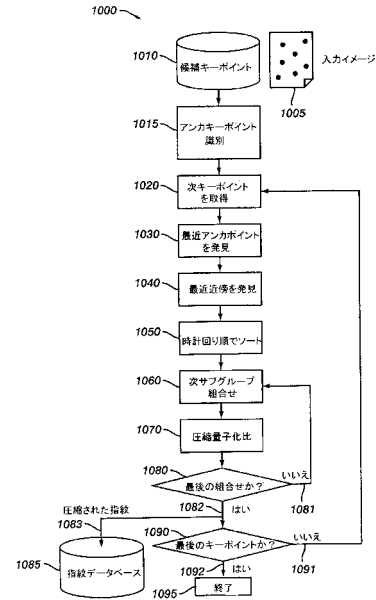
【図 9】

```

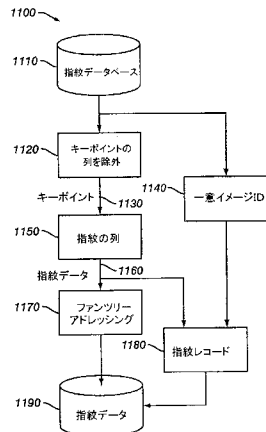
<Fingerprint Data Base>
<Page ID="83">
<Fingerprint Id="0">
<i0="77775663424652324425566166166767777">
<i1="677536333232244255661661667677777">
<i2="346136366736322442556632442567777">
<i3="446036366736332442556634242516677">
<i4="146046166775322225777342232516677">
<i5="14603616375332222777682652266677">
<i6="144034133667533324777564453456667">
<i7="144634636677363324777563462456116">
<i8="44665566654457736652515774667256643">
<i9="13540423202655407710515774667256643">
<i10="25851544312655407710445773665256643">
<i11="77771544310423207710556666665266723">
<i12="777756541154420650146656543515476">
<i13="77772551431340222547446366457655766">
<i14="77772551431340222547446366457655766">
<i15="77772551431350432657446556447651765">
...
<Fingerprint Id="1">
...

```

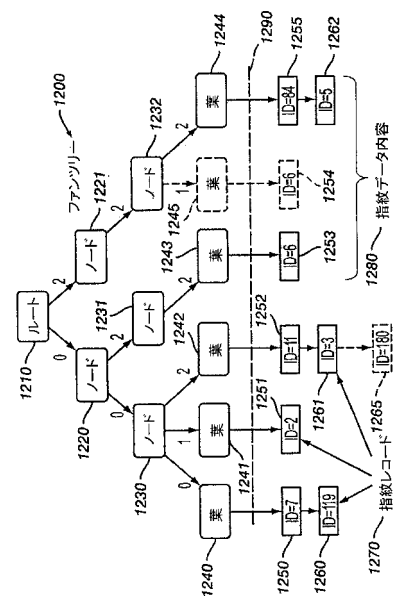
【図 10】



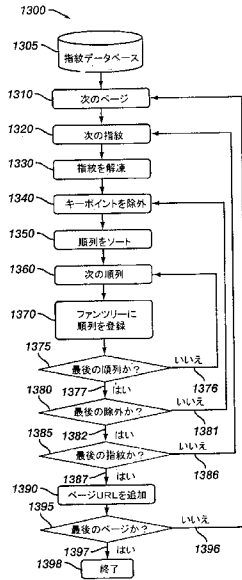
【図 11】



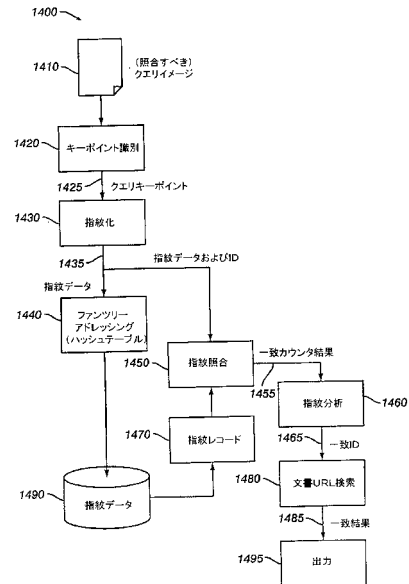
【図 12】



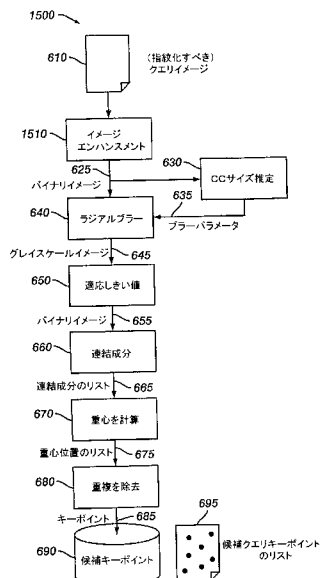
【図13】



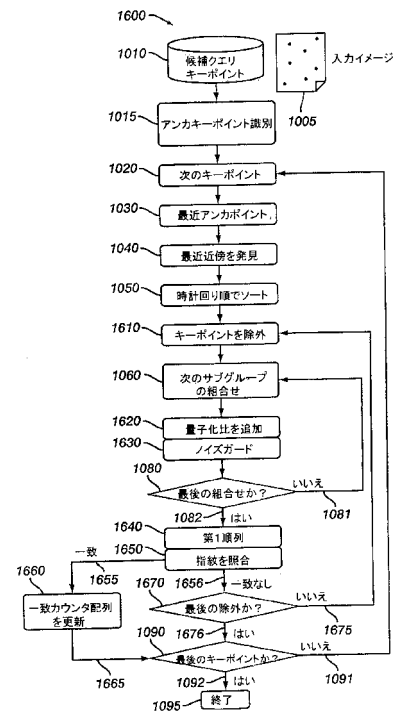
【図14】



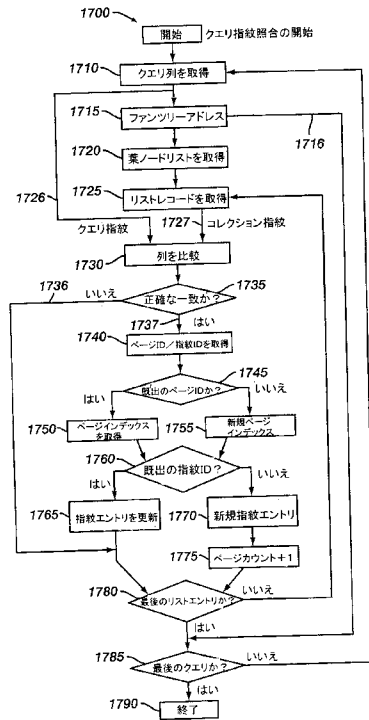
【図15】



【図16】



【図 17】



【図 2 A】



【 2 D 】



フロントページの続き

審査官 岡本 俊威

(56)参考文献 国際公開第2006/092957(WO, A1)

(58)調査した分野(Int.Cl., DB名)

G06T 1/00

G06F 17/30