(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3862414号
(P3862414)
(45)発行日　平成18年12月27日(2006.12.27)　(24)登録日　平成18年10月6日(2006.10.6)

(51)Int.Cl.　　F I
G06T 1/00 (2006.01)　　G06T 1/00 42OP
H04N 1/387 (2006.01)　　H04N 1/387

請求項の数 27　(全 51 頁)

(21)出願番号　特願平10-163736
(22)出願日　平成10年6月11日(1998.6.11)
(65)公開番号　特開平11-73494
(43)公開日　平成11年3月16日(1999.3.16)
審査請求日　平成14年5月1日(2002.5.1)
(31)優先権主張番号　97304101.5
(32)優先日　平成9年6月12日(1997.6.12)
(33)優先権主張国　欧州特許庁(EP)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT-PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　３０００
(74)代理人　100081721
弁理士　岡田　次生
(72)発明者　スティーブン・バーナード・ポラード
イギリス、ジーエル１１、５エスエル、グロスターシャー州、エヌアール・ダースレイ、アレイ、ザ・ストリート　５１

最終頁に続く

(54)【発明の名称】イメージ処理方法

(57)【特許請求の範囲】

【請求項１】

複数の別個に捕捉されたイメージ要素からイメージを復元する方法であって、各イメージ要素はイメージ・センサで捕捉されたイメージ・データと他のイメージ要素に対する該イメージ要素の位置を示す位置データで構成され、該方法は、

複数の別個に捕捉されたイメージ要素を受け取るステップと、

それぞれが、復元されるイメージの特定の空間領域を表す所定寸法の画素の格子からなる、複数のタイルの形で、イメージを復元するステップと、を含み、

複数のタイルの形で、イメージを復元するステップは、各々のイメージ要素に対して、当該イメージ要素の位置データから、当該イメージ要素の空間位置に対応するタイルの場所にタイルが存在するかどうか判断し、当該イメージ要素の空間位置に対応するタイルが存在するように、必要に応じて新しいタイルを生成し、イメージ・データおよび位置データに基づいて、当該イメージ要素の空間位置に対応するタイルの画素に、画素値を割り当てることによって、タイルに書込みを行うことを含む、方法。

【請求項２】

ハンド・スキャナにおいて使用される請求項１に記載の方法。

【請求項３】

位置データは、ナビゲーション手段によって導き出され、ナビゲーション手段は、イメージ・センサによってイメージ要素を捕捉する際に、当該イメージ要素についての位置データを供給する請求項１または２に記載の方法。

【請求項４】
位置データは、他のイメージ要素に対する各イメージ要素の相対位置をイメージ要素の特徴から決定する先行ステップによって導き出される請求項１または２に記載の方法。
【請求項５】
前記タイルは、隣接するが重なり合わない請求項１または２に記載の方法。
【請求項６】
前記タイルは、２次元において、平面を埋め尽くすように構成される請求項５に記載の方法。
【請求項７】
センサ読み取りストリームという形式で、イメージを捕捉し、それぞれのセンサ読み取りは、イメージ・データとこれに対応する位置データから成る請求項３に記載の方法。
【請求項８】
センサ読み取りは、センサ読み取りのグループで構成され、複数のタイルの形で、イメージを復元するステップは、連続的に生じる、センサ読み取りのグループに対して繰り返し実行される請求項７に記載の方法。
【請求項９】
復元されるイメージの特定の空間領域に対して生成される、それぞれのタイルをインデックス化するタイル・インデックスを作成するステップを含む請求項１または２に記載の方法。
【請求項１０】
前記タイル・インデックスは木として設けられ、この木の各ノードは、ノードの２倍の空間解像度を持つ４個の位置を指す請求項９に記載の方法。
【請求項１１】
複数のタイルの形で、イメージを復元するステップにおいて処理されるセンサ読み取りの空間位置に対応するタイルを格納するための有効タイル・バッファを備える請求項１または２に記載の方法。
【請求項１２】
複数のタイルの形で、イメージを復元するステップにおいて、センサ読み取りの空間位置に対応しないタイルの幾つか若しくは全てを圧縮することを含む、請求項１または２に記載の方法。
【請求項１３】
前記センサ読み取りは、センサ読み取りのグループで構成され、複数のタイルの形で、イメージを復元するステップは、連続的に生じる、センサ読み取りグループに対して繰り返し実行され、複数のタイルの形で、イメージを復元するステップでは、以前に処理されたグループの空間位置に対応するタイルを、現在のグループの空間位置に対応するタイルと比較し、新しく必要となったタイルを生成し、現在のグループには存在しない以前のグループに係るタイルを圧縮する請求項１または２に記載の方法。
【請求項１４】
圧縮されたタイルを圧縮タイル格納領域に書き込む請求項１または２に記載の方法。
【請求項１５】
タイル書き込みステップにおいて参照するために、有効タイル・インデックスを設ける請求項１１に記載の方法。
【請求項１６】
ナビゲーション手段は、捕捉イメージを担持した媒体の固有の構造的特性を検出するための１又はそれ以上のセンサを備える請求項３に記載の方法。
【請求項１７】
イメージ・センサが走査装置に備えられる請求項１または２に記載の方法。
【請求項１８】
捕捉イメージ上を非拘束に移動することができるセンサに対して適用される請求項１または２に記載の方法。

【請求項１９】
複数のイメージ要素を別個に捕捉するためのイメージ捕捉装置と、当該複数のイメージ要素からイメージを復元するように構成された単一のプロセッサとを含む走査システムであって、
イメージ捕捉装置はイメージ・センサを備え、各イメージ要素はイメージ・センサで捕捉されたイメージ・データと他のイメージ要素に対する該イメージ要素の位置を示す位置データで構成され、
単一のプロセッサは、
複数の別個に捕捉されたイメージ要素を受け取るステップと、
それぞれが、復元されるイメージの特定の空間領域を表す所定寸法の画素の格子からなる、複数のタイルの形で、イメージを復元するステップと、を実行するように構成され、
複数のタイルの形で、イメージを復元するステップは、各々のイメージ要素に対して、当該イメージ要素の位置データから、当該イメージ要素の空間位置に対応するタイルの場所にタイルが存在するかどうか判断し、当該イメージ要素の空間位置に対応するタイルが存在するように、必要に応じて新しいタイルを生成し、イメージ・データおよび位置データに基づいて、当該イメージ要素の空間位置に対応するタイルの画素に、画素値を割り当てることによって、タイルに書込みを行うことを含む、走査システム。
【請求項２０】
イメージ捕捉装置はナビゲーション手段を備え、ナビゲーション手段は、イメージ・センサによってイメージ要素を捕捉する際に、当該イメージ要素についての位置データを供給する請求項１９に記載の走査システム。
【請求項２１】
他のイメージ要素に対する各イメージ要素の相対位置をイメージ要素の特徴から決定する手段をさらに含む請求項１９に記載の走査システム。
【請求項２２】
イメージ捕捉装置はセンサ読み取りストリームを捕捉するように構成され、センサ読み取りの各々は、イメージ・データとこれに対応する位置データで構成される請求項２０に記載の走査システム。
【請求項２３】
ナビゲーション手段は、捕捉イメージを担持した媒体の固有の構造的特性を検出するための１又はそれ以上のセンサを備える請求項２０に記載の走査システム。
【請求項２４】
イメージ・センサは、捕捉イメージ上を非拘束に移動することができる請求項１９に記載の走査システム。
【請求項２５】
イメージ捕捉装置と単一のプロセッサとが両方とも走査装置に含まれる請求項１９に記載の走査システム。
【請求項２６】
イメージ捕捉装置は走査装置であり、単一のプロセッサはコンピュータシステムに含まれ、走査システムは、複数のイメージ要素を走査装置からコンピュータシステムに供給するための、走査装置とコンピュータシステムとの間の通信インタフェースをさらに含む請求項１９に記載の走査システム。
【請求項２７】
ハンド・スキャナである請求項１９から２６のいずれかに記載の走査システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、イメージ・データとして捕捉されたイメージを復元するためのイメージ処理方法及び装置に関する。とりわけ本発明は、スキャナ、特に拘束されていない表面を走査するハンド・スキャナによって捕捉されたイメージに適用することができる。

【０００２】
【従来の技術】
オリジナル（原画）のイメージを電子的に形成するためのスキャナは周知である。スキャナによって捕捉されたイメージは、一般には、デジタル形式でメモリに格納された画素データ・アレイである。歪みのないイメージを得るためには、原イメージを画素データ・アレイに忠実にマッピングする必要がある。マッピングの忠実性を最大限に担保するために、スキャナは、通常、イメージ捕捉処理の間、オリジナルを機械的に拘束するようになっている。当業界において周知のスキャナのタイプとして、ドラム・スキャナ、フラットベッド・スキャナ、２次元アレイ・スキャナ、用紙送りスキャナ、及びハンド・スキャナが挙げられる。本発明は、とりわけハンド・スキャナに適用することができる。
【０００３】
従来のハンド・スキャナは、ユーザが、光電センサ素子からなる線形アレイをオリジナル上で移動する必要がある。移動は一般には手作業であるが、幾つかの実施態様ではモータ駆動による。アレイ位置情報は、コンピュータの「マウス」の操作で採用されているものと同様の手法によって測定される。線形センサが移動するに従って、オリジナルと接触しているホィール、ボール若しくはローラの回転が検出され、機械要素の回転によって位置情報が測定される訳である。一般に、オリジナルと接触している機械要素の表面は、例えばゴムのような高い摩擦係数を有しており、スリップやスキッドに抗するようになっている。走査処理の間の自由度を並進だけに制限するために、円筒ローラ若しくは堅い軸で連結された２個のホィールを用いてもよい。オリジナルに対する走査方向を固定したり、一対のホィールやローラで課される並進方向の拘束を実現するために、真直な先端や他の取り付け具がしばしば使用される。それにも拘らず、位置エンコーダはスリップやスキッドの影響を受け易く、画素データ・アレイはオリジナル上のイメージとの対応を失いがちである。
【０００４】
ハンド・スキャナは、一般には、イメージ・データを保管したり処理したり利用したりするための別個のコンピュータに直接接続される。スキャナは、一般に、緑色若しくは赤色の発光ダイオードによってユーザにフィードバックして、所望のイメージ解像度に適った速度を維持するようになっている。ハンド・スキャナの中には、走査速度の増大に伴って機械的抵抗が増すような電磁ブレーキを用いて、ユーザがスキャナを速い速度でイメージ上を引き摺るのを防止するものもある。
【０００５】
ハンド・スキャナは、比較的小さな撮像アレイを用いているので、通常、１回の通過ではＡ６より大きな文書を処理することはできない。より大きな文書を構成する複数の走査帯を結合するためには、縫合アルゴリズムを必要とする。走査帯の縫合は、スキャナとは別個のコンピュータを用いた別個の処理としても実現できる。複数のページからなるビジネス文書やレポートをハンド・スキャナで走査することは、退屈な作業であり、しばしば低品位の結果を招来する。スキャナの業界では、イメージの帯を縫合するための幾つかの技術が知られている。これらの技術は、一般には、一対の完全なイメージの帯を必要とし、２つの帯を位置決めするような単一の全体的な変形を生成するものである。国際特許出願公開ＷＯ９６／２７２５７号には、さらに進歩した技術が教示されている。
【０００６】
既に述べたように、ハンド・スキャナには通常幾つかのタイプの固定具が用いられる。固定されない場合には、ハンド・スキャナがオリジナルの上を移動するに従って、幾分の回転が加わる傾向がある。もし、スキャナ移動中にユーザの肘が平坦な表面の上に置かれているならば、スキャナとユーザの肘の間の距離で定まる半径の回転が生ずるであろう。この結果、走査によって得られた電子イメージは歪んでしまうであろう。１つの帯を走査する間に起こる他の曲線的な動きによっても、歪みは生ずるであろう。
【０００７】
走査された電子イメージを形成するためには、イメージ・データとともに得られるナビゲ

ーション情報を取得することが肝要である。米国特許第５５７８８１３号には、スキャナの使い勝手を損なうことなしにナビゲーション情報を得ることができるハンド・スキャナについて記述されている。ナビゲーション情報は、走査されるオリジナルに固有の構造的な特性を検知する少なくとも１つのナビゲーション・センサによって得られる。イメージ・センサのオリジナルに対する移動は、イメージ・センサがオリジナルに対して移動する間の固有の構造的な特性の変動を監視することによって追跡される。監視される固有の構造的な特性とは、紙の繊維やオリジナルのその他の組成のような固有の構造的な特徴のことである。ナビゲーションは、あるいは小斑点に基づくものでもよく、この場合には、ナビゲーション情報を得るためには、イメージ・センサのオリジナルに対する移動はコヒーレントな照明を用いて生成される小斑点のパターンの変動を監視することによって追跡される。走査イメージ表面上におけるスキャナの移動は、本質的には任意なものであり、何にも拘束されない。

【０００８】

「固有の構造的な特性」とは、イメージ・データを形成することやオリジナル上の計画的な位置決めデータとは独立した要因に帰するようなオリジナルの特性である。ナビゲーション情報は、小斑点情報の位置信号、若しくは個々の固有の構造的な特徴の追跡を許すような位置信号のような、固有の構造的な特性の検知に応答した位置信号を生成することによって形成されてもよい。「固有の構造的な特徴」とは、オリジナルの特徴のことであり、原イメージを形成する処理の特性であり、イメージ・データを形成したり若しくはオリジナル上の構造的な位置決めデータとは独立したものである。例えば、もし原イメージを記録した媒体が紙製品であれば、ここで興味のある固有の構造的な特徴は、紙の繊維である。他の例では、光沢のあるオリジナル又は透明なオーバーヘッド・フィルムの上を走査するときに、イメージ・センサのナビゲーションは、反射領域に影響を及ぼす表面構造の変動を追跡することによって定まる。一般に、固有の構造的な特徴は、例えば１０～４０μｍ程度の微視的な、表面組織上の特徴である。

【０００９】

本発明は、国際特許出願公開ＷＯ９６／２７２５７号や米国特許第５５７８８１３号に教示されているものと同様に、ハンド・スキャナに直接的に適用することができる。これら２つは、本特許出願に組み込まれる。

【００１０】

上記の先行技術文献の中で言及されている多くの実施例において入力の非拘束性によって処理上の問題が生じる。イメージを復元するための画素アレイが捕捉された全てのデータを表示できるようにするためには、画素アレイを比較的大きくする必要がある。イメージ上の任意の点からデータ捕捉を開始することができるので、従来の構造では、捕捉すべきイメージよりも数倍大きな画素格子を用いることが必要であった。このような手法では、画素アレイ・データのためにかなりの記憶容量のメモリを割り当てる必要があり、したがって費用の顕著な増大を招来する。

【００１１】

【発明が解決しようとする課題】

本発明は、上述のような非拘束的な走査によって捕捉されたイメージを記憶するために必要なメモリ容量を低減する方法を提供することを目的する。とりわけ、本発明は、コンピュータ上のコスト削減と、任意的に取得された捕捉イメージから復元イメージを形成する速度の増大をもたらすものである。この方法は、特に、フリーハンド走査やフリーハンド・スキャナに好適である。

【００１２】

【課題を解決するための手段】

本発明によれば、複数のイメージ要素からイメージを復元するための方法が提供される。各イメージ要素は、イメージ・センサによって捕捉されたイメージ・データと、他のイメージ要素に対するこのイメージ要素の位置を示す位置データとからなる。この方法は、複数のタイルという形態のイメージを復元することを含み、各タイルはイメージの特定の空

間領域を表す所定次元の画素格子を含む。

【００１３】

位置データは、イメージ・センサによって捕捉されたイメージ上の１つのイメージ要素についての位置データを提供するナビゲーション手段によって導き出されてもよい。これは、とりわけデータ捕捉のために線形センサを用いる場合に優れている。本発明の他の側面では、位置データは、イメージ要素の特徴から各イメージ要素の他のイメージ要素に対する相対的位置を決定するという先行ステップによって導き出される。これは、とりわけ、エリア・センサを用いてデータを捕捉する場合に優れており、重なり合った領域の評価によって捕捉データの相対位置を査定することも可能となった。

【００１４】

タイルは隣接するが、重なり合うことはなく、２次元において、平面を埋め尽くすように構成されることが好ましい。この方法は、とりわけ、センサの読み取りストリームの形態でイメージを捕捉し、各センサの読み取りがイメージ・データとこれに対応する位置データからなるような場合に好適である。このようなデータ捕捉は、線形センサを用いてイメージの表面上をフリーハンド走査する場合に起こる。

【００１５】

この方法は、特に、イメージ上を任意な経路で走査することによって得られた非拘束的な入力に対して好適である。何故ならば、タイルは必要とされるいかなる位置でも作成されるが、実際に必要とされるまでは作成する必要はないからである。ある１つの好適な実施例では、この方法は、もし入力センサの読み取りのグループのイメージ上の空間位置に対応してタイルが存在するならば該位置データから決定されるタイルを管理するステップと、必要に応じて１又はそれ以上の新しいタイルを生成して、グループの空間位置に対応するタイルが存在するようにするステップと、イメージ・データと位置データに基づいたグループを入力センサが読み取ったの空間位置に対応するタイルの画素に画素値を割り当てることによってタイルに書き込むステップの各ステップで構成され、これらのステップは、センサ読み取りの連続的なグループに対して繰り返し行われる。

【００１６】

好ましい実施例では、この方法は、イメージの特定の空間領域について生成された各タイルをインデックスするためのタイル・インデックスを生成することを含んでいる。タイル・インデックスは、木の各ノードがノードの複数の空間解像度における複数の位置を指す木構造の形式で提供されるのが好ましい。

【００１７】

本発明のさらに好適な側面では、既に書き込まれたタイルは、イメージ捕捉が行われている間に圧縮することができる。タイル管理ステップでは、以前に処理されたグループの空間位置に対応するタイルが現在のグループの空間位置に対応するタイルと比較され、新たに必要となるタイルが生成され、現在のグループには存在しない以前のグループのタイルが圧縮されるのが好ましい。圧縮されたタイルは、圧縮タイル記憶領域に書き込まれ、現在のグループのタイルを保持するための有効タイル・バッファから消去される。

【００１８】

本発明は、さらに、上述の方法を実現するための走査装置をも含む。本発明がハンド・スキャナという形態で実装され、且つ、全ての処理を行うに充分なコンピュータ能力を有する場合、この方法は、ナビゲーション装置に修正をフィードバックするためのエラー概算を用いる。このようなエラー概算の例示的な形態の１つとその適用例は、本発明の実施例に関連して以下に詳解されている。このように、リアル・タイムでエラーが同定され、且つ、イメージを復元する際のエラーを修正するのと同様にナビゲーション装置自身の修正に利用されている。あるいは、本発明は、イメージ・データを収集するための走査装置という形態で実装されてもよい。この場合の走査装置は、本発明に従った方法を遂行するためのコンピュータ・システムと相互接続できるように設計されている。

【００１９】

【発明の実施の形態】

以下、本発明に従った走査装置について説明する。この装置は、本発明に従った方法に必要な処理を遂行するためのコンピュータ能力を有していてもよい。あるいは、この装置は、本発明に従った方法を遂行するために、コンピュータ・システムと相互接続できるようにデザインされていてもよい。

【００２０】

以下で説明する実施例では、捕捉したイメージの復元は、イメージ・データとともに得られたナビゲーション情報を用いて、ナビゲーション情報とイメージ情報に基づいてイメージ・データを修正することを含んでいる。修正とは、原イメージと出力イメージとの一致を達成するために、ナビゲーション情報を用いて、得られたイメージ・データを配列する処理のことである。好ましい実施例では、ナビゲーション情報は、走査されるオリジナルに固有の構造的な特性を検知する少なくとも１つのナビゲーション・センサによって得られる。オリジナルに対するイメージ・センサの移動は、イメージ・センサがオリジナルに対して移動する間に、固有の構造的な特性の変動を監視することによって追跡される。ここで監視される固有の構造的な特性とは、紙の繊維や、あるいはオリジナルのその他の組成のような固有の構造的な特徴のことである。ナビゲーションは小斑点でもよく、オリジナルに対するイメージ・センサの移動は、ナビゲーション情報を得るためのコヒーレントな照明を用いて生成された小斑点のパターンの変動を監視することによって追跡できる。

【００２１】

このように、ナビゲーション情報を得るために熟慮された態様は、多岐にわたる。最も広範な態様では、走査軌跡に対する走査装置の曲線的若しくは回転状の移動といった歪みを取り除くために用いられるナビゲーション情報の源には何ら限定がない。ナビゲーション信号は、オリジナル上のイメージ・データの検知（例えばテキスト文字の端部の同定）に応答した位置情報という形態であってもよい。この場合、位置情報は、イメージ信号の操作に用いられる。第２の態様では、例えば小斑点パターンの特性のように、位置信号は固有の構造的な特性の検知に応答している。第３の態様は、個々の固有の構造的な特徴（例えば紙の繊維）の位置を四六時中監視することによって走査装置のナビゲーションを追跡するというものである。

【００２２】

以下で説明される実施例では、イメージ・センサは光電素子からなる線形アレイであるが、ナビゲーションにはナビゲーション・センサ素子からなる少なくとも１つの２次元アレイを使用する。イメージ・センサの各端部に夫々２次元ナビゲーション・アレイを設置することによって、スキャナは３つの移動自由度を有することになる。もしオリジナルが平面的であれば、２自由度は並進で且つオリジナルのなす平面上で互いに直交しており、第３の自由度はオリジナルの平面の法線回りの回転である。２個のナビゲーション・アレイを用いることで、回転を追跡する精度は向上する。但し、各アレイは、単一のナビゲーション・アレイのみを用いたならば必要であったよりも小さなアレイでよい。ここで説明される実施例は、ナビゲーション・センサは２次元アレイを指すが、線形アレイであってもよい。オリジナルの固有の構造的ないかなる特性とも独立した方式によって他の位置追跡手段を走査装置に設置することによって、イメージ・データを修正するためのナビゲーション情報を安定的に得ることができる。

【００２３】

ナビゲーション・センサは、イメージ・センサに対して知られた位置に置かれる。好ましくは、ナビゲーション・センサは、可能な限りイメージ・センサの端点に近接している。これによって、ナビゲーション・センサはイメージ・アレイが移動する間に、オリジナルの端縁を越えてしまう可能性が少なくなる。イメージ・センサは、対象となっているイメージを表した信号を形成する。同時に、各ナビゲーション・センサは、オリジナルの固有の構造的な特性を表した信号を形成する。走査装置は、オリジナルとは接触を保ったまま、オリジナルに対して下降しながら、左から右へ、右から左へと移動を修正するように、フリーハンドで曲がりくねったパターンで移動してもよい。端から端に至る１つの走査帯は、前回の走査帯とは一部が重なる合うようにする。この結果、走査処理の間若しくは走

査処理に続いて、イメージは位置に関連して操作され、そして縫合される。イメージ信号の操作とは、イメージ・データの修正のことであり、この修正はナビゲーション・センサとこのナビゲーション・センサによって検知された固有の構造的特性に基づいて行われる。縫合とは、連続的な走査帯の間に得られたイメージ・データを接触させるのに用いられる処理のことである。縫合した後は、各走査帯のデータ間の位置的な関係は、オリジナルのイメージ中のイメージ・データ同士の位置的関係に従っている。

【００２４】

本発明に採用されたセンサのデザインやイメージ処理方法に関するさらなる側面は、国際特許出願公開ＷＯ９６／２７２５７号や米国特許５５７８８１３号で議論されている。

【００２５】

図１には、携帯型で、手持ち型の走査装置１０を示している。同図中では、オリジナル１４上で曲がりくねった経路１２を伴っている。オリジナルは、１枚の紙、オーバーヘッド用の透明フィルム、又はそれら以外のイメージを持った表面でよい。オリジナルの固有の構造的な特性は、曲がりくねった経路に沿ったナビゲーションを行う間の位置情報を得るために用いられてもよい。同図に示す例では、固有の構造的な特徴の位置が追跡され、その位置情報はイメージ・データを修正するのに使われる。しかしながら、以下では、他の実施態様について説明する。走査装置は、好ましくは自己内蔵バッテリで駆動するが、外部電源やコンピュータ又はネットワーク用のデータ・ポートとの接続部を含んでいてもよい。

【００２６】

図１に示す走査装置１０は、イメージ・ディスプレイ１６を含み、捕捉したイメージを眺めることができる。しかしながら、走査装置を使用する上で、ディスプレイは必須ではない。

【００２７】

図１乃至図３を参照すると、走査装置１０の前面１８には、オリジナル１４とイメージ・センサ２２との間の適切な接触を維持するための枢軸部材２０が配設されている。イメージ・センサ２２は、イメージ・センサ素子の線形的なアレイで構成される。ナビゲーション・センサ２４と２６は、イメージ・センサとは反対の端部に配設されている。ナビゲーション・センサ２４，２６は、枢軸部材２０の上に搭載されているので、ナビゲーション・センサはイメージ・センサに対して固定位置に置かれる。

【００２８】

走査装置１０には３自由度があり、その４ちの２つは並進自由度であり１つは回転自由度である。第１の自由度は、オリジナル１４の端から端までの移動（すなわちＸ軸の移動）である。第２の自由度は、オリジナルに対する上下方向の移動（すなわちＹ軸の移動）である。第３の自由度は、オリジナル１４の端縁に対するイメージ・センサ２２の回転状の不整合を伴った該装置の操作能力である。すなわち、イメージ・センサ２２は、該装置の並進方向に対して直角ではない角度を含んでいるかもしれない。

【００２９】

物理的に小型にしたいという理由のため、イメージ・センサ２２は接触型イメージ装置であることが好ましい。しかしながら、小型性が考慮の対象外であったり比較的小さなイメージが求められているような適用例では、投影光学系を採用したセンサであってもよい。このような適用例では、イメージ・センサ２２を構成する素子は、小さく、且つ、互いにより近接して束ねられねばならない。接触型イメージ装置は、通常、ＳＥＬＦＯＣという商標の下で市販されるレンズを採用する。ＳＥＬＦＯＣは、日本シート・ガラス(株)の商標である。従来例としては稀であるが、接触型のイメージ捕捉は、光源と近接センサとからなるインターリーブ型アレイ素子を使用して行われる。これはいかなるレンズも用いない。走査に適用するためには、従来のイメージ・センサを用いてもよい。イメージ・センサは、照明光源、照明光学系、及びイメージ変換光学系を含んだユニットの一部であってもよい。

【００３０】

イメージ・センサ２２は、感光性の分離型素子の線形アレイとして描かれている。各素子の間隙は、スキャナ１０の空間解像度を決定する役割がある。例えば、１０１．６ｍｍ長の線形アレイであれば、３００ｄｐｉの解像度を達成するためには、１２００個のセンサ素子が必要となる。センサは、電荷結合素子、アモルファス・シリコン・フォトダイオード・アレイ、又は、当業界で周知のその他のタイプの線形アレイ・センサであってもよい。

【００３１】

イメージ・センサを設計する上で重要な考察項目は速度である。イメージ・センサ２２は、１秒当たりおよそ５Ｋサンプルの速度で各画素を捕捉できることが好ましい。線形イメージ・アレイは、一般に、シリアル・データ・ストリームを生成し、画素値すなわち電荷はシフト・レジスタに入れられ、次いでシフト・アウトする。要求された速度を達成するためには、画素値が僅かなセルを介してシフトできるように、イメージ・アレイ全体から出力されるシリアル転送速度が非常に速いか、又は、複数のタップが必要である。このため、並列処理が導入されているが、デジタル処理にとって有利である。

【００３２】

速度の要件のもう１つの帰結は、オリジナルの表面における画素面積と、集光され各アレイ素子に伝送される照射光の立体角との積が、２００マイクロ秒のオーダーの積分時に検出可能な信号を生成する程充分大きあるべきであるということである。向上オプションとして、ある光学要素をセンサに付加することによって、各センサ要素が応答するセンサ・ピッチの有効部分を増すことができる。通常、アレイ・マトリックスの中には未使用の領域があるので、このような集光光学系の感度は増す。

【００３３】

イメージ・センサ２２を簡単に改良することによって、カラー・イメージを検出することができる。３個の互いに平行な線形アレイによれば、カラー・イメージの捕捉ができる。ここで、各々の線形アレイは少なくとも１つの組み込み型フィルタ素子を持ち、各々は入射光中の赤、緑、青の各成分を別個に透過するものとする。あるいは、広帯域の感度を持った単一のアレイが赤、緑、青の各光源で順に照射されるようにしてもよい。

【００３４】

イメージ・センサ２２の動作を向上するための照明については、琥珀色の波長で高い光強度を持つ発光ダイオードからなる線形アレイを使用してもよい。しかしながら、好ましい光源や光学要素を選択するためには、原イメージを担持する媒体にもに依存する。光の波長は、不要な信号は無視して、オリジナル１４上の所定の領域を走査する間に得られたイメージ・データの比を最大にするようなものが選択される。照明光学系は、ＬＥＤドーム型レンズからなるか、若しくは、精密成形された光学要素からなる光パイプを含んでいてもよい。光パイプによれば、光のロスを最小限にして照明光をオリジナルの上に伝送することができる。このような設計によれば、オリジナルの目標領域を、広い角度の範囲で比較的均一に照射することができる。この場合、光沢のある表面の反射を避けるために、法線方向の入射光線を遮るようにしている。

【００３５】

図１では、曲がりくねった経路１２は、断片状の帯、すなわちオリジナル１４の端から端まで続く経路を４つ持つように描かれている。殆どの好適な適用例において有用なイメージ・センサ２２は、２５．４ｍｍから１０１．６ｍｍの間の領域、すなわちＡ４の用紙を４～５回の帯で走査できる長さを有している。後でより充実した説明を行うが、原イメージの忠実な復元を生成するための縫合処理ができるように、帯は互いに重なり合う領域を含んでいる。

【００３６】

走査装置１０は、通常、少なくとも１つのナビゲーション・センサ２４又は２６を含んでいる。好ましい実施例では、該装置は、一対のナビゲーション・センサを含み、各センサはイメージ・センサ２２の端部に対向して配置される。光電素子からなる１次元アレイを用いてもよいが、好ましい実施例では、各ナビゲーション・センサは素子の２次元アレイ

でできている。ナビゲーション・センサ２４及び２６は、オリジナル１４に対する走査装置１０の移動を追跡するのに使用される。

【００３７】

好ましい実施例では、各ナビゲーション・センサ２４及び２６は、オリジナルの固有の構造的な特性に関連するイメージを捕捉して、走査装置１０の位置に関する情報を生成するようになっている。従来技術に係る走査装置の大部分では、固有の構造的な特徴はノイズとして捉えられていた。図１乃至図３で示す本走査装置１０の場合、このような特徴はイメージ・センサ２２に関してはノイズであるが、位置情報を生成するための基礎をナビゲーション・センサ２４及び２６に付与するために利用することができるのである。有益で、高いコントラストを持つ表面組織のイメージは、媒体に固有な、若しくは繊維のように媒体上に形成された構造上の変動を検出することによって生成される。例えば、固有の構造的特徴を構成する谷間の陰と頂上の明るいスポットとのコントラストに基づいてイメージを形成してもよい。このような特徴は、一般には微視的な性質を持ち、印刷媒体では１０μｍ乃至４０μｍの間のサイズを持つ。あるいは、コヒーレントな光線の光沢のある反射は明るい領域と暗い領域とからなるコントラスト・パターンを生成するので、小斑点を用いてもよい。第３のコントラストの情報源は色である。色コントラストは、表面の組織には依存しない。たとえ繊維のない表面を可視領域の光で照射する場合であっても、異なる色の領域の間、例えば異なる灰色の陰の間で、色コントラストは存在する。

【００３８】

しかしながら、本発明を、ナビゲーション情報がオリジナルの固有の構造上の特性に依存しないような適用例に利用することが熟考されている。Ｘ，Ｙ及び角度の位置情報を得て処理するために、非イメージ的な手法を用いることもできる。このような代替例は、米国特許第５５７８１３号で議論されている。

【００３９】

図２に示すナビゲーション・センサ２４及び２６は、オリジナル１４のイメージ上で移動するウィンドウを効果的に観察して、連続的な観察を行う間の２平面内の配置の表示を生成することができる。後に説明するが、ナビゲーション・センサから出力される画素値を処理することで、イメージ・センサ２２から出力されたイメージ・データを適切にマッピングすることができる。ある特定の画素とこれに直近の画素の値は、各画素位置における相関値のアレイを生成するのに利用される。相関値は、表面の構造についての現在のイメージと固有の構造的特徴の既知の位置を表した蓄積されたイメージとの間の比較に基づいており、蓄積されたイメージは位置の基準として機能する。しかしながら、相関処理以外の処理はを用いて入力イメージ・データを操作しすることによって、出力イメージを形成してもよい。

【００４０】

ここで、図４と図５を参照すると、ナビゲーション・センサ２４は、照明光学系と機能的に関連しているように描かれている。もし、オリジナル１４が紙繊維がナビゲーション・センサ２４によって検出可能な紙製品であれば、かするような入射角で光を導くことが好ましい。本質的ではないが、１又は複数の発光ダイオード（ＬＥＤ）２８を用いてもよい。かするような角度３０とは、入射角の余角のことであり、好ましくは０°から１５°の範囲である。但し、オリジナル１４の特性によっては、値が変わることもある。図５では、照明光学系３４を備えた光源２８が描かれている。この光学系は、単一の要素若しくはレンズの組合せと、適切にコリメートされ目標表面を均一に照明するためのフィルタ及び／又はホログラフ要素を備えていてもよい。ナビゲーションに有効な空間周波数情報が向上するように、光源２８から放出される光の波長を選定するべきである。照射領域における固定パターン・ノイズを最小限にするべきである。インクや他の作因を持った印刷素材の上を走査装置が進行する間、媒体が持つ動的範囲が広い反射性を受容するために、光源２８の出力は調整を要することもある。

【００４１】

図４では、光源３５からの光は照明光学系３６でコリメートされ、次いで、アンプ型ビー

ム・スプリッタ３７によって方向変換される。ＬＥＤが出力した光エネルギの一部は、直接に、及び、図４には示されていないビーム・スプリッタを経由して、伝送される。ビーム・スプリッタからの光エネルギは、オリジナル１４を、表面の法線方向から照射する。

【００４２】

図４には、反射し若しくはオリジナル１４からまき散らされてビーム・スプリッタ３７を通過した光エネルギの成分も描かれている。通過光は、要素３８において開口及びフィルタにかけられ、要素３９においてイメージを結像されている。オリジナルからビーム・スプリッタに進み、ビーム・スプリッタで反射した光エネルギ成分は、図示されていない。ナビゲーション・イメージ捕捉光学系の拡大率は、結像した光を検出する２次元センサ・アレイ２４の視野内では一定でなければならない。多くの適用例では、ナビゲーション光学系の変調変換機能、すなわち光学的周波数応答の振幅計測は、ナイキスト周波数の手前の減衰を与えるようでなければならない。但し、ナイキスト周波数は、ナビゲーション・センサのセンサ素子のピッチと光学要素の拡大率とで決まる。光学要素は、背景の照明がノイズを生成するのを回避するようにデザインしなければならない。ウェーブフロント・スプリット型のビーム・スプリッタも利用できる点に留意されたい。

【００４３】

入射角は、オリジナルの材質的な特性に基づいて選定される。かすめるような照射角度は、長い陰および明らかなコントラストを生成し、もしオリジナルの表面に光沢がなければＡＣ信号を生成する。照射角がオリジナルの法線方向に近づくに従って、ＤＣ信号のレベルが増大する。

【００４４】

オリジナルの表面が微視的レベルでは相当程度の凹凸を含んでいるような場合には、オリジナル１４の目標領域をかすめるような角度３０で照射することが好適である。例えば、オリジナルが筆墨紙類、ボール紙、織物、あるいは人の皮膚である場合には、光源２８から出力された光をかすめるような角度で入射すれば、固有の構造的特徴に関する高いＳＮ（信号／ノイズ）比のデータを得ることができる。他方、写真、光沢のある雑誌のページ、オーバーヘッドの透明フィルムのようなオリジナル上でのスキャナの移動を追跡するために位置データが必要であるような場合には、法線方向の入射角のコヒーレントでない光を用いるのが好ましいであろう。コヒーレントでない光を法線方向の入射角で用いた場合、反射領域におけるオリジナルを眺めることによって、組織の内容に富んだイメージを得ることができ、イメージとこれに相関のあるナビゲーションが判る。オリジナルの表面は、微視的な浮き彫りを持っているので、その表面はあたかもモザイク状のタイルか切子面であるかのごとくに光を反射する。オリジナルを構成する無数の「タイル」は、法線方向から僅か散乱した方向に光を反射する。散乱光を含む領域と反射光とを見ることによって、法線方向に対して幾分区々に傾いている無数のタイルで構成されているかのように、表面をモデル化することができる。このようなモデリングは、"Analysis of Light Scattered from a Surface of Low Gloss into Its Specular and Diffuse Components"（Proc. Phys. Soc., Vol.51, 第２７４－２９２ページ（１９３９））と題されたＷ．Ｗ．バーカスの記事と同様である。

【００４５】

図４には、光源３５によるコヒーレントでない光を照射する様子を示している。同図では、照射光はオリジナル１４の表面の法線方向を向いている。図５では、かすめるような角度３０での照射を図解している。第３の実施例では、照射は行わない。その代わり、ナビゲーション情報は、背景光すなわち周囲からの光を用いて蓄積される。

【００４６】

第４の実施例では、法線方向の入射角でコヒーレントな照射光を当てることによって、小斑点に基づいたナビゲーションを可能にしている。走査装置とオリジナルとの相対移動は、ナビゲーション・センサに対する小斑点の動きを監視することによって追跡できる。もし、イメージ光学系なしにコヒーレントな照射を用いたならば、小さな照射領域を選ぶとともにオリジナルの表面とナビゲーション・センサ２４のフォトデテクタとの間隔を比較

的大きくとることによって、ナイキストのサンプリング基準を充分満たす程に充分大きなコヒーレントな照射を持つ主色の小斑点のセル・サイズを得ることができる。ビーム・スプリッタを用いることで、入射光と検出される散乱光の両方向の光を、オリジナルの表面に対して法線方向に近づけることができる。これは、図４に示したものと同様に実現される。

【００４７】

次いで、図６を参照すると、スキャナ１０が表面に印影されたブロック４６を含むオリジナル４４の上を移動する様子が示されている。スキャナ１０はオリジナルの表面ではいかなる運動学的な拘束も受けていないので、ユーザは、丁度手と前腕が肘まわりで回転するように、オリジナルに対して曲がりくねった経路をたどる傾向にある。図６では、走査装置はブロック４６を横切るように曲がった経路４８をたどっている様子が示されている。もし、走査装置の底端縁の方が回転軸を定義する肘に近ければ、該底端縁が短い半径となろう。その結果として、イメージ・センサを構成する各イメージ素子がブロック４６上を通過するのに要する時間と距離は区々になる。破線で示されているように、該装置が第２の位置５２に移動したときには、ブロックの歪んだイメージ５０が捕捉されることになる。

【００４８】

捕捉されたイメージ５０は、後述の処理を施していない状態で格納されたイメージである。しかしながら、イメージ・センサがブロック４６に関するデータを捕捉することによって、ナビゲーション情報が得られる。上述した実施例では、１又はそれ以上のナビゲーション・センサがオリジナル４４の固有の構造的特徴に関するデータを捕捉するようになっている。ブロック４６に対するイメージ・センサの移動を決定するために、走査装置１０に対する固有の構造的特徴の移動が追跡される。これによって、忠実に捕捉されたイメージ５４が形成されるであろう。イメージ５４のことを、ここでは「修正された」イメージと呼ぶことにする。

【００４９】

図７には、ナビゲーション処理の１つの実施例が示されている。ナビゲーション処理は、固有の構造的特徴に関するデータのようなナビゲーション情報についての一連の相関フレームによって実現される。相関とは、一連のフレーム中の固有の構造的特徴の位置を比較するものであり、これによってある特定の時刻におけるナビゲーション・センサの位置に関する情報が得られる。次いで、ナビゲーション情報は、イメージ・データを修正するのに利用される。図７に示した処理は、通常、各ナビゲーション・センサによって実現される。

【００５０】

第１のステップ５６では、参照フレームを獲得する。要するに、参照フレームとは開始位置のことである。後の時刻におけるナビゲーション・センサの位置は、該時刻にナビゲーション・センサから出力された位置データのサンプル・フレームをステップ５８で獲得し、次いで、ステップ６０での参照フレームと後に獲得されたサンプル・フレームとの相関を計算することによって、決定される。

【００５１】

初期参照（基準）フレームを獲得するステップ５６は、イメージ処理の開始に伴って行われる。例えば、この獲得は、単に走査装置をオリジナル上に接触させることをトリガにして実行されてもよい。あるいは、走査装置は、イメージ処理とナビゲーション処理を開始するためのスタート・ボタンを含んでいてもよい。各ナビゲータの照射系に周期的なパルスが発生することによって開始してもよい。もし、規定されたしきい値反射を越える反射信号があったり移動を示す相関信号があれば、参照フレームが獲得される。

【００５２】

ナビゲーション処理はコンピュータ処理によって実現されるものであるが、この実施例の概念は、図７と図８を参照して説明することができる。ここでは、Ｔ字形の固有の構造的特徴６４のイメージを含んだ参照フレーム６２が示されている。参照フレームのサイズは

、例えば走査装置の最大走査速度や、主要な空間周波数、センサのイメージ解像度のような要因に依存する。３２画素（Ｎ）と６４画素（Ｍ）で構成されるナビゲーション・センサの参照フレームの実際のサイズは、２４×５６画素である。

【００５３】

後の時（ｄｔ）では、ナビゲーション・センサは、フレーム６２に変位したサンプル・フレーム６６を獲得するが、該フレームは、実質的に同じ固有の構造的特徴を示すものである。ｄｔという期間は、Ｔ字形の特徴６４が走査装置の移動速度にしてナビゲーション・センサの１ピクセル分よりも小さくなるように設定するのが好ましい。６００ｄｐｉで速度が０．４５ｍ／ｓｅｃの場合、受容しうる時間間隔は５０マイクロ秒である。この相対的な変位のことをここでは、「マイクロステップ」と呼ぶ。

【００５４】

もし、この時間間隔のうちに、走査装置が参照フレーム６２を獲得するステップ５６からサンプル・フレーム６６を獲得するステップ５８まで進んだならば、Ｔ字形の特徴からなる第１及び第２のイメージは、特徴が推移してしまったものとなるであろう。好ましい実施例では、ｄｔは完全な１ピクセル分の移動に要する時間よりも小さいので、図８に示した概念図は、特徴６４が１ピクセルだけ上と右に移動することが許容される。完全な１ピクセルの移動は、単に説明を簡素化するために仮定したに過ぎない。

【００５５】

図８に示す要素７０は、フレーム６８の画素値の８個の直近の画素への連続的な移動を表示したものである。すなわち、ステップ"０"は移動は含まず、ステップ"１"は左上方への対角線的な移動で、ステップ"２"は上方への移動、云々である。このようにして、画素単位で移動したフレームをサンプル・フレーム６６と組み合わせて、位置フレームのアレイ７２を生成する。「位置０」で指される位置フレームは、移動を含まないので、結果はフレーム６６と６８の単なる組み合わせである。「位置３」は、陰のかかった画素数が最小であるが、それゆえ、最も相関のあるフレームということになる。相関結果に基づいて、サンプル・フレーム６６中のＴ字形の特徴６４の位置が、早期に獲得された参照フレーム６２と同じ特徴の位置に対して右上方の対角線であると決定される。このことは、走査装置が時間ｄｔの間に左下方に移動したことを意味している。

【００５６】

相関を求めるために他の手法を採用することもできるが、受容しうる手法は「差分の自乗の合計」と呼ばれる相関方法である。図８に示す実施例では、９個の相関係数（$C_k = C_0$, $C_1 \cdots C_8$）が要素７０の９のオフセットから形成されている。各係数は以下の方程式により決定される。

【００５７】

【数１】

$$C_k = \Sigma_i \Sigma_j \left(S_{ij} - R_{(ij)+k} \right)^2$$

【００５８】

ここで、S_{ij}は位置ｉｊにおいて同じフレーム６６をナビゲーション・センサで計測した値のことであり、R_{ij}は要素７０においてｋ方向に移動したときのフレーム６８をナビゲーション・センサで計測した値である。図８では、ｋ＝３が最も低い相関係数を持っている。

【００５９】

連続するフレームの中から同一の特徴の位置を見付けるのに相関が用いられ、この結果、フレーム間での特徴の変位を決定することができる。走査処理が進歩したことに伴い、これらの変位の合計若しくは積分をとり、関連する光学系の設計に依拠する寸法効果を修正することで、イメージ・センサの変位を決定することができる。

【００６０】

単一の画素の寸法を越えない変位を確保できる程度に充分速いフレーム速度が選ばれているので、既に述べたようにフレーム間の相関は「マイクロステップ」と呼ばれる。過度にサンプリングすれば、サブ画素変位の精度が得られる。図７を参照すると、相関を計算す

るための各計算ステップ６４が行われた後に、マイクロステップが行われたかを判断する判断ステップ７４が実行される。もしマイクロステップが必要であれば、参照フレームを移動するステップ７６が実行される。このステップでは、図８に示したサンプル・フレーム６６が参照フレームとなり、新たなサンプル・フレームが獲得される。次いで、相関の計算処理が繰り返される。

【００６１】

この処理では、相当程度の相関の照合が行われるので、ステップ７６でサンプル・フレーム６６を連続的に移動して参照フレームを指定する際に起こりうるいかなるエラーも蓄積される。この「ランダム・ワーク」誤差の成長速度に何らかの制限を置くために、サンプル・フレームは別個のバッファ・メモリに格納される。この別個に格納されたサンプル・フレームは、後続の相関計算を行うときには新しい参照フレームになる。後者の相関は「マクロステップ」と呼ばれる。

【００６２】

マクロステップを用いるとことで、ｍ個のイメージ・フレームの変位の距離、すなわちｍ個のマイクロステップにまたがったスキャナの変位をより正確に決定することができる。１つのマクロステップに含まれる誤差は、単一の相関計算の産物であり、他方、ｍ個のマイクロステップと等価な誤差は単一のマイクロステップに含まれる誤差の$m^{1/2}$である。ｍ個のマイクロステップに含まれる誤差の平均はｍの数を増やすことによってゼロに近づくけれども、誤差の平均に含まれる標準偏差は$m^{1/2}$として増大する。このように、実施可能な範囲で大きなｍを持ったマクロステップを用いて、蓄積した誤差の標準偏差を減少させることは有益である。但し、マクロステップを定義する２つのフレームは、共通のイメージ内容を持つ顕著な領域を持たない程には離れていない範囲でなければならない。

【００６３】

サンプル周期ｄｔは一定である必要はない。サンプル周期は以前の測定に関する関数として決定してもよい。変動するｄｔを用いる１つの方法によれば、連続的な参照フレーム間の相対変位をある範囲内に保つことによって、変位計算の精度を向上することができる。例えば、上限は１画素の変位であり、下限はナビゲーション・データの処理における四捨五入を考慮して決定される。

【００６４】

図９を参照すると、イメージ・センサ２２で生成されるイメージ信号には、ナビゲーション・データに基づいて「位置タグ」が付される。１つの実施例では、ナビゲーション・センサ２４及び２６から出力された画素値は、ナビゲーション・プロセッサ８０が受け取って、図７及び図８に示す処理を実行するようになっている。計算された相関に基づいて、第１のナビゲーション・センサ２４と第２のナビゲーション・センサ２６の現在位置の座標が決定される。

【００６５】

マクロステップの中では、ナビゲーション・プロセッサ８０は、各ナビゲーション・センサの移動に含まれる並進成分を直接修復する。走査ヘッドの全ての回転を考慮に入れた上で概算された絶対座標を得るために、両方のナビゲーション・センサから出力されたデータを積分しなければならない。プロセッサ８０は、初期の向きに対する走査ヘッドの現在の向きのモデルを維持する要になっている。各ナビゲーション・センサの並進の概算は、このモデルに関連して解釈される。スキャナ・モデルの向きは、順に、それ自身で周期的に更新される。

【００６６】

１つの実施例では、各ナビゲーション・センサのマクロステップは同期がとられているので、もしナビゲーション・プロセッサ８０が一方のナビゲーション・センサ内のマクロステップを必要とするならば、他方のマクロステップも実行させることになる。これによって、各ナビゲーション・センサによって記録された並進の解釈が簡素化される。もし、マクロステップＴで、スキャナの向きが垂直に対してθ度であれば、各ナビゲーション・センサのマイクロステップで記録された並進は後述のように解釈される。

【００６７】
図１０は、番いをなすナビゲーション・センサの動きを示している。各々について、スキャナの向きを規定する座標フレームに対する並進が記録されている（単位ベクトルにはｕ及びｖというラベルが付されている）。第１及び第２のナビゲーション・センサは、夫々、（ｕ１，ｖ１）及び（ｕ２，ｖ２）という大きさを持っている。これらの並進を解釈する目的は、世界座標フレームに対するナビゲーション・センサの位置を更新することにある。要するに、断片的で線形的な概算から経路を復元する訳である。但し、線形部分の大きさは、マクロステップのサイズによって定まる。
【００６８】
以下に示す標準的な三角法の関係に従えば、各単位ベクトルは、スキャナの向きθと関係がある。
【００６９】
【数２】
ｖ＝（ｓｉｎθ，ｃｏｓθ）
ｕ＝（ｖｏｓθ，－ｓｉｎθ）
【００７０】
しかしながら、高価につく三角関数のコンピュータ計算なしに、ｕとｖを概算し続けるべきである。
【００７１】
図１０において、Ｐ１とＰ２は、マクロステップの開始時におけるナビゲーション・センサの世界フレームに対する位置を表している。各々は、（ｘ１，ｙ１）及び（ｘ２，ｙ２）という座標値を持っている。更新された位置、すなわち、後のナビゲーション・センサ・フレームにはＰ１'とＰ２'が付されている。これらは同じ世界フレームに対する位置であり、夫々、（ｘ１'，ｙ１'）及び（ｘ２'，ｙ２'）という座標値を持っている。
【００７２】
限界では（正確な積分を実現するために充分接近していると仮定しているが）、各ナビゲーション・センサの並進のｖ成分は、同じでなければならない（各端部で等しい影響を共有し、回転の限界ではｖ成分を全く招来しない）。ここで、ｖ１とｖ２の平均値をとってみる。
【００７３】
【数３】
ｖ＝（ｖ１＋ｖ２）
【００７４】
このような場合、更新された端点の位置は下式で与えられる。
【００７５】
【数４】
Ｐ１'＝Ｐ１＋ｕ１ｕ＋ｖｕ
Ｐ２'＝Ｐ２＋ｕ２ｕ＋ｖｕ
【００７６】
この時点では、個々のナビゲーション・センサよりもむしろイメージ・センサの物理的な端点に対応した位置概算を行う方が便宜である。これは、ナビゲーション・センサとイメージ・センサの物理位置に関係するキャリブレーション・データを用いて行われる。簡素化のため、ここでの端点の位置は、イメージ・センサの最初と最後のセンサ素子の中心に対応するとしている。
【００７７】
周期的に慣性フレームを更新しなければならない。但し、１つのマクロ・フレームの最後又は次のマクロ・フレームの最初に行うだけでよい。マクロステップは同期がとられていなければならない。すなわち、マイクロステップの最大数を越えているがために、ナビゲーション・センサが次のマクロステップをトリガしたならば、たとえ他のナビゲーション・センサが静止状態を保っていたとしても該センサ内でもマクロステップを開始しなけれ

ばならない。このような手法によれば、ナビゲーション・フレームの回転は常に常に小さい、という付加的な利点がある。これは、今までに経験した最大回転がナビゲーション・センサ間のベースラインで分けられたマイクロステップの最大数よりも常に小さいからである。

【００７８】

慣性フレームを更新することで、スキャナが移動する際に失われた回転成分を取り戻すという効果がある。これは、個々のナビゲーション・センサが経た並進の差分によって暗黙的に決定される。慣性座標フレームは、移動の回転成分を明示的に計算するよりも、スキャナの主軸（ナビゲーション・センサの位置に結ばれる線）との関係から直接的に更新される。慣性フレームのｖ軸は、ナビゲーション・センサ１からナビゲーション・センサ２に向かう主軸に沿った単位ベクトルであり、そのｕ軸はこれに直交する単位ベクトルであり、２Ｄで完結する。

【００７９】

更新されたｖの値は下式で与えられる。

【００８０】

【数５】

$$\begin{aligned} v' &= (v'_x, v'_y) \\ &= (P2'-P1') / |P2'-P1'| \\ &= (P2'-P1') / D \end{aligned}$$

【００８１】

また、これに直交するベクトルｕは下式で与えられる。

【００８２】

【数６】

$$u' = (v'_y - v'_x)$$

【００８３】

長さ｜Ｐ２'－Ｐ１'｜は、スキャナの固定長（モジュール・ナビゲーション誤差）であり、図１０ではＤが付されている。各マクロステップで２乗の合計の平方根を計算する手間を省くので、これは慣性フレームの計算を大いに簡素化する。

【００８４】

ナビゲーション・プロセッサ８０は、イメージ・センサ２２が出力するＮ個の画素値も、画素増幅器８２及びアナログ－デジタル変換器８４を経由して受け取る。図９では、イメージ・センサ２２からの単一のタップと単一のＡ／Ｄ変換器８４しか示されていないが、複数のタップが各Ａ／Ｄ変換器毎に設けられていても本発明の要旨の範囲である。現在の位置座標は、イメージ・センサ内の無数の数の画素に対応するライン・データの端部で「タグ」が付されている。ナビゲーション・プロセッサ８０の出力８６は、それゆえ、位置タグ・データ・ストリームと呼ばれる。

【００８５】

ナビゲーション・プロセッサ８０は、イメージ・センサ２２から画素データを、ナビゲーション・センサ２４，２６から位置情報を、同じフレーム速度で受け取ったりする必要は必ずしもない。位置データのストリームは、一般に線形センサ・アレイの画素ピッチよりも高い空間解像度なので、線形センサの端点の位置の概算を更新するのに用いてもよい。線形イメージ・センサの最適なクロック周波数は、一般に、スキャナの移動速度によって定まる。これによって、最も移動が速い端点では、線形センサ自身の画素ピッチよりも丁度少ない数だけサンプリングされることになる。そうでなければ、過度のサンプリングかサンプリングの欠落が招来される。サンプリングが過度であったならば、イメージ処理パイプラインに沿ったコンピュータ及びデータ処理上の要件が増大するので、イメージ品位にはあまり若しくは全く寄与しない。他方、サンプリングが足りない場合には、イメージ品位は明らかに低下する。

【００８６】
１つの解決策は、過度のサンプリングを常に招来するような固定周波数で線形センサをサンプリングを行うことである。最高速の走査速度についてのサンプリング要件を越えるようなものであってよい。もしスキャナが最高走査速度よりも遅く動けば、ナビゲーション・センサが、現在の端点の位置の概算に基づいて不要なサンプルを廃棄するようになっている。すなわち、有効な線形センサのサンプルとタグが付された位置データがナビゲーション・プロセッサ８０から最後に出力された以降、各端点の相対位置の大きさは、さらに増分の出力をゲートするために用いられる。最も簡素な方法は、１つ又は他の大きさが画素ピッチ（若しくはその有効少数部）と厳密に等しいか又はそれを越えた場合のみ出力を許すことである。この手法によれば、とりわけ最大走査速度よりも丁度小さい速度で走査する場合には、サンプリング不足となる。あるいは、サンプリングの不足を回避するために、もし次の線形センサのサンプルの相対端点位置の予測が画素ピッチを越えるならば、現在のサンプルを出力するようにする。ここで言う予測は、端点位置の変動速度（あるいは高い数値の導関数）に基づいている。サンプリング不足を回避する第３の手法は、増分バッファをナビゲーション・プロセッサ８０に導入することである。このようにすれば、現在の増分の相対位置データが画素ピッチを越える場合には、以前の増分が伝送される。
【００８７】
上述のサンプリング施策の各々は、同期サンプリングに基づくものであり、一般には幾分の過度のサンプリング又はサンプリング不足を伴う。より理想に近い空間サンプリングを実現する好ましい総合的な解決策は、ナビゲーション・プロセッサ８０が線形センサの捕捉に非同期でトリガを与えることである。１つの手法は、捕捉と線形センサのサンプリングのトリガを丁度よい時間で与える最良の点を同定するために、最後の有効な捕捉（若しくは将来のその予測）以降の相対端点位置を直接用いることである。
【００８８】
図１１には、データ・ストリームの増分８８が示されている。該データ・ストリームは、位置座標セル９０，９２，９４及び９６をＮ個の画素セルの対向する端部に含んでいるが、その順序は必須ではない。
【００８９】
イメージ・センサ２２は、走査装置がオリジナルの上を移動するに従ってクロックされる。既に議論されたように、クロックすることによって、最高速で移動するセンサ素子が画素の変位当たりに少なくとも一回サンプリングすることを担保できる。
【００９０】
ナビゲーション・センサ８０の出力８６において位置タグが付されたデータ・ストリームは、図１２に示すようにバッファリングされる。バッファＢは、図１１に示す増分８８を複数備えている。バッファＢを走査全体を収容するために使用してもよい。このような場合、データ圧縮を用いてもよい。このデータは、次いで、直線で囲まれた座標系にマッピングされ、本発明に係る方法に従って修正されて、復元された最終的なイメージを形成するが、この点について以下に説明する。
【００９１】
国際特許出願公開ＷＯ９６／２７２５７号や米国特許第５５７８８１３号で教示された従来技術の方法によれば、直線で囲まれた座標系へのマッピングは、直線状で所定サイズの格子にマッピングすることによって達成される。直線で囲まれたイメージを単一の２次元的なアレイとして表現するので、大容量の作業メモリが必要ということと、予め決められたページの範囲内に走査経路が限定されるという２つの欠点がある。したがって、本発明の実施例では、動的タイリング・システムを採用した改良方法を提供する。このようなシステムでは、イメージ空間は、離散的なタイルの数としてとして表される。タイルは、必要に応じてラン・タイムで割り振られている。
【００９２】
このようなタイル・ベースで表現することによって、データ捕捉の実際の経路には、限られた時勢に進んでいる必要がなくなる。後に詳解する一般的なタイル・インデックス手法

を用いることによって、上述のようなハンド・スキャナを用いていかなる形状の領域を走査することもできる。走査全体の寸法は、走査された総領域のみによって制限され、これは利用可能なメモリを越えている。さらに、本発明の実施例では、タイルが「非有効」になり、他のタイルの処理を決定する必要がなくなったとき（すなわち、修正目的ではそれらが必要なくなったとき）には、データ捕捉を続ける間、これらのタイルは進行中に圧縮することができる。この圧縮によって、処理の間に、直線で囲まれた（rectilinear：直線で囲まれた、以下、直線状または直線的という）イメージを格納するのに必要な作業メモリの総量を大幅に削減できる。

【００９３】

好ましい方法では、有効なタイルの各々は、数バイトからなる２次元アレイとして表される。ここで、各バイトは１つの画素を表す。単一バイトで複数の画素を、又は画素当たり複数のバイトを用いるような他の画素表示も、容易に受け入れられる。便宜のため、２次元の寸法（高さと幅）を持ったタイルが選ばれている。６４×６４はとりわけ適切な選択である。以下で議論される実施例ではこの寸法を採用している。各タイルの左上手の角に関連した直線で囲まれたイメージの位置は、幾何学的手法によって導き出された複数のタイル・サイズである。タイルは重なり合わない。タイルは、これらが表す直線状の領域に隣接している。

【００９４】

個々のタイルは、以下の３状態のうちの１つに属する。

ｅｍｔｐｙ（空）：　現在タイルが存在しないことを意味する。

ａｃｔｉｖｅ（有効、活動）：　２次元アレイで表された現在有効なタイル

ｃｏｍｐｒｅｓｓｅｄ（圧縮）：　既に圧縮された完全なタイル

【００９５】

タイルをインデックス化することは、このような方法を効率的に処理するためには重要である。このことについて、好ましいタイリング手法で採用されている手法は、２つの主要な要素を有している。

【００９６】

・　走査イメージを表すタイルのリストを保管するとともに、個々のタイルを空間的にインデックスするための方法を提供する、インデックス化の手法

【００９７】

・　修正処理の間に発生するイメージへの頻繁なアクセスの表示に基づいた、タイルを使用するオーバーヘッドを最小限にするための、現在の有効なタイルについての高速且つローカルなインデックス

【００９８】

直線状のイメージの座標の各々は、２個の符号なし１６ビット値として表される。このうち１つはｘであり他の１つはｙである。６４×６４のタイルの場合、ｘ及びｙの上位１０ビットは、タイル・インデックスを提供し、これは、当該直線状の画素が見付けだされたタイルを同定するために用いられる。ｘ及びｙの下位６ビットは、タイル内での画素位置を与えている。これらは、有効なタイルを表す２次元アレイのインデックスとして用いられる。もし上位１０ビットによって同定されたタイルがｅｍｐｔｙ、若しくは既に圧縮されていれば、問題の画素をインデックスするために、下位６ビットを直接用いることはできない。これら２つの場合、メモリにそのタイルが割り当てられるか、又は、タイルを伸長する必要がある。伸長に代わるものとして、付加的なタイルが割り当てられる。このタイルは、先に形成されたタイルと後で統合しなければならない。伸長と統合は高価なコンピュータ処理であり、処理の効率化のためには可能な限り省力化するべきである。

【００９９】

１０ビットは、１つの側で１０２４個のタイルからなるタイル空間のインデックスを許容する。６４×６４個のタイルは、３００ｄｐｉであれば、各側が大凡１／５インチであり、幅と深さが２００インチを越えるアドレス可能な空間を生成する。この仮想的なイメージ・サイズは、最も要求されている走査作業を満足させるであろう。例えば、走査経路の

原点がアドレス空間の中心近くに位置していれば、いかなる方向に対しても、タイル空間から外れるまでに１００インチに至るまで走査することができる。作業メモリの浪費や長期間の格納といった問題は、より顕著となる。これらについては以下で詳細に説明する。

【０１００】

タイル・インデックスを表すための特に便宜的な方法は、２次元アレイとして表すことである。このタイル・インデックス形式であれば、同じサイズのフラット・イメージをインデックスするのに、フラット・アレイよりも少ないメモリしか必要でないであろう（イメージの格納自体は無視する）。しかしながら、もし全ての１０ビットが等しく使用されていれば、この手法は、インデックスだけのために２ＭＢの作業メモリを必要とするであろう（各タイルの１６ビットをｅｍｐｔｙにすることを許したとして）。手持ち型スキャナのような適用例では、このような大容量のメモリは現在は非常に不利である。それゆえ、あまり作業メモリを消費しない代替手法が望まれている。

【０１０１】

タイルをインデックスするのにわずか８ビットしか使用しない小さなアレイを採用することが可能であることは、既に判明している。この場合、わずか１３０ｋＢの作業メモリしか必要とせず、また、幅及び高さ方向におよそ２５インチまで走査することが許される。以下の記述では、インデックス手法がいかにして構築されたかを示している。図１３は、このようなアレイ２０１を図解している（但し、わずか４個のインデックス・ビットしか示していない）。タイル・インデックス・アレイ中の各１６ビットのエントリは、ゼロ（これは、アレイ・エントリ２０２が空であることを意味する）に設定されているか、若しくはタイル・ディレクトリ２１１へのインデックス（これらのアレイ・エントリ２０３は図１３ではハッチングが施されている）を含んでいる。タイル・ディレクトリ２１１へのインデックスとして１６ビットを用いることによって、６４Ｋタイルへのアクセスが許容される。これは、３００ｄｐｉのハンド・スキャナに対して、走査の総領域を最大２，５６０平方インチに制限するものである。レター・サイズのページがおよそ１００平方インチ（２５００タイル）であると仮定すれば、何ら問題はなく、より小さなタイル・デレクトリであっても実用は可能である。

【０１０２】

各タイル・ディレクトリ・エントリは、圧縮されているか否かを示すためのフラグと、開始アドレスへのポインタと、圧縮バージョンを格納するのに使用するバイト数を示したレングス・パラメータとを含んでいる。有効で圧縮されたデータ・ブロックをメモリの別個のチャンク（chunk:塊）に保管するのが便利である。結果として、走査の完了に応じて、全てのブロックは圧縮される。圧縮データを作業メモリから切り離すことによって、圧縮メモリを合体したりこれを長期間の保存に転送するために要する余分な処理を最小限にすることができる。nextAvailableDirectoryEntry（次に利用可能なディレクトリのエントリ）というポインタは、どのディレクトリ・エントリが既に使用されているかを示している。これらは、新しいタイルに出くわし、各々が走査当たりに１つのタイルを表すのに使用されるときに、連続的に割り当てられるようになっている。

【０１０３】

特に好ましい手法では、インデックスは４分木で表される。ここでは、タイル・インデックスの１０ビットが使用される場合について議論する。４分木の各ノードは、４個の子供しか持てない。全くいないか、幾つかいるか、あるいは全てにｎｉｌが代入されているかである。ノードは子供へのポインタを持っている。４個の子供（０，１，２，及び３と番号が付されている）は、空間位置に対応する。左上、右上、左下・右下の各々である。木の各レベルは、空間解像度を表しており、この木の直近上位のレベルの２倍の解像度を持つ。この例では、木には１０レベルがあり、タイル・インデックスの１０ビットの各々が１つのレベルに対応し、夫々がｘ及びｙ座標を含んでいる。各レベルでは、ｙインデックス中の適当なビットが最上位又は最下位を示し（夫々０と１である）、ｘインデックスの対応するビットは左又は右を示す（夫々０と１である）。このようにして、ｙビットとこれに続くｘビットで形成される２ビットの番号によって、木の現在のレベルにおける子供

のノード番号が与えられる。この木の「葉ノード」（最も低いレベル、すなわちレベル１０）は、空ゼロに設定される）か、若しくは、タイル・ディレクトリを指している。これは、図１３を参照しながら説明した通りに表される。

【０１０４】

図１４と図１５は、図１３に示した、上述の簡単な２次元のタイル・インデックスについての有効で圧縮されたタイルを表した４ビットの木を示している。図１４は、タイル・インデックスを空間表示したものであり、インデックスによってカバーされた異なる領域に存在する異なる空間解像度を示している。走査領域を表して有効若しくは圧縮されたタイルを持つ領域のみが、インデックス中では高い解像度が表示される。図１５は、空間インデックスに対応するインデックス木を示している。タイル・ディレクトリ・エントリを指している満杯の葉ノードは、木の対応する葉ノードの下に垂直の線で示している。図１４には、４ビットの２進表示で０１０１のｙインデックスと４ビットの２進表示で１０００のｘインデックスを持つタイル・エントリ２２４が示されている。対応するアレイ・セルは、図１３のセル２０４として示されている。図１５には、関連する葉ノード２２５が示されている。図で分かるように、２つのインデックスの連続的なビットの各組で形成される２進表示の番号（各ケースでｙビットが最初に与えられている）は、木の各レベルにおいて葉ノードに導くノード番号を意味する。同図の場合、０１（ノード番号１），１０（ノード番号２），００（ノード番号０），最後に１０（ノード番号１）の順である。

【０１０５】

図１６は、アレイの４つ組２１４としての木を表している。実用例では、１６ビットを持つであろうが（すなわち、ｘ及びｙの各次元で１０のインデックス・ビット）、図１６では４ビットのケースを図解している。アレイ中の最初の４つ組は、常に木のルートノードである。現在の木に含まれる他の全てのノードのアレイ中の位置は、木が構築された順番に依存している。木の構築については以下で説明する。木のインデックスについて説明する前に、木は適切に構築され、且つ、アドレスしたいタイルは既に木の中にあると仮定しておく。

【０１０６】

図１４は、木をインデックスするために用いられるビットを示している。該インデックスは垂直に配列され、ｙ座標インデックスを左手に持っている。上述したように、対応するｙ及びｘ座標ビットの組は、常にルートノード（４つ組のアレイの最初のノード）の子供を示している。最初のビットの組で形成された２ビットのアドレスは、ルートノードの子供を表す４つ組アレイ内の１６ビットの要素の番号（０，１，２，３）を与える。この１６ビット位置に格納された値は、該ノードの子供を表す４つ組についての４つ組アレイ・インデックスを与える。そして、再びこのような４つの子供が現れる。ビットの２番目の組は、同じレベルにおける子供を示している。この処理は、１０番目の組の間で同様に続く（図１６に示す限定的なケースでは、４番目の組まで）。４つ組の値は、ここでは、タイル・ディレクトリ・エントリ番号を与える。木の最後のノードは葉ノードである。

【０１０７】

４分木構造によれば、タイル・インデックスを効率的に表すことができる。木の各葉ノードは、４つのタイル・ディレクトリ・エントリを表している。７００個の４つ組ノードを持つ木は、これを表すためには、丁度７００＊（４つの子供）＊（２バイト）だけ必要となるが、これは木全体でも６ＫＢに満たない。木に含まれる殆どのノードは葉ノードである。１／１６の数が祖父ノード（葉ノードの親の親という意味で）であり、１／６４が曽祖父ノード云々である。これら「先祖ノード」を系列的に合算することができる。これら系列は集結して先祖ノードの合計数になる。このようにして、２０００個のタイルを表した５００を越える葉ノードが設けられている。３００ｄｐｉで走査処理する場合、レター・サイズのページ全体に及ぶ走査領域としては充分な量を凌いでいる。このデータ構造でとりわけ有利な点は、どの方向にも走査できることである。４分木を表すためにさらに大きなアレイを用いた場合、走査可能な領域は線形的に増大する。

【０１０８】

ここでは、４分木の構築について説明する。これは全く複雑ではない。必要なのはポインタを付加することだけである。最高の４つ組アレイ・エントリを越えてポイントするときには、nextAvailableQuad（次に利用可能な４つ）が使われる。最初は、ルートノードが割り当てられているだけであり、その子供に対するポインタは全てゼロ（nil）（４つ組アレイ・インデックス値０で表される）であり、nextAvailavleQuadは４つ組アレイ・エントリ１を指している。木を構築するとき、これをインデックスするかのごとく木の下方に向かって作業する。もしいかなるレベルでも、ｙインデックスとｘインデックスの対応する組で構成される２ビット値がnilポインタを示しているならば、nextAvailableQuadの現在値は置き換えられ、次の４つ組アレイ位置でnextAvailableQuadは増分される。新たに生成されたノードの子供のための４つのエントリは、勿論、ゼロ（nil）を設定しなければならない。このようにして、木は構築により成長するのみである。木のノードを削除する方法を含める必要はない。

【０１０９】

図１７には、ハンド走査を行う際にタイルする方法が、フロー図の形態で示されている。フローには修正処理が含まれている。修正処理はタイルするときに発生する。しかしながら、修正処理の詳細については、ここでは議論せず、この明細書の後半で述べることにする。端点位置情報は、修正に使用される点の前に、タイル・マネージャ２３１によって、増分データのストリーム２３０から読み取られる。増分保持バッファ２３２によって、タイル・マネージャ２３１と修正ユニット２３７の間には適度の遅延時間が導かれる。これは、タイル・マネージャによって試行されたが修正はされなかった所定数の増分を保持するのに用いられる。以下の議論で明らかになるように、保持すべき増分の適切数は、（2^n+1）である（好ましい実施例では、一般に３３個の増分である）。但し、ｎは整数とする。増分保持バッファ２３２が一旦満杯になると、保持バッファ２３２中の全ての増分が修正されるまで、タイル・マネージメントは停止する。増分中の１つのチャンクである最後の増分が次のチャンクの最初の増分になる（ここでは、３３個の増分保持バッファ２３２のために、結合化タイリングと修正処理が３２個の増分ステップだけ進行する）。

【０１１０】

各タイル・マネージメントのラウンドで、増分保持バッファ２３２中の増分（前回のラウンドから繰り返される増分を含む）に関連した端点ジオメトリが解析され、現在有効なタイルのセットが同定される。これは、チャンク中の各増分の端点間を引いた線を通過するタイルのセットを考察することによって可能である。このような全てのタイルのスーパーセット（すなわち、チャンク中の増分によって交差される全てのタイルの組合せグループ）は、増分保持バッファ２３２が完了したと後に続く修正作業の間は有効なセットである。端点位置を結ぶ線が通過するタイルのセットは、標準フレーム・バッファ・ライン描画技術を用いることで決定できる。このような技術の一例は、ジェームス・Ｄ・フォレイ、アンドリース・ヴァン・ダム、スティーヴン・Ｋ・ファイナー、ジョン・Ｆ・フュフス著の"Computer Graphics:Princilple and Practice（コンピュータ・グラフィックス：原理と実用）"第２版，アディション－ウェズレイ出版社、リーディング、マサチューセッツ、１９９０、第７２－８１ページに説明されている。図１８及び図１９には、可能なタイルの空間の部分と、１つの増分の端点位置間を引いた線と交差する有効なタイルのサブセットとを示している。

【０１１１】

このような線が可能なタイルの空間に引かれる前に、直線状のイメージ画素ユニットからタイル・インデックス・ユニットまで、端点位置データのスケールを作らなければならない。ハンド走査には最適とされている６４×６４タイルの場合、６４による割り算を含んでいる。正確に同じタイルのセットが有効であることを担保するためには、全ての小数部の正確さを維持することが、実用上は強く望まれている。このことは、後の修正処理で直面するであろう。

【０１１２】

ここでは、２つの増分の間の空間は両者間の全てのタイルが横たわるのを許容するほどに

は大きくないと仮定されている。６４×６４のタイルの使用が考慮されたハンド走査の実施例では、この仮定は明らかに持ち堪えるであろう。何故ならば、増分間の距離の大きさは一般にせいぜい２出力画素ユニットしかないからである。実際、有効なタイルを同定するために使用される増分のセットを、普遍性を失うことなくサブ・サンプルして、コンピュータ処理のオーバーヘッドを低減するのは可能かもしれない。

【０１１３】

このようにして有効なタイルの各々は明らかにされるので、有効タイル・インデックスのタイル・インデックス・テーブルＮが付加される。線描画処理の間タイルに遭遇しているので、このタイルのインデックスは有効タイル・インデックスのテーブルＮに存在するものと比較される。もし互換性のあるエントリが既に存在すれば、新しいタイル・インデックスは無視される。そうでなければ、新しいタイル・インデックスはテーブルの最後に付加され、テーブル・サイズは１だけ増分する。各タイル・インデックスの組は、ｙインデックスに１０ビット、ｘインデックスに１０ビットを必要とする。好ましい実施例では、以下でさらに詳細に説明するように、ｙインデックスが先に、次いでｘインデックスが続くインデックスの組を表すのに、符合なし１６ビットの整数の組が使用される。

【０１１４】

現在有効なタイル・インデックスについてのテーブルＮは、重要なタイル・マネージャの活動性の根拠として使用される。これらは、

【０１１５】

・もはや有効でなく今は圧縮可能な以前の有効なタイルの同定

・新たに有効なタイルが存在しこれらのために割り当てられたメモリを持つことを担保するために、タイル・インデックス２１４を更新すること

・タイル・インデックス自身を参照することになしタイル・データ・メモリに高速アクセスするために、修正ユニット２３７によって使用されるタイル・キャッシュ２３６を構築すること

【０１１６】

この方法のとりわけ好ましい実施例では、有効タイル・データ・メモリバッファ２１２の中に、タイルのプールが割り当てられている。これは、各タイルによって表されるイメージの部分を表すバイト・データの２次元アレイからなる。有効なタイル・データ・メモリは、隣接する領域に仕切られ、各々の領域は１つのイメージ・タイルを表すには充分な大きさを持つ。有効なタイルの最低限必要な数は、いかなる時も有効なタイルの最大数に等しい。これは、増分保持バッファ２３２内の増分のチャンクの下に横たわるタイルの最大数から決定される。ハンドスキャナに適用した実施例を考察した場合のように、増分間の空間が２出力画素より小さいと仮定すると、チャンクの全体空間はせいぜい３２＊２すなわち６４個の出力画素ユニットであろう。直線で囲まれたイメージに対して４５度で走査する場合、最大水平範囲は６４＊√２すなわち９０画素であろう。最悪の場合、各行につき３タイルだけ広がる。スキャナに沿った最大タイル数は、スキャナの出力画素ユニットにおける長さＬによって定まり、部分タイル（Ｌが正確にタイル・サイズの倍数であれば１）を許容するために、タイル・サイズで割られ、＋２だけ切り取られる。これは、有効タイルの控えめ（スキャナが４５度で全ての長さに対しない場合のように）な、下式の最大数を与える。

【０１１７】

【数７】

maxActiveTiles（最大有効タイル）=3*(L/64+2)

【０１１８】

３００ｄｐｉと仮定すれば、２．５インチのスキャナの場合、この式は以下の通りとなる。

【０１１９】

【数８】

maxActiveTiles2.5=3*(750/64+2)=39 タイル

【０１２０】
５インチのスキャナでは、下式の通りとなる。
【０１２１】
【数９】
maxActiveTiles5=3*(1500/64+2)=75 タイル
【０１２２】
１１インチのスキャナの場合は、下式の通りとなる。但し、割り算の切り捨てがなされている点には留意されたい。
【０１２３】
【数１０】
maxActiveTiles11=90*(3300/64+2)=159 タイル
【０１２４】
ここで説明している実施例のように、グレイスケール・レベルのための１バイトを持つならば、６４×６４のタイルは、４ＫＢのメモリを必要とし、３００ｄｐｉの２．５，５，及び１１インチのスキャナは、夫々、１５６ＫＢ，３００ＫＢ，６３６ＫＢの活動タイル・データ・メモリを必要とする（解像度のレベルの相違や、その結果としてのメモリの要件の相違は、勿論、本発明の要旨の範囲内である）。これらの図は、もはや有効でない以前のタイルは全て、新しいタイルを生成する前に、複写され、圧縮され、圧縮タイル・データ・バッファ２１３に格納されているものと仮定している。この結果、メモリ資源は、もはや現在の増分のチャンクとは交差しなくなったタイルを表していた有効タイル・データ・バッファ２１２から有効タイルのプールに戻される。このことにより、タイル・インデックスとこれに関連するディレクトリ・エントリ経由で、有効タイル・データ・メモリを新しい有効タイルに再度割り当てることができる。この手法は、圧縮処理と修正処理を順に行う必要があるが、圧縮処理を最初に、次いで修正を行わなければならない。実際、並列処理を実装することが好ましい。好ましい実施例では、２重プロセッサ実装を許容するための最低条件に関しては、有効なタイルの数（すなわち有効タイル・データ・バッファ２１２のサイズ）は増大する。１つのプロセッサとこれをサポートするハードウェア・ユニットは、他のプロセッサが圧縮を実行する間に、修正処理を実行する。有効タイルの増加数は、連続するチャンクの間に存在しなければならない最低限の重なり合いの度合いによって決定される。
【０１２５】
ここで、新しい有効タイルがメモリに割り当てられることを担保するために、タイル・マネージャ２３１がどのようにしてタイル・インデックス２１４を更新しているかについて説明する。走査を開始するときには、タイル・インデックス２１４とタイル・ディレクトリ２１１のエントリは、空の４分木である。この時点では、タイル・インデックス２１４を表す４つ組のアレイは、アレイ位置０に単一のルートエントリを持ち、このルートエントリの４つの子ノード・ポインタは全てゼロ（nil）で、nextAvailableQuadは４つ組アレイ要素を１を指している。インデックスnextAvailableDirectoryEntryは、タイル・ディレクトリ２１１の最初のエントリを指している。有効タイル・データ・オフセットＦＩＦＯ２３３は、初期化され、有効タイル・データ・バッファ２１２中のタイル・データの各ブロックについてのエントリを含んでいる。走査を開始したとき、有効タイル・インデックスＮの新しいテーブル中の各エントリは試行され、その位置に関するタイル・インデックスが４分木のタイル・インデックス２１４経由でアクセスされる。もし、４分木のタイル・インデックスがこのタイルを表すのに新しいエントリが必要とするならば、図１６を参照しながら上述したように、新しいエントリが生成される。もし、インデックス・テーブル中のエントリが現在はnilである木の葉ノードを参照するならば、これは、走査経路が新しいタイルに突入したことを示している。この事象では、問題の葉ノードは、nextAvailableDirectoryEntryによって指されている位置のタイル・ディレクトリ２１１の新しいエントリを参照しなければならない。次いで、nextAvailableDirectoryEntryは増分される。この新しいディレクトリ・エントリは、未だ圧縮されていないことを示すための印

が付されるとともに、有効タイル・データ・オフセットＦＩＦＯ２３３から抽出された有効タイル・データ・オフセット値をオフセット経由で指すようになっている。有効タイル・データ・バッファ２１２中のタイル・イメージに関連するデータは、背景値に初期化される。この背景値は、グレイ・スケールとして利用可能な最も明るい８ビット白強度が選ばれる。

【０１２６】

このステップで、最後の走査作業で有効であるが現在はもはや有効でないタイルがどれかが決定される（この明細書で使用されるケースでは、最後の３３増分である）。このようなタイルのインデックスは、この方法の好ましいバージョンでは、完成タイル・インデックスＦＩＦＯ２３４に付加されている。この方法は、以前の３３増分のチャンクについての有効タイル・インデックスのテーブルを保存するために、タイル・マネージャ２３１中に余分な記憶部を必要とする。走査の開始時点では、このテーブルは空である。「古くさい」タイル・インデックスのテーブル中の各エントリは、増分保持バツファ２３２にある現在の３３増分のために新しく獲得された有効テーブルのインデックスと比較される。新しいテーブル中にも存在しないような古いテーブル中のエントリは、タイル・コンプレッサ２３８によって圧縮すべきことを示すために、完成タイル・インデックスＦＩＦＯ２３４に加えられる。

【０１２７】

次いで、タイル・コンプレッサ２３８は、完成タイル・インデックスＦＩＦＯ２３４からタイル・インデックスを取り出し、タイルを圧縮する。圧縮は、タイルによって消費されるメモリ容量を削減するための適切な処理形態であれば、どのようなものでもよい。保持することが所望の品質や形式の表示を達成するためには必要でないと判断されるデータの要素であれば、圧縮によって喪失されてもよい。圧縮は、表示、及び／又は、空間的若しくはグレイスケール解像度の変化を含んでもよい。１つの適切な当業界で周知の手法は、しきい値手法の適用の結果である２値化である（このような手法は、ウイリアム・Ｋ・プラット著"Digital Image PRocessing（デジタル・イメージ処理）"ニューヨーク、ジョン・ウィレイ＆サンズ社版、１９９１年、第５９７－６００ページで議論されている。）。グレイスケールや２値データのための多くの標準的な圧縮手法のいずれであっても、本実施例に適用可能である。２値データのためのグループ４ファックスの圧縮、あるいはグレイスケールのためのＬＺＷを適用することもできる。

【０１２８】

圧縮ステップの間、有効タイル・データ・バッファ２１２から取り出された原タイルは、そのままである。圧縮された（若しくは変化した）形態のタイルの新しいコピーが、圧縮タイル・データ・バッファ２１３内のnextAvailableCompressedData（次の利用可能な圧縮データ）ポインタで示された位置に作成される。このバッファは、タイルが圧縮のために印を付された順番に満たされる。今圧縮されているタイル（タイル・インデックス２３５から得られる）に関連するタイル・ディレクトリ２１１のエントリは、次いで、圧縮タイル・データ・バッファ２１３内の適当な位置を指すように更新され、長さフィールドと圧縮状態が補正される。次いで、今圧縮され手いるタイルの有効バージョンに関連する有効タイル・データ・バッファ２１２内のオフセットは、有効タイル・データ・オフセットＦＩＦＯ２３３の後ろに戻されて、さらに有効なタイルを表すのに必要となったときに使用される。最後に、nectAvailableCompressedDataポインタは、圧縮されたタイルのバイト長に従って増分される。一般に、各タイルは、タイルの内容と採用された圧縮方法に依存して、区々のバイト数に圧縮される。

【０１２９】

上述したように、タイル・キャッシュ２３６は、修正処理におけるコンピュータ処理の効率化のために設けられている。図２０は、タイル・マネージャ２３１によって構築されたタイル・キャッシュを示している。位置（ｘ，ｙ）で示された特定の画素をアドレスするために（但し、ｙとｘは各々１６ビットの行と列の位置のことである）、各座標はまずインデックス成分と「タイル内」位置成分に分解される。これらは、各座標の夫々上位１０

ビットと下位６ビットによって与えられる。タイル・キャッシュを用いて有効タイル・データ・メモリ・バッファ２１２内のタイル・アレイ・データにアクセスするために、各１０ビットのｙとｘのタイル・インデックスを使用する方法が、図２０に示されている。ｙインデックス・オフセットは、現在有効なタイルの中の最小のｙインデックスに対応するが、考慮中の座標のｙインデックスから差し引かれて、ｘインデックス・オフセット・テーブル２４１とタイル・データ・オフセット・テーブル２４２の両方にアクセスするために使用される局所ｙインデックスを与える。ｘインデックス・オフセット・テーブル２４１は、ｘタイル・オフセットの線形テーブルであり、ｙオフセット行の後のタイル・インデックスの各行（すなわちｙタイル・インデックス値）についての最小ｘインデックスを含み、スキャナ長で決定される最大行に至る（ここで言う「最大行」は、ｙインデックス・オフセット＋（Ｌ／６４）＋２であり、付加された２つの因子は、割り算と、後項の部分的に満たされたタイルの商と切り捨てに再び依拠する）。タイル・データ・オフセット・テーブル２４２は、３重アレイであり、１６ビットを含む各要素は有効タイル・メモリ・バッファ２１２をインデックスしている。ｘインデックス・オフセット・テーブル２４１のエントリは、局所ｙインデックスを加えて示されるケースでは、テーブル・ベース・アドレスに与えられる（算術はｘインデックス・オフセット・テーブル・セル・ユニットで実行されると仮定する。この実施例では１６ビットである）。同様に、タイル・データ・オフセット・テーブル２４２中で関連する３重テーブルは、局所ｙインデックスの３倍をタイル・データ・オフセット・テーブルのベース・アドレスに加えることによって与えられる。さらに、関連する行についての局所ｘオフセットを加えることによって、関連するタイル・インデックスをアドレスするようにタイル・データ・オフセット・テーブル２４２へのインデックスは変更される。これは、原ｘタイル・インデックスと、ｘインデックス・オフセット・テーブル２４１からの行についてのエントリの間の相違によって与えられる。

【０１３０】

タイル・データ・オフセット・テーブル２４２中の関連するセルの内容は、有効タイル・データ・メモリ・バッファ２１２内のタイルを表すイメージ・データのアレイのベース・アドレスへのポインタを提供する。各座標の下位６ビットは、タイル内のアドレスを通常通りに形成するために使用される。タイル・データ・オフセット・テーブ２４２ル中の全てのセルが有効である訳ではない。テーブルは、最悪のケース・シナリオにも充分なほど大きく構築され、タイル・データ・オフセット・テーブル２４２によってインデックスできない僅かなタイルが現在有効であることもしばしばある。

【０１３１】

図１７を参照しながら上述した結合タイリングや修正方法に用いるのに適切な修正処理について、以下で議論する。国際特許出願公開ＷＯ９６／２７２５７号や米国特許第５５７８８１３号で議論しているように、修正を達成するためにブレゼンハム・アルゴリズム手法を採用することも可能であろう。しかしながら、欧州特許出願第９７３０２５１９．０号で議論されているようなより洗練された手法が好ましい。該欧州特許出願の内容は、本特許出願に組み込まれる。

【０１３２】

図２１は、タイルのアレイで形成された座標フレームを示している。同図では、最終的な直線で囲まれたイメージを形成するように、増分８８のストリームがマッピングされている。国際特許出願公開ＷＯ９６／２７２５７号や米国特許第５５７８８１３号に係る方法のように、直線状の所定の寸法のイメージ・バッファは存在しない。むしろ、所定のスケールであるが未定の全体寸法からなる直線状のイメージ空間というべきである。各増分に関連する位置タグのＸ及びＹ座標は、直線状のイメージ空間の水平及び垂直寸法（列及び行インデックス）に対応する。増分のストリームに関連する線形イメージ・センサの各端点についての位置タグ（右手に、一片が拡大して示されている。）の軌跡も示されている。これら増分の対（増分＃１及び増分＃２）の中で、直線によって位置タグを結ぶことによって強調表示されている。これらは、走査が折り返して、重なり合った領域内で殆ど交

差するものとして、選択されている。

【０１３３】

直線状のイメージ・バッファの解像度は、スキャナから要求される出力解像度（一般には２００又は３００ｄｐｉ）によって決定されるが、データが計測された端点位置に応じて異なる可能性がある。これは、ナビゲーション・センサ２４の解像度（例えば、ナビゲーションの間に捉えられる紙の繊維などの特徴の空間的な広がりに依存する）によって決定される。このような相違を受容するためには、端点位置データを出力画素解像度にスケールを合わせる必要がある。

【０１３４】

次の処理は、位置タグが付された走査帯中のイメージ増分を、直線状のイメージ空間にマッピングすることである。この処理では充分なイメージ品質を維持することがとりわけ重要である。１つの手法は、各増分に含まれる要素を、直線が端点位置の間に引かれたイメージ・アレイの画素に単にマッピングすることである。マッピングは、各イメージ画素の線形センサ素子に最も近くても、あるいは、センサ素子の間にの補間を含んでいてもよい。この実施例で適用できる適切な線形若しくは立方補間方法は、ウォルバーグ著"Digital Image Warping"第１２７－１３１ページ、ＩＥＥＥコンピュータ学会出版、ロス・アラミトス，ＣＡ，１９９２に記述されている。

【０１３５】

このような方法で修正を実現すると各増分が特異なインパクトを持つから、サンプリング人工物を必然的に招来する。さらに、位置タグ・イメージ・データの原走査帯が直線状の格子に対して厳密に過度にサンプリングされていることが実際上必要であり、さもなければ、直線で囲まれたイメージの中に画素落ちが発生するであろう。１つの解決方法が図２２に図解されている。この図において、１対の増分Ｉ１およびＩ２の端点位置が領域を定義している。これらの増分は、線形センサが読み取った連続するペア（対）であってもよい。連続的に読み取ったものを使用することをここではペア（ＰＡＩＲＷＩＳＥ）手法と呼ぶことにする。あるいは、増分Ｉ１とＩ２は、大きなグループのセンサ増分の境界対であってもよい。このような増分を使用することを、ここではチャンク（ＣＨＵＮＫＷＩＳＥ）手法と呼ぶことにする。ペア手法と比較した場合、チャンク手法は、一群の増分を同時に扱うことから、イメージ品質をわずかに犠牲にするだけで、計算上の利点がある。

【０１３６】

タイルされた直線状のイメージ空間内の画素位置は、整数位置である。増分の対によって定義された領域内のこのような各画素毎に、強度値は、その周辺の領域をマップする線形センサ画素の数を積分することによって計算される。特定の好ましい実施例では、画素位置に及ぶ連続する増分から取られた特定の４つの線形センサ画素を積分するために、種々の形態の２直線補間が用いられる。ここで言う４つの線形センサ画素は、直線で囲まれたアレイ画素の１つの側である線にマップする増分中の２個のセンサ画素と、さらに直線で囲まれたアレイ画素の対向する側の線にマップする隣接する増分中の２個のセンサ画素のことである。

【０１３７】

図２２には、増分Ｉ１及びＩ２の対が走査イメージを定義する直線状の空間の一部分にマッピングされている様子が示されている。簡素化のため、端点位置Ｐ１，Ｐ２及びＰ１'，Ｐ２'は、夫々、増分Ｉ１及びＩ２についての最初と最後のセンサ素子の中心に対応するようになっている。Ａ１，Ａ２及びＢ１，Ｂ２のような中間のセンサ素子も示されている。直線で囲まれたイメージ・アレイ中の画素位置は、端点位置ジオメトリに関する整数位置に対応するように選ばれている。出力直線イメージの解像度は、線形イメージ・センサのそれとは同じではないかもしれない。しかしながら、実際の解像度に拘わりなく、Ｎ個のセンサ画素は、各増分の端点と交わる線上にマップする。説明の簡単化のため、図２２ではＮ＝１８としている。各線形センサは丁度８個の素子を持ち、各増分の長さは、直線で囲まれた画素の解像度単位で計測されるように、線形センサの画素ピッチのＮ－１＝７倍である。

【０１３８】
増分の境界の対で定義された領域内に存在する特定の画素は、２つの増分からなる直線で囲まれた各イメージ行の交差のｘ成分によって与えられる。１つの例は、図２２における、直線で囲まれたイメージ行Ｄ上の画素Ｃ１とＣ２である。簡単なジオメトリから、もし増分Ｉ１が端点座標（ｘ１，ｙ１）と（ｘ２，ｙ２）を持っていれば、行Ｄとの交差のｘ座標は整数値ｙＤを持っているが、これは以下の式のように表される。
【０１３９】
【数１１】
xD=(yD-y1)*(x2-x1)/(y2-y1)
【０１４０】
次いで、行Ｄ＋１との交差は下式で表される。
【０１４１】
【数１２】
x(D+1)=xD+(x2-x1)/(y2-y1)=xD+xInc
【０１４２】
ここで、ｘＩｎｃとは各連続的な行の一定の増分である。同様に、もし増分Ｉ２が端点座標（ｘ１', ｙ１'）と（ｘ２', ｙ２'）を持っていれば、行Ｄとの交差のｘ座標は、整数値ｙＤを持っているが、以下の式で表される。
【０１４３】
【数１３】
xD'=(yD-y1')*(x2'-x1')/(y2'-y1')
【０１４４】
次いで、行Ｄ＋１との交差は下式で表される。
【０１４５】
【数１４】
x(D+1)'=xD'+(x2'-x1')/(y2'-y1')=xD'+xInc'
【０１４６】
この式によれば、各行で交差する範囲を順に計算するためのコンピュータ処理上、効率的である。現在の行との交差は、単に以前の行との交差を増分するだけで計算されるからである。
【０１４７】
まず、領域内にある行の範囲を特定しなければならない。これは、各増分に共通な整数ｙの値の範囲によって与えられる。これらは、下式の範囲の整数値である。すなわち、
【０１４８】
【数１５】
[Ceiling(Max(y1,y1')),Floor(Min(y2,y2'))]=[Y1,Y2]
【０１４９】
但し、スキャナの向きは大凡紙に対して垂直を維持するとする。この手法によれば、スキャナを紙に対して任意の方向に走査させることができる。例えば、もしスキャナが転回されたならば、線形センサ・データは逆向きにされ、端点位置データの極性は逆転される。さらに、もし、スキャナの角度が垂直から４５％以上であれば、ｘ及びｙ位置座標の極性と、イメージの行及び列の双方を反転させることによって、正しい修正が達成される。説明の簡単化のため、この実施例の説明では、ほぼ垂直な場合しか扱っていない。しかしながら、上述したもっと一般的なケースに拡張することは、当業者にとっては容易に想到し得る事柄である。
【０１５０】
領域中の有効な画素は、以下に示す擬似コードに従って同定される。
【０１５１】
【表１】

```
TRANSFORM LOOP1
{
  //INITIALISATION
  y=Y1
  xInc=(x2-x1)/(y2-y1)
  xInc'=(x2'-x1')/(y2'-y1')
  xD=x1+(y-y1)*Xinc
  xD'=x1'+(y-y1')*xInc'

  //MAIN LOOP
  while (y<=Y2)
  {
    [X1,X2]}={Ceiling(xD),Floor(xD')

    x=X1
    //INNER LOOP
    while (x<=X2)
          VISIT PIXEL (x,y)
    xD+=xInc
    xD'+=xInc'
    y++;
  }
}
```

【０１５２】
ここで、フロア（ｆｌｏｏｒ）及びシーリング（ｃｅｉｌｉｎｇ）という演算子は、通常の算術的な解釈を持っている。初期化（ｉｎｉｔｉａｌｉｓａｔｉｏｎ）と内部ループ（ｉｎｎｅｒ　ｌｏｏｐ）の実際の詳細は、以下に記述する異なる方法に特化されている。ここで示されたＴＲＡＮＳＦＯＲＭ　ＬＯＯＰ（変形ループ）の実施例の説明を簡素化するために、ｘＤは常にｘＤ'よりも小さいという仮定が作られている。各行においてｘＤとｘＤ'の順序を試行することによってこのことを一般的なケースに拡張することは容易である。
【０１５３】
次のステップは、線形センサ画素が直線で囲まれた各格子画素にどの程度寄与しているかを決定することである。ここでは、ペア手法に係る実施例について説明する。
【０１５４】
ペア手法を用いた本発明に係る実施例では、修正は、コンピュータ・グラフィックスのテクスチャ・マッピング（例えば上述のウォルバーグの第１８８ページを参照されたい）で用いられるような逆マッピングに基づいて行われる。逆マッピングによれば、増分Ｉ１とＩ２の端点位置を結ぶ２つの線で仕切られた間にある各画素は、増分の対自身の間で定義

された座標空間内のサブ画素位置に逆方向でマッピングされる。図２３には、この状態を図解している。すなわち、図２２では直線で囲まれたイメージにマッピングされるように示された増分Ｉ１及びＩ２の対は、単純な増分空間をも定義し、そこでは単位は線形センサ画素間隔であり、増分はｙ軸と整列し、単位画素だけ離れて配置される。この空間内では、左手の増分の先頭のセンサ素子は座標（０，０）をもつ。

【０１５５】

直線イメージからの画素Ｃ１は、図中では、増分Ｉ１とＩ２の間にマッピングされている。該画素は、サブ画素座標（ｘ，Ｙ．ｙ）を持つ。但し、Ｙは増分の座標値の整数成分であり、ｘとｙはサブ画素オフセットである。夫々の画素値は、周囲の４つの画素について２直線補間を用いることで決定される。すなわち、

【０１５６】

【数１６】

$$BiLinear(I1,I2,Y,y,x)=I1[Y]*(1-x)*(1-y)+I2[Y]*(1-y)*x+I1[Y+1]*(1-x)*y+I2[Y+1]*x*y$$

【０１５７】

直線イメージから増分間の空間に変換するための最良の近似値は、各画素についての２次解法を包含している。これは、コンピュータ処理に含まれる費用のために、現実的な解法としては満足できない。直線イメージの座標から増分の対で定義された座標空間に同質の近似的な線形マッピングを引き出すことは可能であるが、同質でない解法の方が好ましい。このような同質でない解法については、局所的な変換に関する性質についての仮定は全く必要でない。直線状のイメージ空間内で隣接する増分の対の投影がクロス・オーバーを含むような稀なケースを容易に解くことは可能である。

【０１５８】

このような同質でない方法では、［Ｙ１，Ｙ２］の間の直線で囲まれたイメージの各行毎に異なるマッピングが定義されている。これは、端点位置と連結する線を持った行の交差によって再び決定される。このような場合、線上の交差点の距離が回復される。これは、解像度の相違に必要なスケーリングを行った後、線形センサ上の物理位置に対応する。図２３では、ａＤとａＤ'は夫々増分Ｉ１とＩ２上の交差であり、その簡単なジオメトリを以下に示しておく。

【０１５９】

【数１７】

$$aD=(yD-y1)*(N-1)/(y2-y1)$$
$$aD'=(yD-y1')*(N-1)/(y2'-y1')$$

【０１６０】

これに続く行Ｄ＋１の交差は下式の通りとなる。

【０１６１】

【数１８】

$$a(D+1)=aD+(N-1)/(y2-y1)=aD+aRowInc$$
$$a(D+1)'=aD'+(N-1)/(y2'-y1')=aD+aRowInc'$$

【０１６２】

効率的な直列実装のために連続する各行の簡単な再帰的関係が再び与えられる訳である。

【０１６３】

行Ｄ上の直線イメージ中の［Ｘ１，Ｘ２］間の各画素は、２つの各交差点（０，ａＤ）及び（１，ａＤ'）を結ぶ増分空間中の線上の異なる点にマッピングされる。この行上の線形マッピングを仮定すると、直線イメージ中の位置（ｘＥ，ｙＤ）における画素は、増分空間中の位置（ａＥ，ｂＥ）にマッピングされるであろう。ここで、

【０１６４】

【数１９】

$$aE=aD+(xE-xD)*(aD'-aD)/(xD'-xD)$$

bE=(xE-xD)/(xD'-xD)

【０１６５】

である。また、行Ｄ上の次の画素については下式が成立する。

【０１６６】

【数２０】

a(E+1)=aE+(aD'-aD)/(xD'-xD)=aE+aInc

b(E+1)=bE+1/(xD'-xD)=bE+bInc

【０１６７】

ここで、（ａＥ，ｂＥ）における強度は双線形補間によって決定される。

【０１６８】

ＴＲＡＮＳＦＯＲＭ　ＬＯＯＰについての付加的な初期化及び修正された内部ループは、以下の擬似コードで表される。

【０１６９】

【表２】

```
TRANSFORM LOOP2
{
    //INITIALISATION
    y=Y1
    xInc=(x2-x1)/(y2-y1)
    xInc'=(x2'-x1')/(y2'-y1')
    xD=x1+(y-y1)*xInc
    xD'=x1'+(y-y1')*xInc'

    aRowInc=(N-1)/(y2-y1)
    aRowInc'=(N-1)/(y2'-y1')
    aD=(y-y1)*aRowInc
    aD'=(y-y1')*aRowInc'

    //MAINLOOP
    while(y<=Y2)
    {
        [X1,X2]=[Ceiling(xD),Floor(xD')]

        x=X1
        aInc=(aD'-aD)/(xD'-xD)
        aInc=1/(xD'-xD)
        a=(x1-xD)*aInc
        b=(x1-xD)*bInc
        //INNER LOOP
        while(x<=X2)
        {
            A=Floor(a)
            pixel[y][x]=BiLInear(I1,I2,A,a-A,b)
```

```
            a+=aInc
            b+=bInc
        }
        xD+=xInc
        xD'+=xInc'
        aD+=aRowInc
        aD'+=aRowInc'
        y++;
    }
}
```

【０１７０】
対方式の直線画素補間を実現するための本発明に係る方法の他の実施例では、さらに直接的なジオメトリ補間を用いることもできる。このような方法は、直線イメージ画素を増分座標空間に明確にマッピングし直す必要がない。

【０１７１】
コンピュータ処理の必要性が少ないような逆投影手法に関する特定の簡単な実施例では、各増分について補間を行い、次いで、直線イメージ中でこれらの値を直接補間する。直線で囲まれた各イメージ画素を各増分上の最も近い位置にマッピングすることによって、コンピュータ処理のコストは増加するが、よりよいイメージ品質を達成することができる。これらは、直線で囲まれた画素の強度を最も反映するような（線形センサに沿って）補間された強度を持つ増分上の点である。補間された増分強度値の対は、組み合わせられ、線形補間のために再度用いられ、その結果、画素の強度を最も良く表すことができる。このような実施例は、欧州特許出願第９７３０２５１９．０号に記述されている。

【０１７２】
次いで、直線マッピングのチャンク手法を採用した修正手法についての実施例をさらに説明する。チャンク手法は、ＴＲＡＮＳＦＯＲＭ　ＬＯＯＰの最適なバージョンの外部ループは滅多に計算されないということと、多数の出力が素が順次アクセスされるのでメモリ帯域幅を直接的（キャッシュなしに）に向上させることができる、という２つの利点を備えている。

【０１７３】
図２４には、Ｍ個、この場合は４個、の増分からなるチャンクを示している。外側の一対の増分は、Ｉ１及びＩ２が付されているが、チャンク内でマップする有効な画素を特定するために、ＴＲＡＮＳＦＯＲＭ　ＬＯＯＰの外部ループによって再び使用される。好ましい実施例では、対に関するケースについて既述されたように、同質でない逆投影アルゴリズムの拡張形式を持つ直線的なアレイ画素値を決定する。図２５には、図２４でラベルＤが付された行をＩ１及びＩ２で仕切られた４個の増分を含む増分空間に変換する様子が示されている。該変換は、行Ｄの交差点を外側の増分の各々と結び付けることである。図２３に示すような対に関するケースに続いて、これら交差は、Ｉ１及びＩ２について、夫々、線形センサのオフセット間隔ａＤ及びａＤ'を有している。増分空間における交差点の座標は、したがって、（０，ａＤ）と（（Ｍ－１），ａＤ'）である。

【０１７４】
行Ｄ上の間隔［Ｘ１，Ｘ２］の間の直線画素は、増分空間内の２つの交差点（０，ａＤ）及び（（Ｍ－１），ａＤ'）を結ぶ直線上の異なる点をマップする。この行に沿った線形マッピングを仮定すると、直線イメージ中の位置（ｘＥ，ｙＤ）における画素は、増分空間内の位置（ａＥ，ｂＥ）にマッピングするであろう。ここで、ａＥとｂＥは下式で表さ

れる。
【０１７５】
【数２１】
aE=aD+(xE-xD)*(aD'-aD)/(xD'-xD)
【０１７６】
【数２２】
bE=(M-1)*(xE-xD)/(xD'-xD)
【０１７７】
そして、行Ｄ上の次の画素については下式が成り立つ。
【０１７８】
【数２３】
a(E+1)=aE+(aD'-aD)/(xD'-xD)=aE+aInc
【０１７９】
【数２４】
b(E+1)=bE+(M-1)/(xD'-xD)=bE+bInc
【０１８０】
ここで、点（ａＥ，ｂＥ）における強度は、２次補間によって決定される。
【０１８１】
この場合、ＴＲＡＮＳＦＯＲＭ　ＬＯＯＰは以下の擬似コードの通りとなる。
【０１８２】
【表３】

```
TRANSFORM LOOP3
{
    //INITIALISATION
    y=Y1
    xInc=(x2-x1)/(y2-y1)
    xInc'=(x2'-x1')/(y2'-y1')
    xD=x1+(y-y1)*xInc
    xD'=x1'+(y-y1')*xInc'

    aRowInc=(N-1)/(y2-y1)
    aRowInc'=(N-1)/(y2'-y1')
    aD=(y-y1)*aRowInc
    aD'=(y-y1')*aRowInc'

    //MAIN LOOP
    while(Y<=Y2)
    {
        [X1,X2]=[Ceiling(xD),Floor(xD')]

        x=X1
        aInc=(aD'-aD)/(xD'-xD)
        bInc=(M-1)/(xD'-xD)
        a=(x1-xD)*aInc
        b=(x1-xD)*bInc
        //INNER LOOP
        while(x<=X2)
        {
            A=Floor(a)
            B=Floor(b)
```

```
            pixel[y][x]=BiLinear(IB,IB+1,A,a-A,b-B)
            a+=aInc
            b+=bInc
        }

        xD+=xInc
        xD'+=xInc'
        aD+=aRowInc
        aD'=aRowInc'
        y++;
    }
}
```

【０１８３】
ここで、ＩＢはチャンク中のＢ'番目の増分のことであり、ＩＢ＋１は、チャンク中の（Ｂ＋１）'番目の増分である。
【０１８４】
チャンク手法は、走査経路は殆ど均一であり滑らかに変化するだけという事実に依拠している。このことは、空間的にも（走査は均一な形状のはずである）、走査経路に沿ったサンプリング周波数の点からも（走査は均一にサンプリングされるべきである）、真実に相違ない。チャンク手法は、特に、図１７で図解されたタイリング方法によく適合する。もし増分保持バッファ２３２が（Ｓ＋１）増分を保持するために適用されたならば、（Ｔ＋１）増分を持った均一なチャンクを採用することができる。ここで、ＴはＳの整数因子である。例えば、もし３３増分が増分保持バッファ２３２に保持されているならば、均一な修正チャンクをいかなる２（ペア手法の限界），３，５，９，１７，若しくは３３個の増分からでも形成することができるであろう。修正ステップは、それゆえ、増分保持バッファ２３２の連続的な各補充の間にＳ／Ｔ回だけ発生する。
【０１８５】
走査経路の均質さの度合いに依存してチャンクのサイズを動的に変化させることによって、チャンク手法を拡張することもできる。均質性がよい場合には、イメージの歪なしに大きなチャンクを受容することができる。しかし、走査経路若しくはサンプリング方式が急速に変動する場合には、人工物の生成なしには、小さなチャンクサイズのみしか受容できない。限界的なケースでは、実施例はペア手法に逆戻りする。
【０１８６】
既に議論されたように、メモリに使用可能な直線イメージ空間は、線形センサの解像度（一般には２００若しくは３００ｄｐｉ）で、Ａ４サイズの単一のレターを容易に受容することができる。直線状のイメージ空間には寸法上の制限は全くないので、最終的なイメージは、走査経路の初期開始位置に拘わらず、良く形成されるであろう。適切に整列するために、走査は仮定された方向（例えば、常に紙の側と平行であるとか）で始まるか、若しくは、方向は走査の内容によって修復されて最終イメージを再方向付けするために使用されなければならない。ページ上の主方向を自動的に決定する方法は、後述で使用されているが、例えばＨ．Ｓ．バイアド著"The Skew Angle of Printed Docunet（印刷文書のスキュー角）"（Proc 4th SPSE Conference Symposium on Hybrid Image Systems, Rochester, Newyork 1987）のような文献で知られている。
【０１８７】

次のステップは、重なり合った連続するイメージ帯を縫い合わせることである。この目的は、最終的に修正され復元されたイメージを結果としてタイルされた形式で格納するように、イメージを複数の帯を組み合わせることである。イメージを圧縮すれば、なお有効であろう。これには、蓄積されたナビゲーションエラーの殆どを同定し矯正するとともに、残りの全てのエラーをマスクしなければならない。

【０１８８】

ここで説明される実施例では、ナビゲーション・データのストリームは、縫合に必要な位置合わせ情報を供給するようになっている。ナビゲーション信号はエラーを蓄積する傾向があるので、これは、特徴オフセットの解析によって導かれる修正信号をフィードバックすることによって継続的に補正される。

【０１８９】

しかしながら、まず、ナビゲーションエラーのないと仮定したとして、イメージ帯の縫合方法について説明することにする。

【０１９０】

重なり合っている領域の中には、２つのイメージ帯を縫い合わせる為に必要なものがある。帯は、オリジナルの丁度走査された領域上を走査装置の経路が逆行することによって仕切られる。帯は、スキャナが走査されるオリジナル上を打つ間に捕捉されたイメージ・データで構成される。後続の説明では、「帯」という言葉は、このようなデータをマッピングすることで形成される復元イメージの一部を呼ぶ場合にも用いられる。

【０１９１】

図２１中の増分の端点位置の軌跡によって表される走査によって２つの重なり合う帯が生成される。図２６では、帯＃１で構成される直線状のイメージ空間にマッピングされた箇所の一部は、その帰路で、重なり合う領域１０２を満たしながら、帯＃２に対応する走査部分に再マッピングされる。時刻Ｔでは、部分的な帯は、このようにしてかなり走査される。このようなケースでは、帯＃２を帯＃１の先端の直線状のイメージ空間にマッピングすることを単に続けるだけで、満足のいく縫合が実現される。図２１に戻って、増分＃１の場合と丁度同じ方法で全ての長さに沿って増分＃２をマッピングすることができる。また、図２６の重なり合った領域１０２中の各画素についても同様である。帯＃１から帰結される直線状のイメージ空間における最終的な変換は、単に帯＃２から得られる結果と置き換えられる。

【０１９２】

この事象では、図１７に関して言及された方法を、もはや有効でなく圧縮されたタイルの内容が置換されるように修正する必要がある。このことは、あたかも各タイルによって要求されるメモリに従って、コンプレッサ２３８によって圧縮されたタイルにメモリが割り当てらるかのように、圧縮されたタイルを記憶するために付加的メモリを必要とすることになりそうである。もしタイルの第２バージョンが圧縮後に第１バージョンよりも多くのメモリを必要とするならば、困難を生じる。第１の帯に部分的にのみ満たされており、非有効（線形センサの端縁のタイル）とされるタイルの場合にも、困難を生じる。もしこれらが第１の経路で圧縮されているならば、第２の「満杯」のバージョンは、明らかにより多くのメモリを消費するであろう。可能な手法は、満たされないタイルを決して圧縮しないことである。タイルの各コーナーを評価して、情報が既に圧縮ステップでこれに書き込まれたか否かを判断することによって、部分的に満たされたタイルをチェックすることができる。部分的に満たされたタイルは、それゆえ、非有効になったとき（完全なイメージを取得するための帯の重なり合いに依存する）に削除されるか、若しくは別個のメモリ領域に格納される。

【０１９３】

帯＃２の重なり合わない（すなわち帯＃１によってマッピングされていない）部分によってマッピングされた全ての画素が適当にマッピングされていると仮定すれば、帯＃１の満たされたタイルとしてマッピングされた重なり合った領域１０２における画素を帯＃２の画素によって置き換えないようにする手法は、タイリング処理にはあまり困難を与えない

。すなわち、帯＃１によってマッピングされた領域は、帯＃２によってマッピングされた領域を切り取る訳である。原理上は、帯＃１と帯＃２が重なり合った領域の外の全ての画素を正しくマッピングしていると仮定すれば、帯＃１又は帯＃２、あるいはこれらの組合せのいずれからも、重なり合った領域内の画素を等しく取得することができる。

【０１９４】

実際には、帯＃１と帯＃２の位置タグ間での蓄積された誤差のため、ナビゲーション誤差がないと仮定したこの簡素な手法は、あまり良い結果をもたらさない。

【０１９５】

本発明に従って縫合する手法の１つの実施例について、図２７及び図２８を参照しながら説明することにする。図２７には、処理ステップとデータ・バッファを示し、他方、図２８は、帯＃１と帯＃２に関して利用される処理に関係している。前述したように、イメージ増分データは、タイルされた直線状のイメージ空間にマッピングされている。図２８は、戻り経路である帯＃２についての直線状のイメージ空間上にマッピングするときの、帯＃１内に捕捉されたイメージの使用について示している。ナビゲーションの補正は、帯＃１と帯＃２の間の重なり合った領域内の特徴の相関をとることによって計算される。

【０１９６】

図２８では、２つの部分１０４と１０５に分割された重なり合った領域を強調表示している。図２８に示すように、帯＃１を収集する間、四辺形状のイメージ・セグメント１３０（以下、「位置決めタイル」と呼ぶことにする）は、重なり合った領域１０５内の帯の低い方の端縁に沿って周期的に分類されている。後方の経路（帯＃２）上では、領域１０５上の帯＃２の重なり合った領域１０４は、帯＃１の位置決めタイル１３０を含んでいるが、これらの帯＃１によって供給された満たされたタイルの上書きを許さないことによって、切り取りられる（すなわち削除される）。帯＃１の位置決めタイル１３０は、重なり合った領域１０４が切り取られた後は、帯＃２に残った部分の先端に配置されることになる。もしナビゲーション・データが完全であれば、位置決めタイル１３０の位置と帯＃２中の該タイルを再度走査したイメージの位置の間のオフセットは全くないであろう。さらに現実的には、ナビゲーション誤差の中には、最後の位置決めが実行されて以来蓄積されていくものもある。２つの帯中のこのタイルの外観の間のオフセットは、イメージ・データに関連するナビゲーション位置タグを将来更新するために用いられる矯正因子を生成する。これによって、蓄積される総誤差を最小限にすることかできる。このようにして、帯同士が重なりあった領域中に明らかな歪みを招来する程にナビゲーション・データ中に蓄積される総誤差が大きく成長することが、回避される。

【０１９７】

帯＃１と帯＃２の縫合を、増分のストリーム中の帯同士が最接近している点から開始することによって、ナビゲーション誤差を最小限にすることができる。これら走査サンプル中で接近している帯の間では、ナビゲーション誤差は最小のはずである。帯に沿って誤差を予測し処理するために縫合された帯の間で検出された誤差を累進的に使用しながら、連続的な帯の間の「ターニング・ポイント」（図１で示された捕捉手法に従って取得され、有益である。）のまわりでマッチングを継続的に行うことによって、ジップ若しくはジッパーの閉鎖と本質的には類似した様式の自己誘導的な縫合制御が可能となる。

【０１９８】

ここでは、図２７と図２８を参照しながら、帯＃１と帯＃２を縫合するための処理ステップについて説明する。図２７は、イメージ増分バッファＢとタイルされた直線状のイメージ空間１００を示している。図２７は、特徴位置バッファ１３１とイメージ・バッファ１３２も示している。このタイプの構成は、国際特許出願公開ＷＯ　９６／２７２５７号に示されている。しかしながら、ここで説明されるタイル化手法では、特徴データ・バッファ１３５がさらに必要である。位置決めタイル１３０が構築されたとき、位置決めタイルを含んだイメージ・タイルのコピーを取得する必要がある。これは、このようなイメージ・タイルが非有効になったときに、これらはコンプレッサ２３８によって圧縮され、これらを含んだデータはもはや即座に縫合に利用することができなくなるからである。したが

って、これらのデータを圧縮していない形式で付加的な特徴タイル・バッファにコピーすることは有益なのである。ここで、特徴位置バッファ１３１からは、付加的な特徴データ・バッファ１３５内の各関連するタイルにポインタが設けられている。処理ステップは以下に示す通りである。

【０１９９】

１．上述したように、帯＃１を収集する間、位置決めタイル（１３０）は、重なり合う領域１０５の低い方の端縁に沿って周期的に分類されている。１つの位置決めタイル全体を上述した相関のために使用することができる。しかしながら、好ましい実施例では、直線状のグレースケール・イメージのタイル（例えば１５×１５画素）からなる高周波数のコントラスト（以下、「位置決め特徴」と呼ぶ）を持つ小さな領域が、帯＃１を捕捉した結果としてタイルされた直線状のイメージ空間１００内に復元されたイメージの一部を形成する位置決めタイルの中で見いだされる。図１７に示されているように圧縮が施されている実施例では、位置決めタイル１３０を含んだこれらのイメージ・タイルは、このようなイメージ・タイルが圧縮される前に、付加的な特徴データ・バッファ１３５にコピーされる。

【０２００】

２．帯＃２を直線状のイメージ空間１００にマッピングするのを始める前に、位置決め特徴の位置タグ（これは、タイルされた直線状のイメージ空間１００内の各位置決め特徴の位置を定義する）が特徴位置バッファ１３１に格納される。

【０２０１】

３．帯＃２がマッピングされると、直線状のイメージ空間が帯＃２によって書かれるのに領域に先立って、直線状のイメージ空間位置決め特徴位置が同定される。これは、直線状の捕捉ウィンドウ１０７を定義することによって実現される。図２８で示されるウィンドウは、イメージ増分に等しい長さと、数画素分の幅を持ち、現在直線状のイメージ空間１００にマッピングされている最中の帯＃２のイメージ増分に先行している。特徴位置バッファ１３１に格納された位置決め特徴位置が捕捉ウィンドウ１０７の中にあるとき、該位置決め特徴位置は選ばれる（一度に唯１つの位置決め特徴しか選ばれない）。

【０２０２】

４．選ばれた位置決め特徴位置は、帯＃２が直線状のイメージ空間にマッピングされるときに、位置決め特徴の予測位置を決定するのに用いられる。以前の位置決め特徴に関して見つかったナビゲーション誤差は、現在の誤差概算（誤差バッファ１３４に格納されている）を用いて現在の位置決め特徴の位置を予測するときに考慮される。したがって、現在の位置決め特徴を見つけるための探索領域は、可能な誤差増分を説明するに充分な程度に大きければよい。

【０２０３】

５．空間イメージ・バッファ１３２は、位置決め特徴位置を格納するのと同様に、選ばれた位置決め特徴のまわりで見つかった直線状の帯＃２のイメージ・データのイメージ・マッピングを、増分バッファＢから一時的に格納するのに用いられる。別言すれば、時刻Ｔにおいて、図２８で示されるイメージ増分１３８は、バッファ１３２中にマッピングされるのである。空間イメージ・バッファ１３２のサイズは、位置決め特徴に加えて必要な探索領域を格納するのに充分でなければならなず、周知の誤差概算アルゴリズムに従って計算されるサイズである。

【０２０４】

６．空間イメージ・バッファ１３２に格納された帯＃２のイメージ・フラグメントは、次いで、関連する位置決めタイル１３０のために特徴位置バッファ１３１によってポイントされている特徴データ・バッファ１３５に格納された対応するイメージ・フラグメントと比較される。このようにして、以前の誤差概算とともに蓄積されたときに、ナビゲーション誤差の概算を更新するようなオフセットが得られる。この更新された概算は、誤差バッファ１３４に位置タグと一緒に格納される。

【０２０５】

誤差概算の位置タグは、直線状のイメージ空間に関する現在位置決め特徴の中心の位置のことである。これは、誤差概算が関係する（すなわち、イメージ位置にマッピングする帯＃２に対応する第１の増分）増分バッファＢ中の増分を決定するのに用いられる。これは、誤差概算を計測する増分が完全に収容されていることを示す。

【０２０６】

他の相関手法を採用することもできるが、２つのイメージ・フラグメント間のオフセットを計算するための受容しうる手法は、「差分の自乗の合計」相関である。小さな探索領域は、特徴の原位置のまわりに決定され、相関係数は以下の方程式で決定される。

【０２０７】

【数２５】

$$C_{k,l}=\Sigma_i\Sigma_j\left(T_{i,j}-I_{i+k,J+l}\right)^2$$

【０２０８】

ここで、$T_{i,j}$は帯＃１の特徴のグレイスケール値であり、$I_{i+k,J+l}$は帯＃２から得られた新しい特徴のグレイスケール値である。インデックスｉとｊは特徴内の画素位置を指定し、ｋとｌは提案される並進オフセット（探索領域にとどまるよう拘束されている）を指定する。相関アレイの結果における最も小さな要素は、２つの特徴間のオフセットである。位置決めタイル内の位置決め特徴は、イメージの変動を最大にするように選ばれる。この結果、相関方法の精度が向上する。この手法の適用例は、国際特許公開ＷＯ　９６／２７２５７号でさらに議論されている。

【０２０９】

第２の処理フェーズでは、誤差を最終的な復元されたイメージから誤差を取り除くために、記録された位置における誤差概算を考慮に入れながら、帯＃２のイメージ・データが直線状のイメージ空間に書き込まれる。これは、個々のイメージ増分の検出データの端点に関連する位置データを修正することによって行われる。この処理は、図１７に示された方法によって、タイリング作業、修正、及び圧縮のときにどのタイルが有効タイルとして必要かを判断する前に完了する点で有利である。

【０２１０】

好ましい手法によれば、縫合が向上する。何故ならば、帯の間の重なり合う領域上を移動するナビゲーション・センサの位置についての誤差が同定され、最終的なイメージを復元する前に矯正されるからである。加えて、イメージ増分の先部と底部の計算された位置間の誤差は、以前の帯を収集する間に蓄積されるが、次の帯の開始時に不本意な人工物を招来することなしに一度に吸収される。このことは、図２８に、上述の第１及び第２の処理フェーズから導き出された帯＃１と帯＃２の間の転換点の右手の端縁間の不連続性によって示されている。

【０２１１】

縫合ステップは、必要とするイメージ捕捉の特性（より具体的で、しかし排他的ではない走査）に関連して異なる方法を用いてもよい。１つの代替案は、ページの上端縁から底縁まで走査の実行を必要とする。このような場合、縫合は、１つの帯の底部と次の帯の上部でのみ行われるだけでよい。他の手法は、ページのどの部分から走査を開始してもよいが、初期走査方向を維持することを必要とする。この場合、帯の両端縁上の位置決め特徴を同定する能力を要するが、一度走査方向が設定されると、現在の帯の１つの側でのみ誤差概算を続けるだけでよい。他の手法では、いずれの方向への走査も許されるし、例えば螺旋状の走査のように走査方向を変えることも許容される。この第３の手法では、帯の両側縁の位置決め特徴を同定する能力を要するのみならず、走査方向が変わる場合には帯の先部と底部の両方について誤差概算を保たなければならない。後者の手法によれば、ユーザには大いなる柔軟性が与えられるが、コンピュータ処理上のオーバーヘッドは高い。

【０２１２】

好ましい実施例では、イメージ復元、縫合、及びイメージ管理を行うための処理電子部品が、図１の走査装置を定義するハウジングの中に含まれている。走査イメージは、瞬時に、イメージ・ディスプレイ１６上に表示されてもよい。走査装置は、位置タグが付された

イメージ・データを格納するためのメモリを含んでもよいが、処理やファイル管理のための電子部品やファームウェアはない。

【０２１３】

図３を参照しながら言及したように、ナビゲーション及びイメージ・センサ２２，２４及び２６は、好ましくは、枢軸部材２０上に搭載されている。１つの実施例では、枢軸部位は、少なくとも１つのエラストマによってハウジングのリマインダに接続されている。エラストマの一端はハウジングの静止部位に接続され、他端は枢軸部材に接続されている。エラストマはヒンジのように働く。したがって、枢軸部位は、摩擦部品を用いることなく「浮遊」状態となる。電力、制御及びデータのための信号は、遮蔽されたフレキシブル・ケーブルを介してセンサに導かれるので、電磁波障害を最小限に抑えることができる。枢軸部材を回動自在に取り付けるために他の方法を用いてもよい。もし枢軸部材を削除して、センサをハウジングに固定したら、イメージ捕捉の間に、走査装置１０が全く傾かないように過度の注意が必要である。この実施例では、光源や光学部品を設計する上で過大な注意を払わなければならない。

【０２１４】

平面的なオリジナルを走査するという一例に従って本発明を説明し図解してきたが、このことは重要ではない。事実、当業者であれば、３次元的なイメージを走査するために数余の技術が利用可能であることを容易に想到し得るであろう。しかしながら、好ましい実施例は、一枚の紙、透明、若しくは写真のような媒体上にイメージが形成されており、且つ、走査装置は媒体に接触するというものである。

【０２１５】

しかしながら、上記で示したように、本発明の他の側面を異なる種類のイメージ捕捉に等しく適用することができる。本発明は、従来のデジタル・カメラのような領域センサによってイメージ・データを捕捉する場合にも等しく適用することができる。デジタル・カメラを用いて大解像度で大きなイメージを捕捉するためには、各々が大きなイメージの一部を捕捉するような一連のイメージ・フレームを撮像し、デジタル・カメラで捕捉された異なるイメージ・データ間で重なり合った領域ができるようにオーバーサンプリングし、これら重なり合った領域を用いて異なる要素間の相対位置を設定しながら走査するのが、効果的な手法である。これら相対位置から、本発明に従った手法によってイメージを復元することができる。分離した位置にセンサを要しない。領域イメージ・センサを持つ位置センサを設けることも当然可能である。この場合には、好ましい実施例に関する本発明の第１の側面を完全に適用することは可能である。しかしながら、捕捉方式によっては、このような位置センサは高価で、不便で、不可能でさえある。

【０２１６】

ここで、相対位置を決定するステップについて説明する。当業者であれば、より複雑な場所をアドレスすることが可能なことを理解できるであろう。この手法の最も簡単な適用例は、隣接するイメージに関する順序で１つのイメージの位置を関係付けるのに幾つかのパラメータしかない単一の大規模な変形が用いられるような場所である。捕捉されたイメージが平面的（例えば平坦な文書やホワイトボード）、若しくは実際上平面的（領域センサの移動に対して特徴が極めて遠いな風景画）であれば、これを適用することができよう。連続する別々のスナップショットからパノラマを捕捉するときにはよくあることであるが、もしセンサの動きがイメージ装置の光学上の中心まわりの回転（若しくは回転に近い）であれば、走査された場面に拘わらず、適用することもできよう。上記したように、捕捉には位置データは全く要らない。どのイメージ・フレームが空間的に隣接している（但し、より拡張的な特徴照合ステップによって計算する必要があれば）かを把握することだけで適当なのである。

【０２１７】

例えば、２Ｄ平面の投影的な変形という技量によって、上述の場所の各々について相対位置を決定することができる。一般的な変形は８個のパラメータを持ち、以下のような式となる。

【０２１８】
【数２６】
$x'=(m_0x+m_1y+m_2)/(m_6x+m_7y+1)$
【０２１９】
【数２７】
$y'=(m_3x+m_4y+m_5)/(m_6x+m_7y+1)$
【０２２０】
ここで、（ｘ，ｙ）は１番目のイメージの座標であり、（ｘ'，ｙ'）は２番目のイメージの座標であり、（m_0，m_1，m_2，m_3，m_4，m_5，m_6，m_7）は変形のパラメータである。
【０２２１】
当業者には理解できるように、この変形を決定するための方法は、イメージ処理の技術文献に豊富にある。上述したように、中心的なアイデアは、イメージの組の間の重なり合った部分を用いて、イメージ間に全体として存在する相対的な変形を決定することである。連続するイメージを単一の世界座標フレームに関して変形して組み立てることができる。リチャード・スツェルスキ著"Video Mosaic for Virtual Environment（仮想環境のためのビデオ・モザイク）"（ＩＥＥＥ、コンピュータ・グラフィックと応用、１９９６年３月、第２２－３０ページ）とそこで引用された参考文献には、この手法の一例が開示されている。
【０２２２】
一度イメージ・フレームが世界座標フレームに関して組み立てられると、上述のタイリング手法を採用したり、図１７で示したように、タイルされたイメージを本質的に達成することが、容易に可能となる。この発明は、例として次の実施態様を含む。
【０２２３】
（１）複数のイメージ要素からイメージを復元する方法であって、各イメージ要素はイメージ・センサで捕捉されたイメージ・データと他のイメージ要素に対する該イメージ・データの位置を示す位置データで構成され、該方法は、イメージを複数のタイルの形式で復元するものであり、各タイルは、イメージの特定の空間領域を表す所定寸法の画素の格子で構成される、ことを特徴とするイメージ処理方法。
【０２２４】
（２）上記（１）の処理方法において、位置データは、ナビゲーション手段によって導き出されてもよい。ナビゲーション手段は、イメージ・センサによって捕捉されたイメージのイメージ要素についての位置データを供給する。
【０２２５】
（３）上記（１）の処理方法において、位置データは、他のイメージ要素に対する各イメージ要素の相対位置をイメージ要素の特徴から決定する先行ステップによって導き出されてもよい。
【０２２６】
（４）上記（１）から（３）のいずれかの処理方法において、前記タイルは、隣接するが重なり合わなくてもよい。
【０２２７】
（５）上記（４）の処理方法において、前記タイルは、２次元の切りばめ細工に適用してもよい。
【０２２８】
（６）上記（１）から（５）のいずれかの処理方法において、センサ読み取りストリームという形式で、イメージを捕捉するようにしてもよい。センサ読み取りの各々は、イメージ・データとこれに対応する位置データで構成される。
【０２２９】
（７）上記（６）の方法は、もしイメージ中の１又はそれ以上の入力センサ読み取りの空間位置に対応するタイルが存在するなら位置データから決定し、１又はそれ以上の新しい

タイルを必要に応じて生成して１又はそれ以上のセンサ読み取りの空間位置に対応するタイルが存在するようにすることによって、タイルを管理するステップと、イメージ・データと位置データに基づいて１又はそれ以上の入力センサ読み取りに対応するタイルの画素に画素値を割り当てることによってタイルに書き込むステップ、という繰返しステップで構成されていてもよい。

【０２３０】

（８）上記（７）の処理方法において、前記１又はそれ以上のセンサ読み取りは、センサ読み取りのグループで構成され、また、タイルを管理しタイルに書き込むという前記ステップは、センサ読み取りの連続的なグループに対して繰り返し実行してもよい。

【０２３１】

（９）上記（８）の処理方法において、タイルを管理するステップは、タイルに書き込むステップを実行する前に、Ｎ個の連続するセンサ読み取りに対して実行してもよい。また、タイルに書き込むステップは、Ｍ個の連続するセンサ読み取りのグループに対して実行してもよい。ここで、ＮやＭは１より大きい正の整数であり、（Ｎ－１）／（Ｍ－１）は正の整数である。タイルに書き込むステップは、さらにタイルを管理するステップの（Ｎ－１）／（Ｍ－１）だけ前に、実行してもよい。

【０２３２】

（１０）上記（１）から（９）のいずれかの処理方法において、イメージの特定の空間領域について生成された各タイルをインデックスするタイル・インデックスを生成するステップを含んでいてもよい。

【０２３３】

（１１）上記（１０）の処理方法において、タイル・インデックスは木として設けられてもよい。この木の各ノードは、ノードの複数の空間解像度における複数の位置を指している。

【０２３４】

（１２）上記（１１）の処理方法において、各ノードは、ノードの２倍の空間解像度を持つ４個の位置を指してもよい。

【０２３５】

（１３）上記（７）または（８）の処理方法において、タイル管理ステップにおいて処理される１又はそれ以上のセンサ読み取りの空間位置に対応するタイルを格納するための有効タイル・バッファを備えていてもよい。

【０２３６】

（１４）上記（７）、（８）または（１３）の処理方法において、タイル管理ステップでは、１又はそれ以上のセンサ読み取りの空間位置に対応しないタイルの幾つか若しくは全てを圧縮してもよい。

【０２３７】

（１５）上記（１４）の処理方法において、タイル管理ステップでは、以前に処理されたグループの空間位置に対応するタイルを、現在のグループの空間位置に対応するタイルと比較してもよい。また、新しく必要となったタイルを生成し、現在のグループには存在しない以前のグループに係るタイルを圧縮してもよい。

【０２３８】

（１６）上記（１３）、（１４）または（１５）の処理方法において、圧縮されたタイルを圧縮タイル格納領域に書き込んでもよい。

【０２３９】

（１７）上記（１３）の処理方法において、タイル書き込みステップにおいて参照するために、有効タイル・インデックスを設けてもよい。

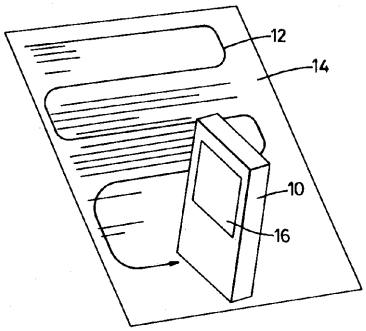
【０２４０】

（１８）上記（１）から（１７）のいずれかの処理方法において、ナビゲーション手段は、捕捉イメージを担持した媒体の固有の構造的特性を検出するための１又はそれ以上のセンサを備えていてもよい。

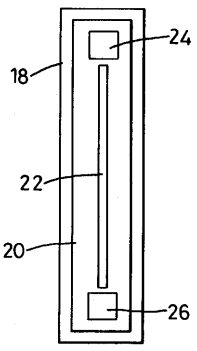
【０２４１】
（１９）上記（１）から（１８）のいずれかの処理方法において、走査装置にセンサが備えられていてもよい。
【０２４２】
（２０）上記（１）から（１９）のいずれかの処理方法は、捕捉イメージ上を非拘束に移動するセンサに対して適用してもよい。
【０２４３】
（２１）上記（１）から（２０）のいずれかの方法を実装する走査装置。
【０２４４】
（２２）上記（１）から（２０）の方法を実施するためコンピュータ・システムに相互接続されるよう設計されたイメージ・データを収集するための走査装置。
【０２４５】
【発明の効果】
以上詳記したように、本発明によれば、非拘束的な走査によって捕捉されたイメージを記憶するために必要なメモリ容量を著しく削減する方法を提供することができる。とりわけ、本発明によれば、コンピュータ上のコスト削減と、任意的に取得された捕捉イメージから復元イメージを形成する速度の増大をもたらす。この方法は、特に、フリーハンド走査やフリーハンド・スキャナに好適である。
【図面の簡単な説明】
【図１】本発明に従った携帯型走査装置が曲がりくねった経路を進む様子を斜視した図。
【図２】図１の走査装置のイメージ・センサとナビゲーション・センサを後方から眺めた図。
【図３】図１の走査装置の、イメージ・センサとナビゲーション・センサが露呈した斜視図。
【図４】図３のナビゲーション・センサの１つに設けられた照明系について模式的に示した側面図。
【図５】図４を参照して説明した照明を供給するための、発光ダイオードや光学部品についての模式的な側面図。
【図６】図１で示した走査装置のイメージ捕捉動作を概念的に示した図。
【図７】図１の走査装置のナビゲーション処理についての１実施形態を示す図。
【図８】図７中の選び出されたステップを模式的に示した図。
【図９】図８の各ステップを実現するための構成要素についてのブロック図。
【図１０】一組のナビゲーション・センサの移動を示した図。
【図１１】図９中のナビゲーション・プロセッサから出力される位置タグの付いたデータ・ストリームの増分を表した図。
【図１２】位置タグが付された複数の増分を格納するためのバッファを表した図。
【図１３】本発明の実施例に従ってタイルする手法を表した図。
【図１４】図１３の手法に従ったタイル・インデックスを空間的に表した図。
【図１５】図１４のタイル・インデックスに対応するインデックス木を示す図。
【図１６】４部分からなるアレイとして表された図１５のインデックス木を示す図。
【図１７】本発明の実施例に従って走査データをタイルし、修正し、圧縮する様子を模式的に表した図。
【図１８】図１７の手法の下で有効タイル・インデックスの構造を示す図。
【図１９】図１７の手法の下で有効タイル・インデックスの構造を示す図。
【図２０】図１５で図解された手法に従ったタイル・キャッシュの構造を示す図。
【図２１】線形イメージ・センサによって抽出された端点の軌跡を示した直線状のイメージ空間を示した図。
【図２２】図２１に示す直線状のイメージ空間の一部分で、２つの線形イメージ・センサの位置がマッピングされた走査イメージを定義した部分を示した図。
【図２３】図２２に示す直線状のイメージ空間内の１つの画素を、２つの線形イメージ・

センサの位置で定義される空間上に逆マッピングした様子を示した図。
【図２４】図２１に示す直線状のイメージ空間の一部分で、マッピングされた一連の線形イメージ・センサ位置によって走査イメージを定義した部分を示した図。
【図２５】図２４に示す直線状の空間内の画素を線形イメージ・センサ位置によって定義された空間上に逆マッピングした様子を示した図。
【図２６】図１に示す走査装置によって取得された走査帯の集合を示した図。
【図２７】連続する走査帯を縫い合わせるための２ステップ処理に用いられるバッファを示した図。
【図２８】図２７に示す２ステップ処理を用いて収集された走査帯を図解した図。
【符号の説明】
１０　走査装置
１２　オリジナル１４上の曲がりくねった経路
１４　オリジナル
１６　イメージ・ディスプレイ
１８　走査装置１０の前面
２０　枢軸部材
２２　イメージ・センサ
２４，２６　ナビゲーション・センサ
２８，３５　発光ダイオード
３４，３６　照明光学系
３７　アンプ型ビーム・スプリッタ

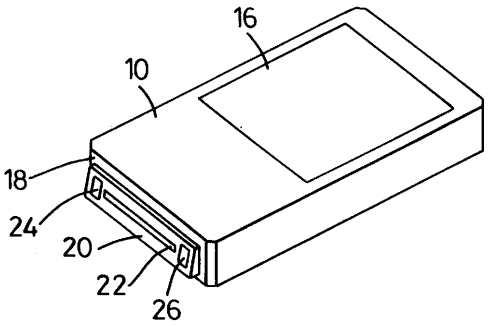
【図１】



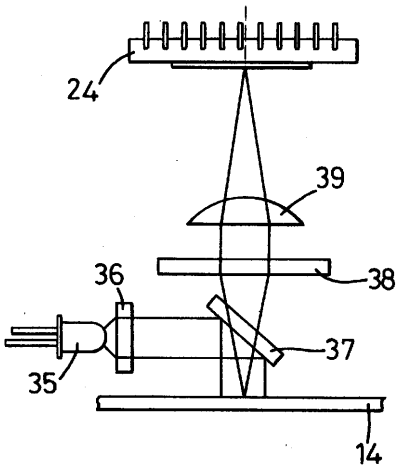
【図２】



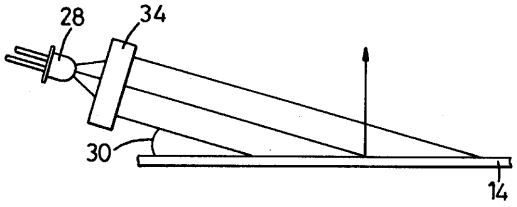
【図３】



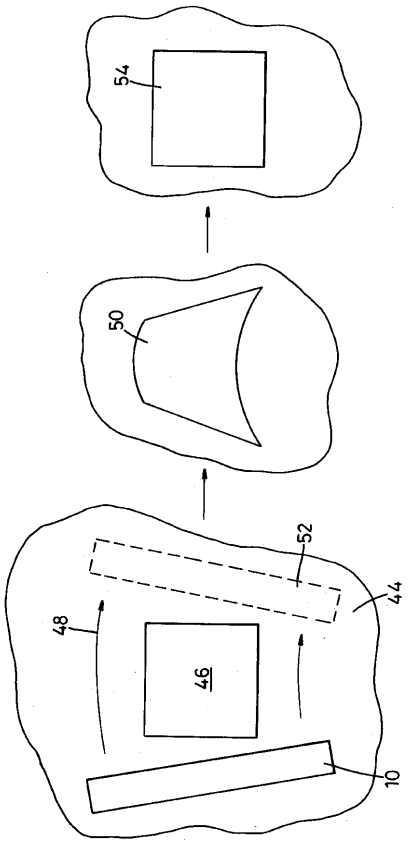
【図４】


24
39
36
38
37
35
14

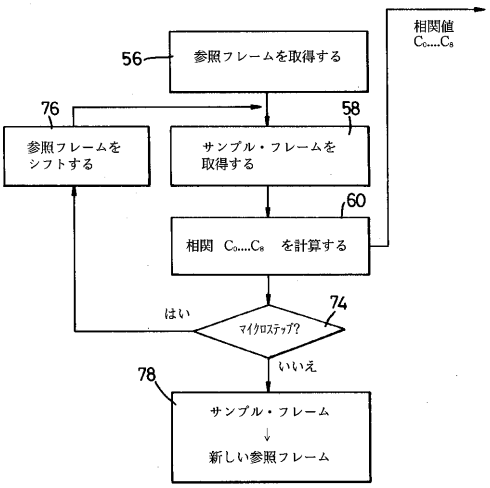
【図５】


28
34
30
14

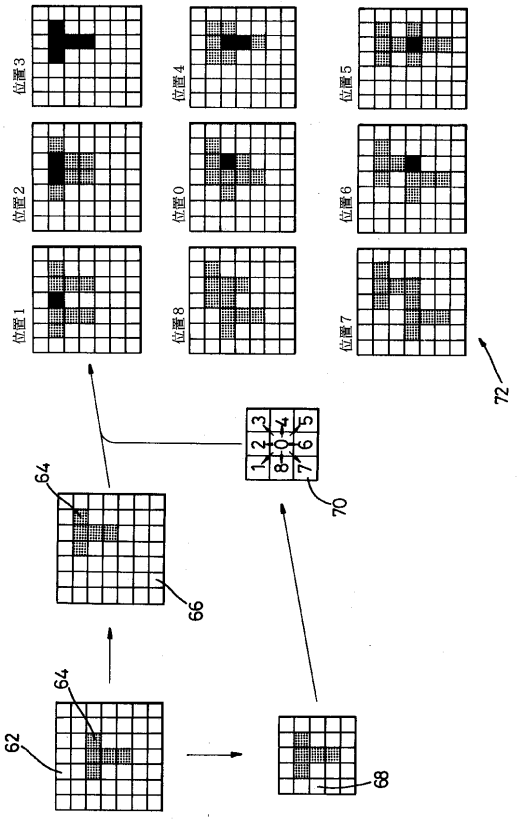
【図６】


54
50
52
44
48
46
10

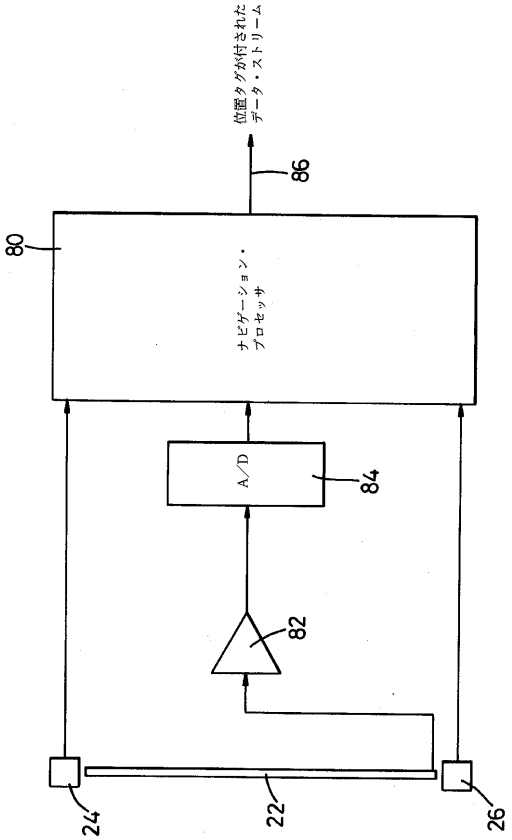
【図７】


相関値
C0....C8
56
参照フレームを取得する
76
参照フレームを
シフトする
58
サンプル・フレームを
取得する
60
相関 C0....C8 を計算する
はい
74
マイクロステップ?
いいえ
78
サンプル・フレーム
↓
新しい参照フレーム

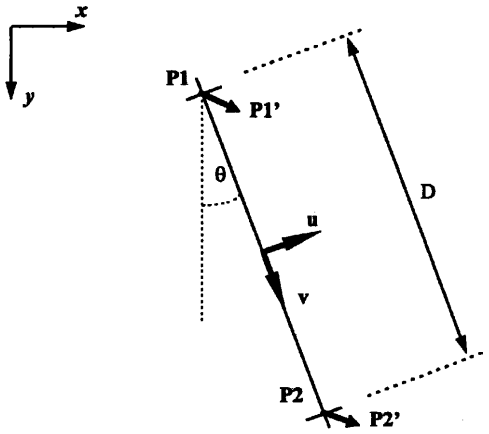
【図８】


位置3
位置4
位置5
位置2
位置0
位置6
位置1
位置8
位置7
72
1
2
3
8
0
4
7
6
5
70
64
66
64
62
68

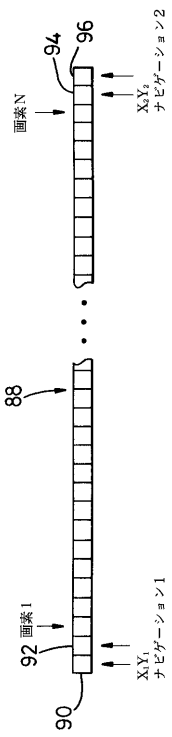
【図９】



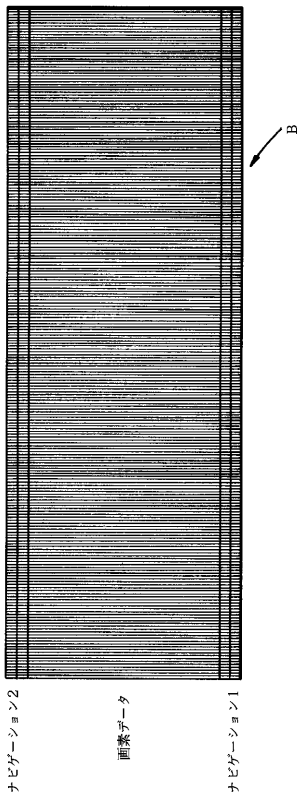
【図１０】



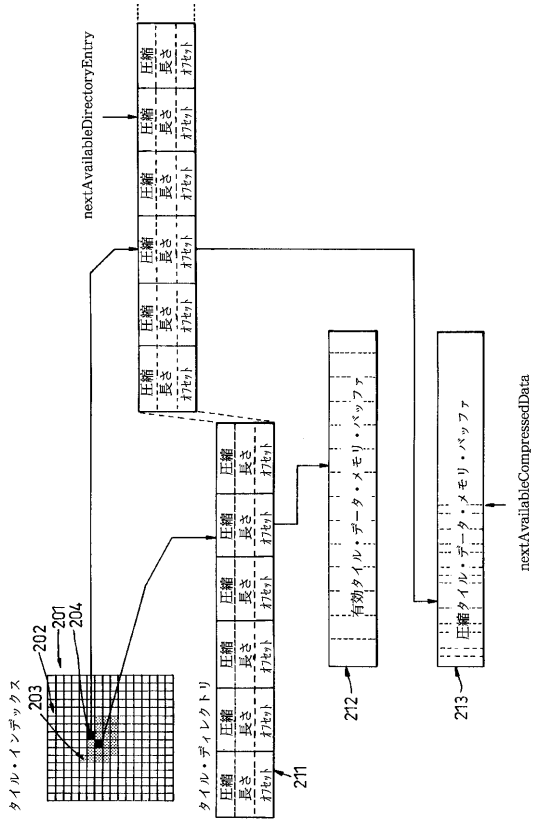
【図１１】



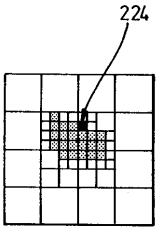
【図１２】



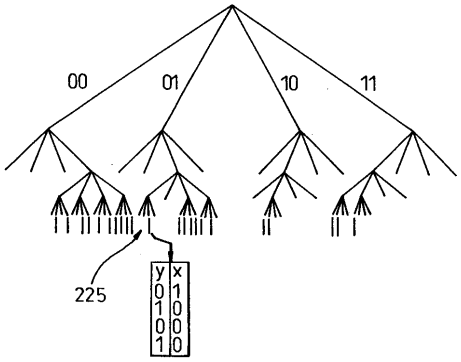
【図１３】



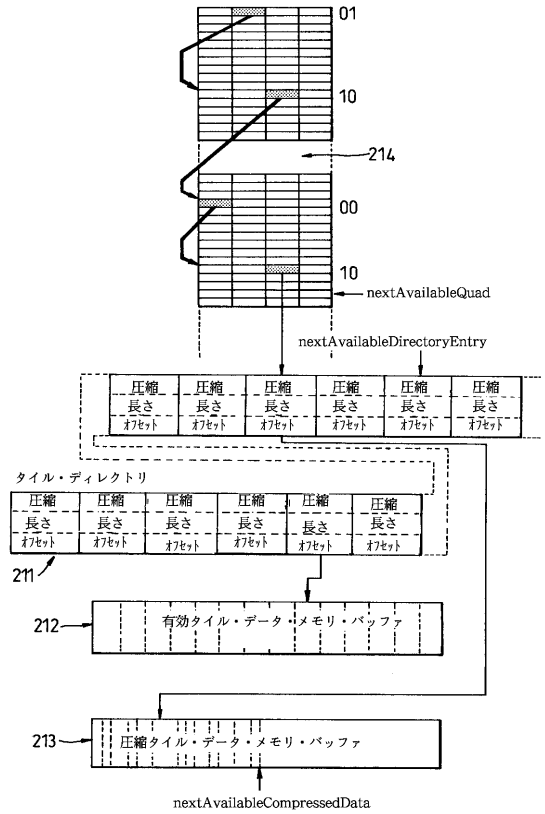
【図１４】



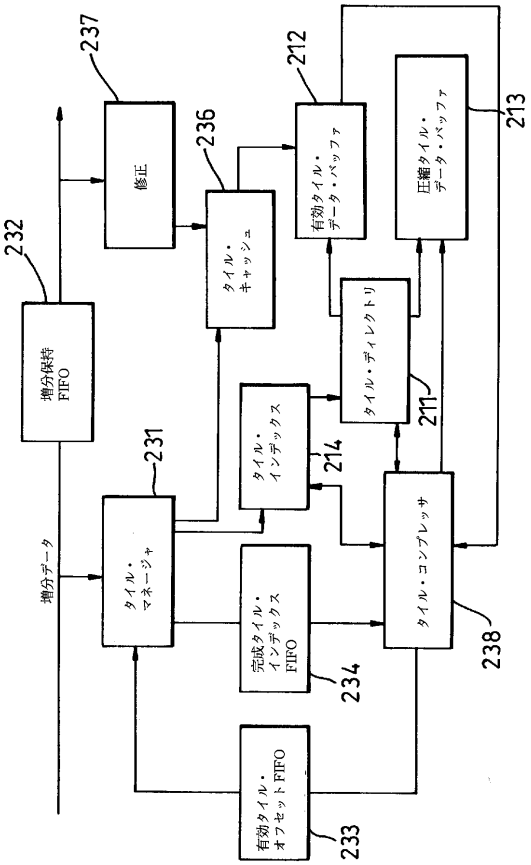
【図１５】



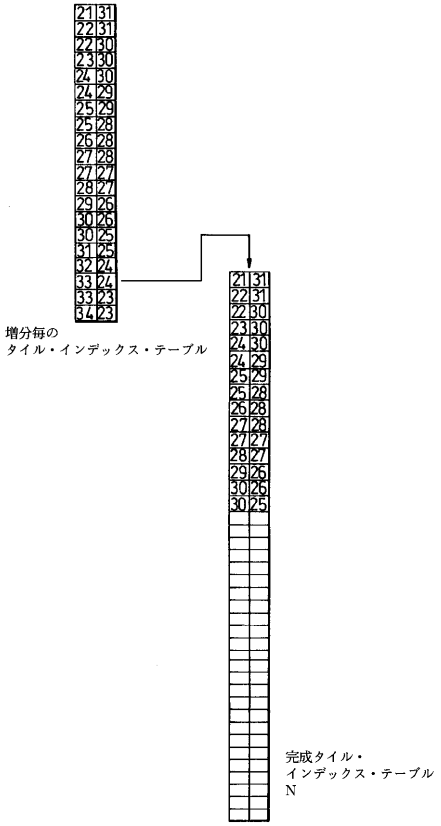
【図１６】



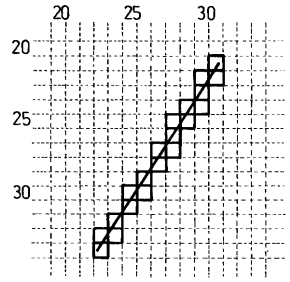
【図１７】



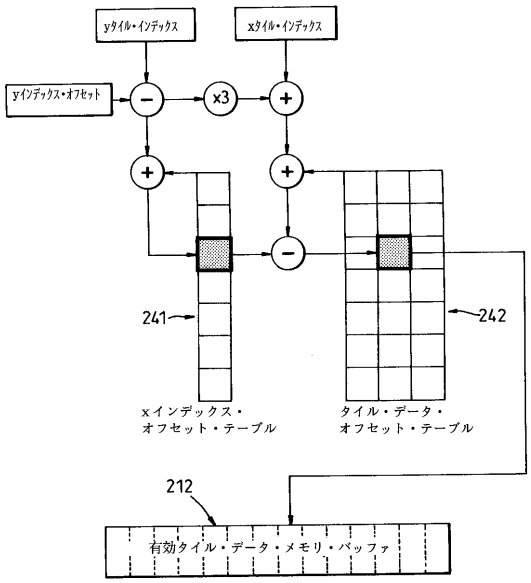
【図１８】



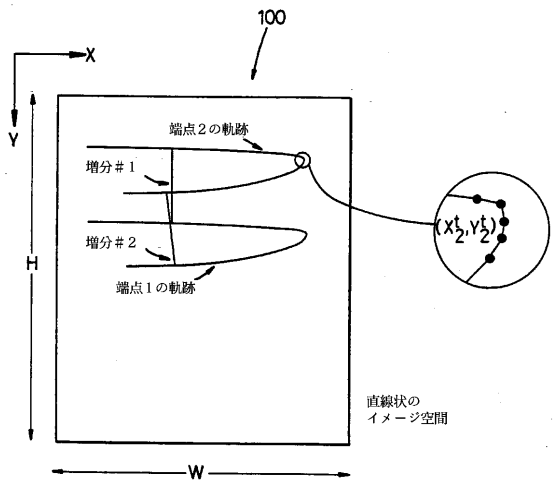
【図１９】



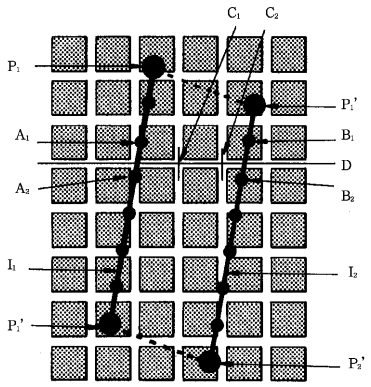
【図２０】



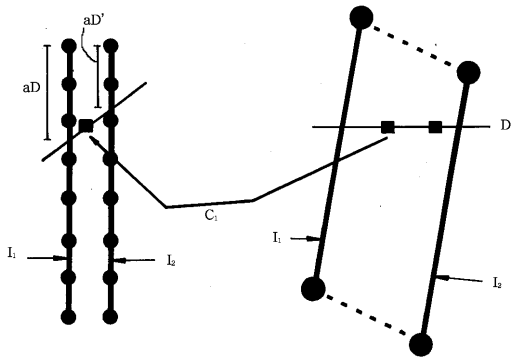
【図２１】



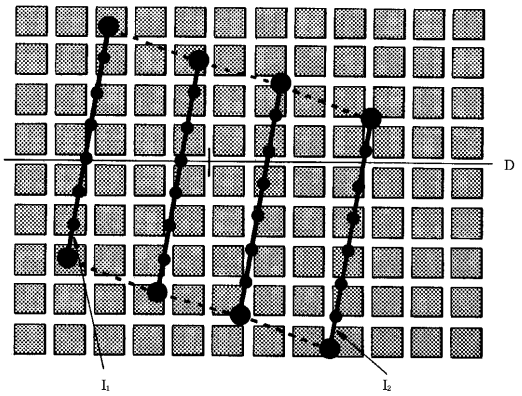
【図２２】



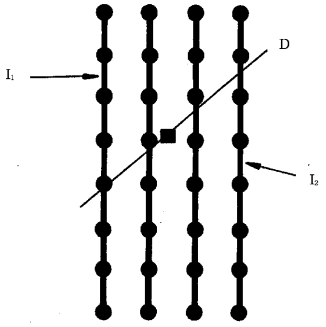
【図２３】



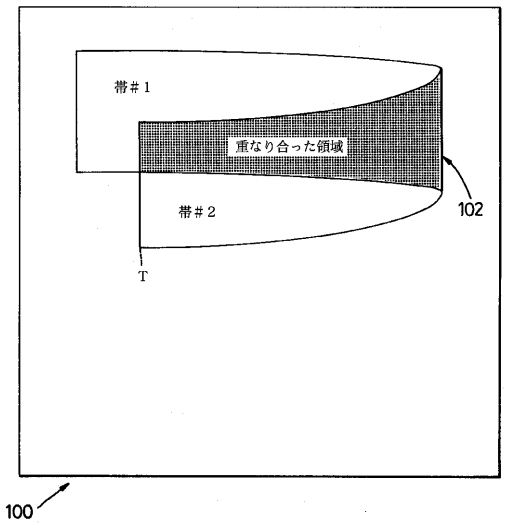
【図２４】



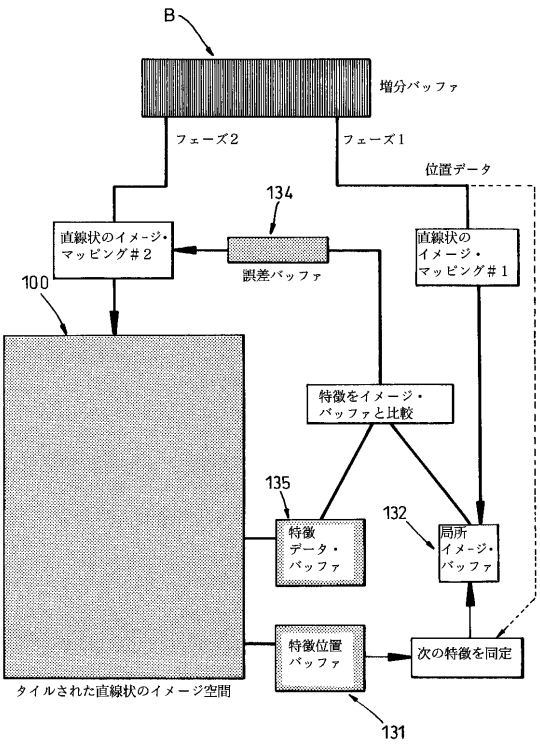
【図２５】



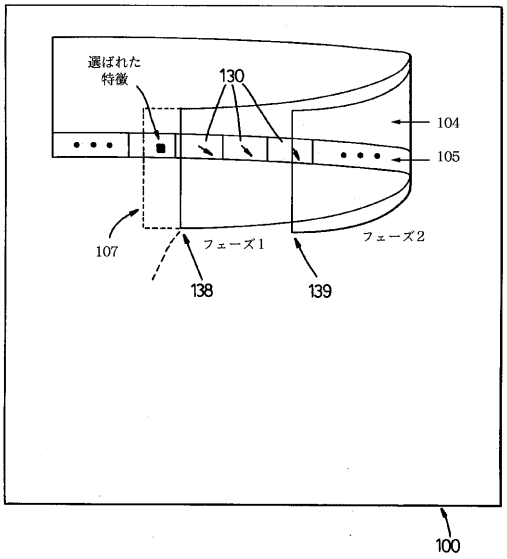
【図２６】



【図２７】



【図２８】



フロントページの続き

(72)発明者　リチャード・オリバー・カーン
イギリス、ビーエス１２、６エーワイ、ビリストル、ブラッドリー・ストーク、ザ・コモン・イースト、ローズ・コテージ（番地なし）

審査官　脇岡　剛

(56)参考文献　特開平０７－２１２５４１（ＪＰ，Ａ）
特開平０８－２６５５１８（ＪＰ，Ａ）
特開平０１－１５９７６８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06T 1/00
H04N 1/387