

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成20年6月19日(2008.6.19)

【公表番号】特表2005-507535(P2005-507535A)

【公表日】平成17年3月17日(2005.3.17)

【年通号数】公開・登録公報2005-011

【出願番号】特願2003-540928(P2003-540928)

【国際特許分類】

G 06 F 3/042 (2006.01)

【F I】

G 06 F 3/03 330 J

【誤訳訂正書】

【提出日】平成20年4月21日(2008.4.21)

【誤訳訂正1】

【訂正対象書類名】特許請求の範囲

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【特許請求の範囲】

【請求項1】

位置コーディングパターンの一部を構成する部分領域からセンサーにより取り込まれた画像に基づいて、前記部分領域の前記位置コーディングパターンにおける位置を計算し、前記位置の計算が失敗した場合に、それぞれが前記位置コーディングパターンにおける位置を定める複数のマッチング部分領域において前記位置コーディングパターンがどのようなパターンを有するかについての情報と、取り込まれた前記部分領域の画像から得られる情報とのマッチングを実行し、

前記マッチングの結果、前記複数のマッチング部分領域のそれぞれが定める位置のうち最も可能性があるとされた位置を、前記部分領域の前記位置コーディングパターンにおける位置として容認することを特徴とする位置デコーディング方法。

【請求項2】

前記マッチングは、以前に容認された位置の近隣のマッチング領域において行われる請求項1に記載の方法。

【請求項3】

前記マッチングは、前記位置コーディングパターンにおいて一度に一つの次元について行われる請求項1または請求項2に記載の方法。

【請求項4】

前記位置コーディングパターンは、複数の要素によって各位置をコーディングするものであり、前記複数の要素の各々は、少なくとも二つの異なる値をとり得るものであり、

前記マッチングは、

前記マッチング部分領域の各要素がとる値を決定することと、

それぞれの前記マッチング部分領域について、取り込まれた前記部分領域の画像から得られる情報に基づいて、前記部分領域の各要素が前記マッチング部分領域の各要素と同一の値をとる確度を決定することと、

を含む請求項1～請求項3のいずれか一項に記載の方法。

【請求項5】

取り込まれた前記部分領域の画像から得られる情報は、一組の確率値を有しており、前記確率値の各々は、前記部分領域の複数の要素の一つが前記少なくとも二つのとり得る値の一つをとる確率を示すものであり、

前記確度は、前記マッチング部分領域の要素がとる値についての確率値に基づいて決定される請求項4に記載の方法。

【請求項6】

前記確度は、前記マッチング部分領域の対応する要素と同一の値をとる前記部分領域の要素の数に基づいて決定される請求項4に記載の方法。

【請求項7】

前記位置コーディングパターンにおける位置の各々は、シンボルの行列によってコーディングされており、各シンボルが一つの要素を構成するものである請求項4～請求項6のいずれか一項に記載の方法。

【請求項8】

前記位置コーディングパターンにおける位置の各々は、シンボルの行列によってコーディングされており、前記シンボルの行列は、複数の群に分割され、各群が一つの要素を構成する請求項4～請求項6のいずれか一項に記載の方法。

【請求項9】

前記位置コーディングパターンにおける位置の各々は、シンボルの行列によってコーディングされており、前記シンボルの行列は、複数の群に分割され、一つの要素が、少なくとも二つの群によって定義される請求項4～請求項6のいずれか一項に記載の方法。

【請求項10】

前記シンボルの各々は、前記位置コーディングパターンにおける基準位置から四つの所定の方向のうちの一つの方向に変位されるマークを含む請求項7～請求項9のいずれか一項に記載の方法。

【請求項11】

一つの群に含まれる複数のシンボルの値は、循環数列における部分数列に対応しており、前記循環数列は、所定長さの前記部分数列の前記循環数列における場所が曖昧さなしに決定されるという特性を有する請求項8または請求項9に記載の方法。

【請求項12】

一つの群に含まれる複数のシンボルの値は、循環数列における部分数列に対応しており、前記循環数列は、所定長さの前記部分数列の前記循環数列における場所が曖昧さなしに決定されるという特性を有しております、

前記少なくとも二つの群により定義される要素は、前記循環数列におけるそれぞれの群の場所の間の差分である請求項9に記載の方法。

【請求項13】

前記部分領域からセンサーにより取り込まれた画像に基づいて計算された位置及び／又は前記マッチングの結果最も可能性があるとされた位置を、取り込まれた前記部分領域の画像から得られる情報を用いて、容認可能であるかどうか検証する請求項1～請求項12のいずれか一項に記載の方法。

【請求項14】

前記位置コーディングパターンにおける位置の各々は、複数の位置番号を用いてコーディングされており、前記複数の位置番号は、少なくとも二つが組になって差分数を定義し、複数の前記差分数によって前記位置が決定されるものであり、

前記マッチング部分領域において前記位置コーディングパターンがどのようなパターンを有するかについての情報は、前記複数の差分数を含んでおり、

取り込まれた前記部分領域の画像から得られる情報は、位置番号を有しております、

前記マッチングは、前記部分領域の画像から得られる位置番号のいかに多くが、前記差分数について、前記マッチング部分領域と対応するかを決定することを含む請求項1～請求項13のいずれか一項に記載の方法。

【請求項15】

コンピューターで実行されたときに、請求項1～請求項14のいずれか一項に従った方法を前記コンピューターに行わせるプログラムコードを有することを特徴とするコンピュータープログラム。

【請求項 1 6】

コンピューターで実行されたときに、請求項1～請求項14のいずれか一項に従った方法を前記コンピューターに行わせるコンピュータープログラムが格納されたことを特徴とするコンピューター読み取り可能な記憶媒体。

【請求項 1 7】

位置コーディングパターンの一部を構成する部分領域からセンサーにより取り込まれた画像に基づいて、前記部分領域の前記位置コーディングパターンにおける位置を計算する手段と、

前記位置の計算が失敗した場合に、それぞれが前記位置コーディングパターンにおける位置を定める複数のマッチング部分領域において前記位置コーディングパターンがどのようなパターンを有するかについての情報と、取り込まれた前記部分領域の画像から得られる情報とのマッチングを実行する手段と、

前記マッチングの結果、前記複数のマッチング部分領域のそれぞれが定める位置のうち最も可能性があるとされた位置を、前記部分領域の前記位置コーディングパターンにおける位置として容認する手段と、

を備えたことを特徴とする位置デコーディングデバイス。

【誤訳訂正 2】

【訂正対象書類名】明細書

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【書類名】明細書

【発明の名称】位置コーディングパターンをデコードするための方法およびデバイス

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、概して、位置コーディングパターンのデコーディングに関し、より詳細には、位置コーディングパターンの、センサーによって記録された、部分領域から決定される情報に基づいて位置を計算するステップを有する位置デコーディングのための方法に関する。

【0002】

本発明は、また、位置コーディングパターンをデコードするのに使用され得るデバイス、コンピュータープログラムおよび記憶媒体に関する。

【背景技術】

【0003】

最小の所定のサイズの位置コーディングパターンの部分領域についての位置コーディングパターンにおける絶対位置が、部分領域における情報および所定のアルゴリズムを用いて計算され得る性質を有する位置コーディングパターンがある。そのような位置コーディングパターンの例は、各位置がシンボルを用いてコーディングされる米国特許U.S.5,477,012号(O.Sekendur)、ならびに、各位置が複数のシンボルによってコーディングされる国際特許WO 92/17859(Hewlett Packard Co.)、国際特許WO 00/73983(Anoto AB)、国際特許WO 01/26033(Anoto AB)および米国特許U.S.5,973,110(Xerox Corp.)に開示されている。

【0004】

このタイプの位置コーディングパターンは、例えば、位置コーディングパターンが提供されたベース上でユーザーユニットを用いて手によって書かれおよび/または描かれた情報のデジタル記録のために、使用され得る。ユーザーユニットが移動している間に、位置コーディングパターンの画像は、ユーザーユニットのチップに局所的に連続的に記録される。画像の各々における一つ以上のシンボルは、位置にデコードされる。デコードされ

た位置は、合わせてベース上でのユーザーユニットの移動のディジタル記述を構成する。

【0005】

位置コーディングパターンがベース上で完全に再現されれば、そして、位置コーディングパターンが記録画像内で完全に画像化されれば、位置デコーディングが非常に良好に機能することが予想され得る。しかしながら、実際には、位置コーディングパターンは、ベースに適用されたときに、多少歪むおそれがある。ユーザーユニットのイメージングシステムにおける欠陥のために、また、ユーザーが書くときにユーザーユニットを傾けると位置コーディングパターンが遠近法によって画像化されるために、ユーザーユニットが位置コーディングパターンを完全に画像化しないということも起こる。このことにより、位置コーディングパターンから正しくない位置がデコードされるという結果も生じ得る。

【0006】

正しくなくデコードされた位置は、位置コーディングパターンが、誤り検出を可能とする冗長な情報を含んでいると、発見され得る。また、正しくなくデコードされた位置は、以前におよび／またはその後にデコードされた位置までの距離が不合理である場合にも、発見され得る。

【0007】

正しくなくデコードされた位置が検出されると、その位置は保持されまたは拒絶され得る。位置が保持されると、ディジタル的に記録された位置は、ベース上のユーザーユニットの移動を正確に表現しないであろう。代わりに、正しくなくデコードされた位置が拒絶されるならば、ディジタル的に記録された位置に「ギャップ」が存在するであろう。多くのそのようなギャップは、ベース上に何が書かれたかを解釈する可能性を妨害し得る。

【特許文献1】米国特許U.S. 5,477,012号明細書

【特許文献2】国際公開WO 92/17859号パンフレット

【特許文献3】国際特許WO 00/73983号パンフレット

【特許文献4】国際特許WO 01/26033号パンフレット

【特許文献5】米国特許U.S. 5,973,110号明細書

【発明の開示】

【発明が解決しようとする課題】

【0008】

本発明の一般的な目的は、この問題を解決することにあり、位置コーディングがなされたベース上で生成される情報のディジタル表現の品質を改善することにある。

【0009】

ある特定の目的は、位置コーディングパターンの画像から、正確にデコードされ得る位置の数を増大させることにある。

【0010】

さらにもう一つの特定の目的は、デコーディングのために必要とされる時間を不適当に増大させることなしに、正確にデコードされる位置の数を増大させることにある。

【0011】

他の特定の目的は、デコーディングのためのメモリースペースおよび／またはプロセッサー容量のための要求を不適切に増大させることなく、正確にデコードされる位置の数を増大させることにある。

【課題を解決するための手段】

【0012】

これらの目的そして次の説明から明らかになるであろう他の目的は、添付される特許請求の範囲の各請求項に従った方法、コンピュータープログラム、記憶媒体およびデバイスを使用して全体的にまたは部分的に達成される。

【0013】

第一の態様によれば、本発明は、より詳細には、位置コーディングパターンの一部を構成する部分領域からセンサーにより取り込まれた画像に基づいて、部分領域の位置コーディングパターンにおける位置を計算するステップを有する位置デコーディング方法に関する

る。この方法は、位置の計算が失敗した場合に、それぞれが前記位置コーディングパターンにおける位置を定める複数のマッチング部分領域において位置コーディングパターンがどのようなパターンを有するかについての情報と、取り込まれた部分領域の画像から得られる情報とのマッチングを実行するステップと、マッチングの結果、複数のマッチング部分領域のそれぞれが定める位置のうち最も可能性があるとされた位置を、部分領域の位置コーディングパターンにおける位置として容認するステップとをさらに含む。

【0014】

位置の計算が失敗したときにマッチングが実行されることにより、位置が得られる。ここで得られる位置は、多くの場合、失敗した計算を通しておそらく得られる位置よりも、部分領域の画像化時のセンサーの位置をより良く反映する位置である。さらに、デジタル表現におけるギャップが回避される。

【0015】

マッチングは、位置計算が失敗したときに直ちに行なわれ得る。その代わりに、マッチングは、もっと後の段階において、さらに多くの位置情報が記録されたときに、行われることもできる。これにより、マッチングが単純化され得る。

【0016】

位置計算は、例えば、計算における部分的な結果または最終的な結果が明らかに正しくないために、失敗することがある。部分的な結果が位置コーディングパターンのデコーディングにおいて得られるべきでない値を含むときに、その部分的な結果は明らかに正しくない可能性がある。デコードされた位置が、直前に先行しまたは直後に続く正確にデコードされた位置から時間によって決まるある距離以内に位置されていなければ、最終的な結果は、明らかに正しくない可能性がある。あるいは、最終的な結果があまりに信頼できないほど離れているとみなされるならば、その最終的な結果は、明らかに正しくない可能性がある。

【0017】

ここで、センサーによって画像化される部分領域が、マッチング部分領域よりも大きいサイズであり、そして特に位置の計算を可能とするのに厳密に必要であるよりも一層多くの情報を含み得ることが指摘されるべきである。

【0018】

マッチングは、位置を定義する位置コーディングパターンにおいて考えられる全ての部分領域について行なわれ得る。しかしながら、位置コーディングパターンが大きければ、マッチングは時間がかかるものになる。その代わりとして、決定されるべき位置が以前に容認された位置に相対的に近接して位置される可能性が最も高いので、マッチングは、以前に容認された位置の近隣の限定されたマッチング領域にのみ実施されるように制限され得る。位置計算が失敗した部分領域と時間的に近いところで画像化された部分領域に基づいて容認された位置がいくつかあるならば、一以上の容認された位置が、限定されたマッチング領域を定めるために使用され得ることはもちろんである。

【0019】

限定されたマッチング領域のサイズは、例えば、それについての位置計算が失敗した部分領域の画像化の前および／または後にセンサーの速度および／または加速度を用いて、予め決定されまたは動的に決定され得る。

【0020】

失敗した位置計算についての部分領域が、以前に容認された位置からの距離 d （但し、 d は、位置コーディングパターンにおける位置と位置の間またはシンボルとシンボルの間の距離である）以内に配置されると期待できるとすれば、失敗した位置計算についての部分領域における情報は、位置コーディングパターンにおける $(2d + 1)^2$ の部分領域からの情報とマッチングする必要がある。その代わりとして、マッチングが、位置コーディングパターンにおいて一度に一次元について行われるならば、失敗した位置計算についての部分領域における情報は、 $2(2d + 1)$ の部分領域からの情報とマッチングする必要があるだけである。しかしながら、このことは、位置が、位置コーディングパターンにお

いて異なる次元で別々にデコードされ得るものであることを必要とする。また、マッチングは、全ての次元について実施される必要がないことも強調されるべきである。位置計算が一つの次元について成功するものの他の次元については失敗するならば、位置計算が失敗した次元においてマッチングが実施されれば充分である。

【0021】

位置コーディングパターンは、各位置が、各々少なくとも二つの異なる値をとり得る複数の要素を用いてコーディングされるように設計され得る。マッチングでは、画像化された部分領域が要素のレベルでマッチング部分領域と比較され、その二つのマッチングされる部分領域で要素が同一の値をとる確度が決定される。

【0022】

確度は、例えば、対応する要素の値の数から構成され得る。他の例として、画像化された部分領域からの情報は、一組の確率値から構成され得る。より詳細には、組における各確率値は、画像化された部分領域の要素の一つがとり得る値の一つをとる確率を示し得る。確率値の組は、各要素のとり得る値の各々についての確率値、または、単にある要素と値についての確率値を含み得る。確度は、マッチング部分領域の要素がとる値についての確率値に基づいて決定される。マッチング部分領域の要素が、既知の位置コーディングパターンの構成内容によって知られている値の組み合わせをとるならば、例えば、画像化された部分領域における要素がこれらの値をとる組み合わせ確率を決定するために、対応する要素の値についての確率値がかけ算され得る。画像化された部分領域から決定される要素の確率値に基づくマッチングによって、部分領域における情報は、その部分領域の要素の値が決定されるだけよりも大きな程度まで利用されるであろう。

【0023】

上述された要素は、例えば、シンボルに対応し得る。シンボルは、位置コーディングパターンがそれによって構成され、位置コーディングパターンを使用するときにセンサーによって画像化されるものである。要素は、より高いレベルまたは処理されたレベルにおける情報にも対応し得る。シンボルは、例えば、群に分割され得る。ここで、群は、要素を構成するものであり、要素は、群におけるシンボルの値によって異なる値をとり得るものである。さらにまた、要素は、二つ以上の群が共同して定義され得る。したがって、画像化された部分領域は、位置コーディングパターンにおいて見出されるのと同一の分解能でマッチング部分領域とマッチングされる必要はなく、マッチングにおける時間と、画像化された部分領域からの情報がマッチングを待って保存されることとなるときのメモリースペースとの両方を節約し得る。

【0024】

マッチングの付加的な最適化は、利用可能な情報を全部はマッチングしないことによってなされ得る。

【0025】

マッチングを実施した後、画像化された部分領域に基づいて容認される位置は、部分領域からの情報を用いて検証され得る。この部分領域からの情報は、普通は、位置計算ノマッチングに必要とされるよりも多くの情報を含んでいる。マッチングが常に結果として位置を得るので、検証ステップを行うことが有利であり得る。マッチング領域が、以前に容認された位置の近隣で選択されると、この位置への距離は、マッチングによって得られた位置を決定するのに用いることができないので、画像化された部分領域において利用可能な付加的な情報を用いて位置を検証することが、有利であり得る。検証は、有利には、確率計算を用いてなされ得る。

【0026】

位置コーディングのための上述された方法は、A n o t o A B によって開発され且つ例えば国際特許W O 0 1 / 2 6 0 3 3に記述されている位置コーディングパターンと一緒に有利に使用され得る。このパターンの変形は、本出願における付録Aに記述されている。位置コーディングパターンは、位置が、位置コーディングパターンにおける各次元において別々にデコードされ得るという性質を有している。

【 0 0 2 7 】

上述の方法は、ハードウェアまたはソフトウェアにて成し遂げられ得る。それゆえ、本発明は、コンピューター内で実行されたときに、コンピューターに、方法の請求項のいずれか一項に記載の方法を実行させるプログラムコードを含むコンピュータープログラムにも関連している。

【 0 0 2 8 】

当該コンピュータープログラムは、位置がデコードされる、部分領域を画像化するセンサーと同一のユニット内にまたは他のユニット内に配設されるプロセッサーによって実行され得る。

【 0 0 2 9 】

コンピュータープログラムは、RAM、ROM、光学式ディスク、磁気テープ、または他の何らかの商業的に入手可能な記憶媒体のような記憶媒体の形態で格納され且つ配布され得る。記憶媒体は、伝播信号でもあり得る。

【 0 0 3 0 】

もう一つの態様によれば、本発明は、位置コーディングパターンの一部を構成する部分領域からセンサーにより取り込まれた画像に基づいて、部分領域の位置コーディングパターンにおける位置を計算する手段を備えた信号処理ユニットを有するデバイスに関する。信号処理ユニットは、位置の計算が失敗した場合に、それぞれが位置コーディングパターンにおける位置を定める複数のマッチング部分領域において位置コーディングパターンがどのようなパターンを有するかについての情報と、取り込まれた部分領域の画像から得られる情報とのマッチングを実行する手段と、マッチングの結果、複数のマッチング部分領域のそれぞれが定める位置のうち最も可能性があるとされた位置を、部分領域の位置コーディングパターンにおける位置として容認する手段を、さらに備えている。

【 0 0 3 1 】

当該デバイスの利点は、上述の方法の説明から明白である。

【 0 0 3 2 】

信号処理ユニットは、例えば、適切にプログラムされたプロセッサーによって実現されてもよく、ASIC（特定用途向け集積回路）またはFPGA（フィールドプログラム可能ゲートアレイ）のような、特に適合されたハードウェアによって実現されてもよく、デジタルまたはアナログ回路あるいはそれらの組み合わせによって実現されても良い。

【 0 0 3 3 】

このように、信号処理ユニットは、デコーディング方法のステップの各一つを実施するための手段を有する。

【 発明を実施するための最良の形態】**【 0 0 3 4 】**

さて、本発明は、添付図面を参照して実施の形態によってより詳細に説明されるであろう。

【 0 0 3 5 】

図1は、本発明がそこで使用され得る情報管理のためのシステムを概略的に示している。当該システムは、一枚の紙の形態のベース1、ユーザーユニット2および外部ユニット3を具備している。その一枚の紙は、位置コーディングパターン4を備えており、その小さな一部分のみが拡大された状態で概略的に示されている。ユーザーユニット2は、通常のペンと同様の方法で一枚の紙1上で書くのに使用され、また、同時に何が書かれたかをデジタル形式で記録するのに使用され得る。デジタル的に記録された情報は、ユーザーユニット2および/または外部ユニット3において処理され得る。外部ユニット3は、自動的に（オンラインで）またはユーザーによって信号が与えられたときに、それが送信され得るものである。デジタル的に記録された情報が一枚の紙の上でそれを正確に反映するために、位置デコーディングが可能な限り正確に且つ完全になされることが重要である。

【 0 0 3 6 】

以下により詳細に説明される位置コーディングパターン4は、所定の規則に従って配列されまたは設計されるグラフィカルシンボルから作られている。そのことは、位置コーディングパターンの部分領域が正確に画像化され、その部分領域が所定の最小サイズを有しているならば、位置コーディングパターンにおける部分領域の位置は、画像化される部分領域における一つ以上のシンボルおよび所定のデコーディングアルゴリズムを用いて曖昧でなく決定され得ることを意味する。位置は、デカルト座標系または他の座標系における一対の座標として取得され得る。

【0037】

ユーザーが、一枚の紙1の上にユーザーユニット2を用いて書くとき、ユーザーユニットは、当該ユーザーユニットの先端にて位置コーディングパターンの部分領域の画像を連続的に記録するであろう。各画像は、位置にデコードされる。そのようなデコードされた位置の配列は、一枚の紙の面上におけるユーザーユニット2の運動のディジタル表現であって、一枚の紙の上に書かれたもののディジタル表現を構成する。

【0038】

位置コーディングパターンのデコーディングは、ユーザーユニット2において、または外部ユニット3において、または多少処理された形で位置コーディングパターンの画像がそこに送信される何らかの他のユニットにおいて行なわれ得る。

【0039】

上述されたように、部分領域における情報は、所定のデコーディングアルゴリズムを用いてデコードされる。これは、部分領域が正確に画像化される限り良好に機能する。しかしながら、現実の条件のもとで、ユーザーユニットは、例えば貧弱な照明条件、または部分領域が遠近法によって画像化されるようにする傾斜位置におけるユーザーのペンの保持、または画像を変形させるイメージング機構のために、しばしば部分領域を全く正確に画像化することができない。そして、デコーディングアルゴリズムが、シンボルを誤解するまたはシンボルを見逃すということが起こり、それは位置が不正確にデコードされるという結果を生ずる。また、エラーは、位置コーディングパターンが一枚の紙の上にあまり正確には再現されず、一枚の紙に適用されたときにある程度変形された場合にも、生じ得る。何らかの理由のために、例えばあまりにも信頼できないとみなされること等により、容認され得ないならば、位置は、不正確であるとみなされることもできる。

【0040】

ある条件のもとにおいては、例えば、位置コーディングパターンが、エラー検出を可能とする冗長な情報を含むならば、あるいは、デコードされた位置が、以前にまたはその後にデコードされる位置に関して不合理であるならば、エラーは発見され得る。

【0041】

図2は、不正確にデコードされた位置の問題がどのようにして解決され得るかを、概略的に示している。一枚の紙200は、位置コーディングパターンを備えている。位置コーディングパターンは、明確さのために図中には示されていない。第一の部分領域Aは、いかなるデコーディングエラーも検出されることなくデコードされ、それゆえ正確にデコードされた位置として容認されている位置P1をコーディングする。第三の部分領域Cは、やはり正確にデコードされた位置として容認されている位置P3をコーディングする。ユーザーユニットは、部分領域Aを画像化した後、部分領域Cを画像化する前に、位置P2を有する部分領域Bを画像化している。しかしながら、歪んだ画像のせいで、部分領域Bにおける情報は、断続線によって示されている部分領域B'に対応する誤った位置P2'にデコードされる。

【0042】

部分領域Bを使用できないとして拒絶する代わりに、マッチング動作が実施される。マッチング動作は、直ちに実施される必要はないが、後に、一つ以上の位置が正確にデコードされたときに、実施されても良い。この例においては、デコーディングは、部分領域Cの正確なデコーディングの後に実施される。

【0043】

より詳細には、画像化された部分領域 B から決定される情報が、位置コーディングパターンが部分領域 A および C のまわりのマッチング領域における複数の異なる部分領域においてどのように構成されているかについての情報と、マッチングされまたは比較される。マッチングが行なわれ得る領域の一例が、鎖線によって与えられている。当該領域は、所定領域として選択され、または、動的に選択されることができる。マッチング領域を選択するとき、一つ以上の容認された位置が考慮に入れられ得る。その位置の対応する部分領域は、その位置計算に失敗する以前および／または失敗した後に記録されてもよい。特に、位置計算に失敗した部分領域の記録の前および／または後にユーザーユニットの速度および加速度が、マッチング領域の査定のためのベースとして決定され且つ使用され得る。

【0044】

マッチングは、一つまたは二つの次元において行なわれ得る。位置コーディングパターンの構成によっては、位置デコーディングは、いくつかの場合において、各次元について別々に行なわれ得る。そのとき、第一の次元における位置計算が成功し且つこの次元について容認された位置が結果として生じるが、第二の次元における位置計算が失敗し且つ明らかに不正確である第二の次元における位置が結果として生じるということが起こるかもしれない。ほとんどの場合において、そのとき、マッチングは、正確な次元においてのみ行なわれるべきである。

【0045】

例えば、図 2において、位置 P 2 の x 座標の計算が成功しそして y 座標の計算のみが失敗したと仮定する。そうすると部分領域 B ”に対応する位置 P 2 ”が取得される。y 方向における不適正な位置は、容認された位置 P 1 および P 3 からあまりに遠く離れて位置されるので、検出される。そのとき、画像化された部分領域 B における情報は、範囲 M ’で y 座標を変化させる複数の異なる部分領域からの情報とマッチングされることのみを要する。

【0046】

このようにして、マッチングは、画像化された部分領域から生じ且つそれゆえいつも 100% 正確であるとは限らない情報と、複数の異なるマッチング部分領域における位置コーディングパターンの様相についての既知の情報との間で実行される。マッチング部分領域と画像化された部分領域との各マッチングにおいて、画像化された部分領域がそのマッチング部分領域を画像化したものである確率が決定される。マッチング領域における全てのマッチング部分領域がマッチングされたときに、最も高い符合確率が取得されたマッチング部分領域に対応する位置が選択される。

【0047】

もちろん、一枚の紙の上に書かれているもののデジタル記録が、リアルタイムでなされ得るように、マッチングは、できる限り迅速であることが望ましい。そして、マッチングは、可能な限り少ないプロセッサー容量およびメモリースペースを必要とするであろう。後者は、通常、限られたプロセッサー容量および限られたメモリースペースを持っていると予想され得るユーザーユニットにおいて、マッチング動作が実行されるときに特に望ましい。それゆえ、種々の方法でマッチングを最適化することが望ましい。

【0048】

上述の例において、不適正な位置 P 2 ’ および P 2 ”は、紙の上の位置である。しかしながら、それらは、紙の上でない位置であることもあり得る。

【0049】

さらにまた、図 2において、一枚の紙に関する部分領域のサイズは、明瞭さのために実際の実施の形態に比して大きく拡大していることが指摘されるべきである。

【0050】

図 3 は、位置をコーディングする部分領域 300 の例を示している。この部分領域は、ドット 310 にて各々概略的に図示されている 6 × 6 のシンボルを含む。明瞭さのために、シンボルの異なる値は示されていないが、全てのシンボルは同一のドットで示されている。各シンボルは、所定数の異なる値、例えば「0」および「1」をとり得る。シンボル

の値は、ドットのサイズ、ドットの位置またはドットの形状のような、複数の異なるパラメーターを用いてコーディングされ得る。複数の異なる数のシンボルを有する部分領域が使用され得ることは言うまでもない。

【0051】

既知の内容の部分領域を有する部分領域300のマッチングは、複数の異なるレベルで行うことができる。例えば、マッチングは、シンボルレベルで行うことができる。この場合、部分領域300から決定される各シンボル値は、既知の内容の部分領域における対応するシンボル値と比較し得る。そして、正しいシンボル値の数は、部分領域300が既知の内容の部分領域に対応する確率の指標として使用され得る。代わりに、部分領域300におけるシンボルについての値をデコーディングするとき、各シンボルがとり得る可能性のある値、例えば「0」および「1」、をとる確率が、決定され得る。各マッチングされた部分領域について、組み合わせ確率が決定され得る。この組み合わせ確率は、部分領域300におけるシンボルが、マッチング部分領域における値の組み合わせを正確にとる確率である。最大の組み合わせ確率を有するマッチング部分領域の位置は、部分領域300についてのデコードされた位置として選択される。

【0052】

マッチングは、その代わりとして、「より高い」レベルにてなされ得る。シンボルは、群をなして、例えば行または列ベクトルとして、配列され得る。ベクトル値は、各ベクトルについて決定される。図3は、320で示された一つの群の例を示している。シンボル値の代わりに、群値またはベクトル値が、このように各々の部分領域のマッチングにおいて比較される。そのことは結果として各部分領域についてより少ない比較で済む。シンボルレベルでのマッチングと同様に、各群／ベクトルが、その可能性のある値の各々をとる確率を決定すること、および、各マッチングした部分領域についての組み合わせ確率を決定すること、も可能である。このことは、各部分領域のマッチングにおいて、より少ない確率が組み合わされるべきであることを意味する。群レベルのマッチングは、それがシンボル値の代わりに群値を格納しましたは計算するのに充分であるので、複数の異なるマッチング部分領域においてどのようにして構成されるかについての情報を得ることに関しての単純化をも意味している。

【0053】

マッチングは、画像化された部分領域からさらに大きな程度まで処理された情報についてもなされ得る。例えば、二つ以上の群から生ずる情報がマッチングで比較され得る。

【0054】

以下において、位置デコーディングは、本出願人によって開発され、特に、本出願の提出の時点では公然と入手可能ではない、国際特許出願PCT/SE02/01243によって、その保護が請求されている、特定の位置コーディングパターンを参照してより詳細に例示され且つ説明されるであろう。位置コーディングパターンおよびそれについてのデコーディングは、図4～図9を参照して付録Aに詳細に説明されるであろう。しかしながら、以下に、そのパターンの簡潔で単純化された説明を行う。

【0055】

位置コーディングは、以下において差分数列と称される数列に基づいている。この差分数列は、もしも所定長さの任意の部分数列、例えば5つの差分数を有する部分数列を採用すれば、この部分数列は、差分数列における曖昧でなく決定される位置を常に有するという性質を有する。言い換えれば、この部分数列は、差分数列に一度だけあらわれ且つそれゆえ位置決定のために使用され得る。より詳細には、差分数列は、位置コーディングパターンのy軸に沿うだけでなくx軸にも沿って「延びている」。

【0056】

実際の位置コーディングパターンは、四つの異なる値、0～3をとり得る単純なグラフィカルシンボルから作られる。各シンボルは、例えば、ドットの形状を有し得て、そして公称の基準位置またはラスターポイントに関連する四つの異なる方向のうちの一つにおける距離を変位され得るマークからなっている（図6を参照されたい）。公称の基準位置は

、目に見えないラスターにおけるラスター・ラインの間の交差点からなる。シンボルの値は、変位の方向によって決定される。各シンボル値0～3は、 x 座標をコーディングするために用いられる1ビットと y 座標をコーディングするために用いられる1ビットに、すなわち「0, 0」「0, 1」「1, 0」および「1, 1」のビット対に、変換され得る。したがって、コーディングは、 x 方向と y 方向に別々になされるが、グラフィカルコーディングは x および y ビットに共通のシンボルを用いてなされる。

【0057】

各位置は、 6×6 のシンボルによってコーディングされる。この 6×6 のシンボルは、このように位置について、 x 座標のために 6×6 ビット行列に、また、 y 座標のために 6×6 ビット行列に変換され得る。 x ビット行列について考えると、これは各6ビットの6列に分割され得る。一つの列におけるビットの各列は、 6 ビット長の循環主数列における部分数列を構成する。この循環主数列は、6ビットの長さを有する部分数列が選択されると、その部分数列が主数列における曖昧でなく決定された場所を持つという性質を有する。6つの列は、このようにして主数列における6つの場所に対応する6つの位置番号または列値に変換される。これら6つの位置番号の間に、5つの差分数が対をなして形成され得る。この5つの差分数は、差分数列の部分数列を構成し、同様に曖昧でなく決定される位置、すなわち x 軸に沿って曖昧でなく決定される位置を有する。ある x 座標について、位置番号は、 y 座標に依存して変動するであろう。他方で、差分数は、全体の位置コーディングパターンにおける列において循環して繰り返す主数列に従って常に位置番号が変化するから、 y 座標とは無関係に同じであろう。

【0058】

対応して、 y ビット行列における6行は、主数列における6つの位置番号を定義する。これら6つの位置番号は、5つの差分数を定義する。この5つの差分数は、差分数列の部分数列を構成し、それゆえ y 軸に沿って曖昧でなく決定される位置を有する。

【0059】

位置コーディングパターンが使用されるとき、上述されているように、位置コーディングパターンの異なる部分の画像が記録される。通例、画像は、 6×6 よりもかなり多くのシンボルを含んでいる。実際には、デコーディングは、標準的に、画像において最善の 8×8 シンボルに基づいている。余分なシンボルが、とりわけ、エラー検出および／またはエラー訂正のために、使用される。それは、付録AおよびBにより詳細に説明されている。

【0060】

位置コーディングパターンは、面上にいつも完全に再現されるとは限らない。そして、ユーザーエンティティは、ときには画像化の間に位置コーディングパターンを変形させることがある。そのため、ドットの変位、すなわちシンボルの値を、全く間違いなく決定することは、困難であるかもしれない。それゆえ、現実の実施の形態においては、その代わりとして各シンボルが4つの異なる可能性のある値の各々をとる確率を決定することが可能である。したがって、画像化された部分領域は、行列における各行列要素について4つの確率値を有する行列にデコードされる。そこでは各行列要素は、シンボルに対応する。この行列は、次には、二つの行列、つまり、一つは x 座標についての行列そして一つは y 座標についての行列に分割され得る。これら二つの行列の各々は、行列における各行列要素についての二つの確率値を含んでいる。一つは、行列要素が値「1」を持つ確率であり、一つは、行列要素が値「0」を持つ確率である。位置コーディングパターンの各画像の処理は、このように、結果として二つの確率値行列 P_x および P_y を生じる。これらの行列がどのように決定され得るかは、図10～図19を参照して付録Bにより詳細に説明されている。付録Bの内容について、特に、本出願の提出の時点では公然と入手可能ではない、国際特許出願PCT/SE02/01246において、保護が請求されている。

【0061】

以下においては、上述されたAnnoto(アノト)位置コーディングパターンに基づく位置デコーディングが、図20におけるフローチャートを参照して説明されるであろう。

位置デコーディングは、望ましくはプロセッサーと、適切なプログラムコードとを用いて実施される。このプログラムコードは、図1におけるユーザーユニット2または外部ユニット3または何らかの他のユニットにおいて有効であり得る。

【0062】

プログラムコードに対する入力信号は、上述された二つの確率値行列、すなわちx座標およびy座標にそれぞれ関連するPxおよびPy（図18における行列34および35に對応する）を含んでいる。これらの確率行列に基づいて、x位置およびy位置が計算される（ステップ1800）。これがどのようにして実行されるかは、付録AおよびBにより詳細に説明されている。簡単には、行列PxおよびPyの各々について、各列および行におけるビット列が有する主数列において最も可能性のありそうな場所が決定される。このようにして取得された場所または位置番号（付録Bにおいては「列値」とも称する）から、差分数がX方向およびy方向についてそれぞれ決定される。差分数は、X方向およびy方向について差分数列の部分数列をそれぞれ形成する。部分数列は、各々差分数列において曖昧でなく決定された位置を有する。位置は、付録Aに説明された方法で決定され得る。x数列の位置は、x座標を決定し、そしてy数列の位置は、y座標を決定する。

【0063】

座標計算は、三つの場合に帰着し得る。二つの座標の計算が成功したか（1801）、または、一方の座標の計算が成功したが、他方の座標の計算が失敗したか（1802）、あるいは、両座標の計算が失敗したか（1803）、のいずれかである。

【0064】

もしも、座標計算の間にエラーが全く検出されなければ、これは成功したとみなされ、そしてその位置が容認された位置として格納される（ステップ1804）。位置のさらなるチェックがなされないならば、画像化された部分領域に関する全ての他の情報は無用とされ得る。

【0065】

もしも、一つの座標の計算が失敗したら、この座標についての情報が、マッチングに使用するために保存されると同時に、それについての計算が成功した座標は格納され、その位置のさらなるチェックがなされないならば、関連する情報は無用とされる（ステップ1805）。

【0066】

もしも、二つの座標の計算が失敗したら、もちろん二つの座標についての情報が保存される（ステップ1806）。

【0067】

保存された情報は、位置計算が基礎とする全情報、すなわち確率値行列Pxおよび/またはPy、あるいは、多少処理された形のこの情報、であり得る。例えば、各列が可能性のある位置番号の各々をとる確率値が計算され、保存される。その代わりに、例えば、位置番号の最も可能性のある組み合わせのみが、随意選択的に、関連付けられた確率と一緒に保存される。上述された情報の異なる組み合わせまたはサブセットも保存されるであろう。

【0068】

マッチングは、直ちにまたは後に、さらなる位置がデコードされたときに、行なわれ得る。

【0069】

さて、x座標の計算が成功したが、y座標のデコーディングが失敗したと仮定されたい。x座標は、y座標行列における行によって定義される全ての考えられる位置番号についての確率値も持つものとして保存される。

【0070】

マッチングが実行されるとき、最初に、マッチングが行われるべき領域が決定される（ステップ1807）。この場合、x座標が既知であるので、マッチングはy軸に沿って行われる必要があるのみである。y軸に沿う範囲は、以前におよびその後に容認されたy座

標に基づいて決定され得る。

【0071】

それから実際のマッチングが行われる。 x 座標は既知であるから、マッチング領域における各 y 座標について y 座標が対応する位置番号は既知である。これらの位置番号について格納される確率は、メモリーからフェッチされ、そして組み合わせ確率が、問題の位置番号についての確率の積として計算される(ステップ1808)。組み合わせ確率が、マッチング領域における全ての y 座標について計算されたとき、最も大きな組み合わせ確率がそれについて得られる y 座標が選択される(ステップ1809)。以前に格納された x 座標および選択された y 座標が、容認された位置として格納され(ステップ1810)、そして、その位置のさらなるチェックがなされないならば、マッチングのために保存された全ての情報は無用とされる。

【0072】

代わりの実施の形態においては、各行についての最も可能性のある位置番号のみが保存され得る。この場合、対応する位置番号の数は、各 y 座標について比較され、対応する位置番号の最も大きな数が得られる y 座標が選択される。この実施の形態は、保存された情報の量がより少ないという利点を有する。他方では、このことは、画像化された部分領域が、 y 方向における位置に関してマッチング部分領域に対応するより粗い確度に帰着する。

【0073】

さらに他の代わりの実施の形態において、最も可能性のある位置番号に加えてこれらについての確率も保存される。最も可能性のある y 座標は、それから対応する位置番号の数、これらについての組み合わせ確率および非対応位置番号についての対応する確率に基づいて選択され得る。

【0074】

もしも位置計算が、両座標について失敗するならば(ステップ1803)、例えば y 座標行列 P_y における行の各々についての最も可能性のある位置番号、および x 座標行列 P_x における列の各々についての最も可能性のある位置番号が、保存される。

【0075】

マッチングは、一度に一つの座標について行われる。マッチングが x 座標から始まると仮定されたい。最初に、マッチング領域が、 x 軸に沿う範囲として決定される(ステップ1811)。その領域のサイズは、上述されたのと同一のパラメーターに関して決定され得る。その後に、実際のマッチングが行われる(ステップ1812)。

【0076】

この場合、 y 座標は既知でないから、異なる x 座標がそれに対応する位置番号は既知ではなく、差分数のみである。考えられるマッチング方法は、格納された位置番号の間の差分を計算し、そしてこれらをマッチング領域における異なる x 座標についての差分とマッチングすることである。しかしながら、不適正な位置番号は、二つの不適正な差分を生じるから、これは最適な方法ではない。

【0077】

他の考えられる解決法は、マッチングされる x 座標について差分数を生じさせる位置番号の全ての考えられる組み合わせを検査することであろう。差分数にマッチングする位置番号の各組み合わせについて、位置番号についての確率値が選択され、掛け合わされる。最高の確率値がそれについて得られる x 座標が決定される。この解決法は、良好に機能するが、多くの計算を必要としそして従って長い時間がかかる。さらにまた、全ての位置番号についての確率値にアクセスすることが必要とされる。

【0078】

マッチングを最適化するために、マッチングは、その代わりとしてアルゴリズムを用いて実行され得る。このアルゴリズムは、符合した x 座標についての差分数に関して少なくとも1つの後続の位置番号にどのくらい多くの位置番号が対応するかを計算する。位置番号は右から左に考える。

【0079】

画像化される部分領域から決定される8つの位置番号 $p_0 \sim p_7$ がアクセス可能であると仮定されたい。これらの位置番号は、差分数の部分数列を構成する7つの差分数 $s_0 \sim s_6$ と符合される。差分数の部分数列は、画像化された部分領域からの情報と符合される×座標をコーディングする。差分数は、既知の位置コーディングパターンの構成内容によって決定され得る。

【0080】

そこで、アルゴリズム関数は次の通りである：

```
Numbermatch=0
For k=6 down to 0
  For i=0 to 6-k
    If p_{k+i+1}-p_k = sum(j=k:k+i) (s_j)
      Numbermatch=Numbermatch+1;
    最も内側のループへの割り込み;
  End
End
End
```

当該アルゴリズムが実行されるとき、変数Numbermatchは、差分数の部分数列における差分に関して少なくとも1つの後続の位置番号に対応する位置番号の数を含む。

【0081】

全てのマッチングが実施されたとき、Numbermatchが最大であった×座標が選択される。座標は、成功した座標として保存される。加えて、×座標についての位置番号は、以下に説明するように、もしかしたら後続する検証ステップにて使用されるべく、容易に決定され得る。

【0082】

さて、上述のアルゴリズムは、実施例を用いて説明されるであろう。×座標について以下の位置番号の部分数列Pを生じさせるシンボルを含む部分領域が画像化されていることを想定されたい。これらの位置番号は差分数の部分数列Dを定義する。さらに、部分領域は正確に画像化されず、正しくない部分数列 P^* が、画像化された部分領域からの情報に基づく位置計算において取得されると仮定されたい。最後に、主数列は63ビット長であり、そのことは主数列における位置を示す位置番号の間の差分が63を法として計算されることを意味する。

【0083】

| | | | | | | | | |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| P = | 2 3 | 1 2 | 5 4 | 4 3 | 7 | 1 8 | 1 1 | 3 5 |
| D = | 5 2 | 4 2 | 5 2 | 2 7 | 1 1 | 5 6 | 2 4 | |
| P^* = | 2 3 | 5 5 | 5 4 | 4 3 | 7 | 4 2 | 1 1 | 3 5 |

従って、位置番号1番(55、すなわち P^* における左から二番目の位置番号)および5番(42、すなわち P^* における左から六番目の位置番号)は、誤ってデコードされる。上述で説明されたアルゴリズムは、この実施例において次のように機能するであろう。

【0084】

Numbermatch = 0

【0085】

P 6 :

(35 - 11) mod 63 = 24。P 6 は符合。Numbermatch = 1

【0086】

P 5 :

$(11 - 42) \bmod 63 = 32$ (56 であるはず)。非符合。
 $(35 - 42) \bmod 63 = 56$ ($17 = (56 + 24) \bmod 63$ となっているはず)
 。非符合。P5は非符合。

【0087】

P4:

$(42 - 7) \bmod 63 = 35$ (11 となっているはず)。非符合。
 $(11 - 7) \bmod 63 = 4 = (11 + 56) \bmod 63$ 。P4は符合。Numbermatch = 2

【0088】

P3:

$(7 - 43) \bmod 63 = 27$ 。P3は符合。Numbermatch = 3

【0089】

P2:

$(43 - 54) \bmod 63 = 52$ 。P2は符合。Numbermatch = 4

【0090】

P1:

$(54 - 55) \bmod 36 = 62$ (42 となっているはず)。非符合。
 $(43 - 55) \bmod 63 = 51$ ($31 = (42 + 52) \bmod 63$ となっているはず)
 。非符合。
 $(7 - 55) \bmod 63 = 15$ ($58 = (42 + 52 + 27) \bmod 63$ となっているはず)。非符合。
 $(42 - 55) \bmod 63 = 50$ ($6 = (42 + 52 + 27 + 11) \bmod 63$ となっているはず)。非符合。

$(11 - 55) \bmod 63 = 19$ ($62 = (42 + 52 + 27 + 11 + 56) \bmod 63$ となっているはず)。非符合。
 $(35 - 55) \bmod 63 = 43$ ($23 = (42 + 52 + 27 + 11 + 56 + 24) \bmod 63$ となっているはず)。非符合。

P1は非符合。

【0091】

P0:

$(55 - 23) \bmod 63 = 32$ (52 となっているはず)。非符合。
 $(54 - 23) \bmod 63 = 31 = (52 + 42) \bmod 63$ 。P0は符合。

Numbermatch = 5

【0092】

したがって、アルゴリズムは、5つの位置番号が、差分数の部分数列における差分に関して少なくとも1つの後続の位置番号に対応するという事実に到達する。対応する位置番号の数は、符合確率の度合いとして用いられる。

【0093】

×座標がどのように符合し得るかの上述の実施例は、位置計算が両座標について失敗した場合にももちろん機能し、その上、×座標の代わりにy座標で始まる場合にも機能する。

【0094】

第一の座標がマッチングによって決定されたとき、第二の座標は、位置計算が単に一つの座標について失敗した場合について、上述されたのと同様の方法で、または、二つのうちの第一の座標について失敗したのと同様の方法で、決定され得る。

【0095】

上述のアルゴリズムは、同時に評価されるいくつかのマッチングによって、また、より効率的に処理される差分数の合計によって、最適化され得る。アルゴリズムは、次の通りである。そこでは、各マッチングについての s_{s_i} が $\text{sum}(s_0 \dots s_i)$ に等しい。

【0096】

```

For each matching (各符合探索)
  For k=0 to 6
    For i=0 to 6-k
      dp=pk+i+1-pk
      If dp==ssi
        Numbermatchk=Numbermatchk+1
        最も内側のループへの割り込み;
      End
    End
  End

```

Numbermatch₀が、これまで最大であるかどうかをチェックする。最大である場合、Numbermatch₀の値および対応するx座標を記憶する。

```

  For j=0 to 5
    Numbermatchj=Numbermatchj+1
  End

```

```

Numbermatch6=0
d = 差分数列における次の先行する差分数
For j=6 to 1
  ssj=ssj-1+d
End
ss0=d
End

```

【0097】

マッチングは、有利には、図20において断続線によって示される検証ステップ（ステップ1815）で終了され得る。実際、マッチングのアルゴリズムは、常に、以前に容認された位置に近接する位置を見つけるから、エラーの増大される危険が存在する。このことは、符合した位置が正しいか否かを査定するために、容認された位置からの距離を調べることができないことを意味している。検証ステップは、画像化された部分領域において有効な全ての情報を用いて、取得された位置についての確率を計算するステップを有していてもよい。通例、これは、位置計算に使用される8×8シンボルよりもかなり多いシンボルを含んでいる。例えば、それは16×16シンボルを含んでいても良い。これらのシンボルの各々について、確率値（付録Bにおいて値確率とも称される）は、シンボルがとり得る値の各々について決定されている。一対の座標が、最善の8×8領域を用いて決定されているとき、これら8×8シンボルの値は既知であり、例を挙げれば、対応する8×8ビットのxおよびy行列におけるそれぞれ列および行におけるビット数である。列および行は二進数の循環主数列の部分数列を構成するから、差分数列を知って、より大きな16×16領域において列および行がどのように続いているか、が決定される。16×16領域におけるビット数についての確率値を用いて、16×16領域全体の確度を計算し、16×16領域についての全体の確率が最終的に容認されるデコードされた位置について充分に大きいかどうかを決定するしきい値とこの確度とを比較することができる。

【0098】

全体の方法が図21におけるブロック図に要約されている。最初に、位置計算1901が行われる。位置計算が成功すれば、続いて位置の検証1902が行われる。位置の検証

が成功すれば、その位置は容認される。もしも位置計算が失敗すれば、マッチング 1903 が行われる。位置計算が成功した位置についても検証が失敗した場合には、マッチングが行われる。マッチングは、常に位置を生じさせる。続いてこの位置の検証 1902 が行われる。マッチングによって取得された位置についての検証が成功すれば、その位置は容認される。そうでなければ、位置デコーディングは失敗し、画像化された部分領域について位置が記録されることはない。位置計算が成功し、直後の検証もマッチング後の検証も失敗したが、計算された位置とマッチングで得られた位置が等しいという特別な場合には、その位置が容認され得る。

【0099】

位置コーディングパターンは、位置デコーディングのための異なるタイプのデバイスを用いて読み取られ且つデコードされ得る。そのようなデバイスは、位置コーディングパターンのディジタル表現を作るためのセンサーと、ディジタル表現にて位置コーディングパターンを識別し且つ同一物をデコーディングするための信号処理ユニットとを有することができる。この信号処理ユニットは、メモリーおよび適切なプログラムまたは特有のハードウェアまたはディジタルおよび/またはアナログ回路ならびにそれらの適切な組み合わせからなるプロセッサーから構成し得る。

【0100】

センサーは、マークの画像が、白黒で、グレースケールでまたはカラーで取得されるように、位置コーディングパターンを画像化するのに適するいかなる種類のセンサーでもあり得る。センサーは、いかなる適切な波長帯域においても電磁的放射に感應する固体の単一またはマルチチップデバイスであり得る。例えば、センサーは、CCD(電荷結合素子)エレメント、CMOS(相補型金属酸化物半導体)エレメント、またはCIS(電荷注入デバイス)エレメントを含んでいても良い。その代わりに、センサーは、マークの磁気的な性質の検出のための磁気センサーアレイを含んでいても良い。さらにまた、センサーは、マークのいかなる化学的、音響的、容量的、または誘導的性質の画像を形成すべく設計されていても良い。

【0101】

位置デコーディングのためのデバイスの例が、図22に示されている。デバイスは、図1におけるユーザーユニットを構成し得る。それは、ペンとほぼ同一の形状を有するケーシング 2011 を備えている。ケーシングの一つの短手側に開口部 2012 が設けられている。位置決定がその上で行われる面に対して接触しまたはそこから短い距離に配置されることを意図している。

【0102】

ケーシングは、本質的に、光学部品、電子的回路部品、および電力供給源を収容している。

【0103】

光学部品は、画像化されるべき面を照明するための発光ダイオード 2013 と、二次元画像を記録するためのCCDまたはCMOSセンサーのような、光検出エリアセンサー 2014 とを備えている。デバイスは、ミラーおよび/またはレンズ系のような、光学系を備えていても良い。発光ダイオードは、赤外発光ダイオードとすることができます、そしてセンサーは、赤外光に対して感應するものとすることができます。

【0104】

デバイスに対する電力供給は、ケーシングにおける分離された区画に搭載されるバッテリー 2015 から取得される。電力供給は、外部電力源(図示されていない)からケーブルを経由して取得することもできる。

【0105】

電子的回路部品は、センサーからの画像を読み取り且つこれらの画像に基づいて位置計算、マッチングおよび検証を行うべくプログラムされたプロセッサー、ならびにワーキングメモリーおよびプログラムメモリーを有するプロセッサーユニット 2016 を備えている。

【0106】

さらにまた、デバイスは、この実施の形態においては、その助けを借りてその上で位置決定が行われるべき面上に通常の顔料ベースの筆跡を書くことを可能とするペン先2017を備えている。ペン先2017は、それが使用されるか否かをユーザーがコントロール可能するために、延長可能で且つ引き込み格納可能とすることができる。ある特定の応用においては、デバイスは、全くペン先を持つ必要がない。

【0107】

適切なことに、顔料ベースの筆跡は、赤外光に対して透明なタイプからなりそしてマークは赤外光を吸収する。赤外光を放出する発光ダイオードおよび赤外光に感應するセンサーを用いることによって、上述された筆跡がパターンと緩衝することなしに、パターンの検出が、行われる。

【0108】

さらにまた、デバイスは、それを用いてユーザーがデバイスを付勢し且つコントロールするボタン2018を備えていても良い。デバイスへそしてデバイスからの情報の無線伝達のために、例えばIR(赤外)光、ラジオ波または超音波を用いる、トランシーバー2019を備えていても良い。デバイスは、位置または記録された情報を示すためのディスプレイ2020を備えていても良い。

【0109】

デバイスは、異なる物理的なケーシング、センサーおよび位置コーディングパターンの画像を取り込むためにそして、信号処理ユニットと位置コーディングを実行するために必要であるその他の構成要素が第二のケーシング内に配置されている場合に、それらを第一のケーシング内に配置された第二のケーシングに伝達するために必要な他の構成要素に分割され得る。

【0110】

付録A (APPENDIX A)

位置コードは、一以上の次元における位置をコード化するために用いられる。説明を単純化するために、位置コーディングは、まず、この例においてはx-方向における、第一次元においてのみ考慮される。循環主数列が、この方向におけるコーディングのために用いられ、該循環主数列は、所定長さの各部分列の前記循環主数列における場所が、明確に決定される性質を有している。この例においては、前記所定長さは6である。したがって、もしも六つの連続する数が、前記循環主数列におけるいずれかの場所からとられたら、これら六つの数が、この順序で前記循環主数列においてただ一度だけ生ずるであろう。該性質は、もしも前記主数列の末端が該主数列の始端に連結されている場合にも適用される。前記主数列は、それ故に循環と呼ばれる。この例においては、二進主数列が使用される。もしも、六つの数を有する部分列の場所が、明確に決定されるならば、前記主数列は、したがって、 $2^6 = 64$ の最大長さを持つことができ且つ長さ6の前記部分列は、前記主数列における場所0~63を持つことができる。しかしながら、もしも、長さ63の主数列が選択されると、以下に説明されるであろうように、改善されたエラー訂正特性が得られる。以下においては、したがって、主数列の長さは63で且つそれ故それが0~62の範囲におけるただ一つの場所を定義すると仮定される。前記主数列のこの長さによれば、各部分列を前記主数列における場所に変換し且つその逆にするテーブルを持つことが実際に可能である。

【0111】

前記主数列の開始が次の通りであると仮定されたい：

0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0...

そのとき、例えば、前記主数列において、部分列0, 0, 0, 0, 0, 0は、明確な場所0を有し、部分列1, 1, 1, 1, 1, 0は、明確な場所9を有し、そして部分列1, 1, 1, 0, 1, 0は、明確な場所11を有する。

【0112】

前記位置コーディングは、前記循環主数列の異なる循環または循環移動を利用すること

に基づいている。 x -方向における位置をコード化するために、前記主数列が、プリントアウトされまたは何か他の方法で配列され、面を横切る、すなわち位置がコード化されるべき方向に直交する y -方向における、列について種々の方法で循環されまたは循環的に移動される。前記主数列は、もしも y -方向において、前記主数列の前記長さに対応するものよりも多くの位置がコード化されようとする場合に必要とされる、同一の列について繰り返しプリントアウトされ得る。それから、前記主数列の同様の循環が、すべての繰り返しに使用される。しかしながら、異なる循環は異なる列に使用され得る。

【0113】

図4は、七つの列 $x \sim x + 6$ を有する一枚の紙A100を概略的に示している。前記主数列の異なる循環における最初の数のみがプリントアウトされる。全体の主数列は、ボックスABによって概略的にマークされている。加えて、図4においては、六つの数の第一の部分列Sが、破線を用いて描かれたボックスによって各列においてマークされている。上述されたように、前記部分列Sの各々は、前記主数列における明確に決定された場所を有する。

【0114】

隣接する列の各対は、差分数dを定義する。差分数dは、それぞれの列における第一の部分列の主数列における場所の間の差分によって与えられる。もしも、その代わりに、各列における一ステップ下の部分列の場所の間の差分がとられたら、結果は、場所が同様の方法で転置されるので、同様となるであろう。したがって、差分数dは、列におけるどの「高さ」において、前記主数列における部分列の場所が比較されるかにかかわりなく、同様となるであろう。したがって、各対の列について、差分数dは、 y -方向において一定である。

【0115】

列 $x + 5$ における主数列の第一の部分列は、0, 0, 0, 0, 0, 0であり、それは前記主数列における場所0に対応する。列 $x + 6$ における主数列の第一の部分列は、1, 0, 0, 0, 1, 1であり、それは前記主数列における場所57に対応する。これらの列の間の差分または循環移動は、したがって57であり、それ故 $d(x + 5) = 57$ である。前記差分数が前記主数列の長さを法として決定されることが指摘されるべきである。

【0116】

このような方法でコード化され得る差分数は、0 ~ K - 1の範囲内にあり、ここでKは前記主数列の長さで、この場合K = 63であり、したがって、それによって0 ~ 62の範囲内における差分数をコード化することが可能となる。前記主数列の異なる循環を選択することによって、以下において一次数列または一次差分数列と呼ばれる、差分数の列を作ることができ、それは、所定長さの各部分列は、一次数列において明確に決定される場所を有する。この例においては、前記所定長さは5である。五つの差分数からなる各部分列は、前記一次差分数列における明確に決定する場所を有し、このことは x -方向における位置をコード化するのに用いられ得る。図4において、部分列 $S(x) \sim S(x + 5)$ は、前記一次数列の部分列を与える五つの差分数 $d(x) \sim d(x + 4)$ をこのようにしてコード化するであろう。

【0117】

実際に前記部分列は、それらの明示的な値によっては正常にプリントアウトされないが、グラフィカルなコード化によって正常にプリントアウトされることも指摘されるべきである。前記主数列の二進数の列は行列を形成し、それは以下においては x -行列と呼ばれるであろう。

【0118】

もしも、前記主数列の長さがKであるならば、前記一次差分数列におけるベースはKに等しくなり且つその最大長さは、 K^5 、すなわちこの場合 63^5 になるであろう。しかしながら、部分列を前記一次差分数列における場所に変換するためのテーブルを使用すること、そして非常に大きいベースについてその逆をすることは、実際には可能ではない。しかしながら、もしも、前記ベースが管理可能なテーブルを使用できるようにするために縮

小さされれば、コード化可能な位置の数が減小されるであろう。

【0119】

この問題は、特定の x -座標に対応する一次差分数列における部分列およびその逆を決定するために、前記位置コードをコーディングし且つデコーディングするときに、以下においては、二次差分数列または二次数列と呼ばれる、より短い差分数列の一次差分数列を作ることによって、そして、そのために前記部分列およびそれらの場所が小さなテーブル内に配置され得る、前記二次差分数列を用いることによって、解決される。

【0120】

前記二次差分数列は、次のようにして決定され得る：

第一に、差分数は、異なる差分数の個数が少なくとも二つの要素に因数分解され得るような範囲内にのみ許可される。長さ63の主数列を有する例においては、差分数は、長さ54の範囲内でのみ許可される。数54は、実際には、 $2 \times 3 \times 3 \times 3$ として因数分解され得る。その代わりに、60の差分数、すなわち $5 \times 3 \times 2 \times 2$ の差分数が、選択され得るが、テーブルが可能な限り小さくなるようにするために、可能な限り小さな要素を与える数を選択することが好都合であるかもしれない。

【0121】

第二に、二次差分数列の同一の数は異なる差分数の数が因数分解され得る要素の数として形成される。この例においては、それ故、二次差分数列の数は、4に等しい。

【0122】

加えて、我々は、各要素に、それぞれの二次差分数列においてベースを形成させる。この例においては、それ故、我々は、ベース2を有する一の二次差分数列およびベース3を有する三つの二次差分数列を得る。

【0123】

したがって、前記二次差分数列の差大長さは、もしも前記二次差分数列が長さ5の部分列が該二次差分数列において特有の場所を有するはずであるという特性をも有するとするならば、それぞれ32および243である。前記二次差分数列のそのような長さによれば、部分列を場所およびその逆に変換することが、実際に可能である。もしも、その代わりとして、我々が60の異なる数を使用すべく選択したならば、我々はベース5を有する二次差分数列を持つであろうし、そしてそれ故、かなりより多くのメモリを使用するテーブルを与える、3125の最大長さが、しかしながら、なお管理できる。

【0124】

最後に、前記二次差分数列の長さは、該長さが対をなして相対的に素であるように選択される。このことは、各対をなす二次差分数列について、一の差分数列の長さが、前記第二の差分数列の長さと共に通のどのような要素も持たないようにすることを意味する。このことは、加えて、もしも前記差分数列の各々が繰り返されるならば、各二次差分数列からの部分列の同一の組み合わせは、 $L = l_1 \times l_2 \times \dots \times l_m$ の場所の後まで生じないであろうことを意味し、そこでは、 l_1 は、二次差分数列1の長さであり、 l_2 は、二次差分数列2の長さであり、そして最後の二次差分数列1の長さである l_m まで、同様である。もしも、二つの二次数列だけしかなければ、もちろん、 $L = l_1 \times l_2$ である。

【0125】

このことは、図5に概略的に図解されている。最上部には、ここでは単にXによって示されている差分数による一つの長い列において延びる一次数列PDが示されている。下部には、二次数列A1～A4が概略的に示されている。垂直線は数列が再開する場所を示している。一次数列の部分列および対応する二次数列の部分列は、破線によって示されている。部分列は、二次数列における異なる場所に対応することがわかる。

【0126】

二次差分数列の部分列の組み合わせによる一次差分数列の部分列の表現は、この例においては、全単射である。これは、しかしながら、必要なことではない。

【0127】

第二次元における、例えばこの場合y-方向における、位置コーディングは、第一次元

における位置コーディングと同様の原理に従って実行され得る。前記主数列は、それから、面上の行についての、すなわち図4におけるx - 方向についての、異なる循環移動によって配列される。差分数は、隣接する行の間で定義され且つこれら差分数は、二次差分数列から作られ得る一次差分数列を形成する。第二次元における他の主数列、他の一次差分数列および他の二次差分数列を使用することが可能である。前記種々の数列におけるベースも、前記第一次元においてコーディングのために使用されたそれらのもの以外とすることができる。この例においては、しかしながら、同一の列が、x - 方向とy - 方向との両方に使用される。x - 方向におけるコーディングに対応する方法において、y - 方向におけるコーディングは、結果として、行の値が主数列の二進値からなる行列を得る。

【0128】

もしも、x - 行列およびy - 行列が重ね合わされたならば、結果として得られるXY - 行列における各点について、x - 行列からの1ビットとy - 行列からの1ビットとが存在するであろう。それ故、各点において、次の四つのビット組合せが、取得され得る：0, 0 ; 0, 1 ; 1, 0 ; および1, 1。これらの異なる組み合わせは、例えばドットA30が、ラスタにおける公称点A31からの四つの方向のいずれか一つの方向に変位され、各公称点は、ラスタにおける線A32間の交差に対応する、図6a～図6dに示された方法によって、面上にグラフィカルにコード化され得る。このタイプのグラフィカルなコーディングは、国際特許出願WO01/26033号に非常に詳細に説明されており、そこでは、しかしながら、実際に潜在する位置コーディングが異なる方法にて実行される。この例において使用されるコーディングは次の通りである：

| 値 | 変位 | ビットの対 |
|---|----|-------|
| 1 | 右 | 0, 1 |
| 2 | 上 | 0, 0 |
| 3 | 左 | 1, 0 |
| 4 | 下 | 1, 1 |

ラスタは、仮想的であり得て、その場合、それ故、位置コードにて面上を明示的にプリントアウトしない。

【0129】

図7は、位置コード410を有する一枚の紙400の形態における製品の一部を概略的に示している。図7における位置コードは、例えばデジタル化された手書きに適するバージョンに関連して非常に大きく拡大されていることは、強調されるべきである。加えて、ラスタ420は、図7にマークされている。しかしながら、これは、通常はプリントアウトされない。

【0130】

もちろん、他のタイプのグラフィカルなコーディングが、異なるサイズの4ドット、異なる形状を有する四つのマーク、または異なる回転を有する四つのマークのような、四対のビットについて選択され得る。

【0131】

位置コーディングが、主数列における異なる部分列の場所の間の差分に基づいており、且つ同様の差分が、部分列の異なる対を用いて得られ得るので、一次差分数列が、種々の方法において達成され得る。より詳細には、一次差分数列の63の異なる「バージョン」が、その部分列、すなわち主数列のどの回転が、x - 行列における非常に最初の列(x =

0) および y -行列における非常に最初の行 ($y = 0$) を始めるべく選択されたか、に依存して、達成され得る。したがって、 x -方向および y -方向へのコーディングのために同一の一次差分数列を利用する位置コードの $K \times K$ (この例では $63 \times 63 = 3969$) 個の異なる「バージョン」を生成することが可能である。位置コードのこれらの異なるバージョンは以下においてセクションと呼ばれる。 x -方向における位置コードについてのセクション数 s および y -方向における位置コードについてのセクション数 y_s は、それぞれ x -方向および y -方向についての付加的な座標として示され得る。

【0132】

上の例においては、主数列を用いて理論的にコード化され得る差分数のサブセットのみが、より詳細には、63の可能性のある差分数から54のみが、使用されている。理論的に可能性のある差分数の部分領域は、種々の方法で選択され得る。この例においては、(前記主数列の長さを法とした)ゼロに近い差分数が、位置コードにおける所望されていない規則性を作り得る。それ故、この例においては、差分数は、範囲 (5, 58) 内に選択される。このことは、一次数列における部分列と二次数列における部分列との間の変換に関して考慮されなければならず、そしてそれは以下において説明されるであろう。

【0133】

今、位置コードが面上に配置されると仮定されたい。読み取りデバイスは、少なくとも 6×6 ドットに対応する位置コードのサブセットを検出するエリアセンサを持ち得る。読み取りのとき、前記読み取りデバイスは、前記位置コードに関連して種々の方法で回転されて保持され得る。理論上で位置コードはそれが 0, 90, 180 または 270 度にわたって回転されようとされまいと同様に見えるので、位置コードのイメージは、位置コードと読み取りデバイスとの間の回転をそれ自体で明らかにすることはない。位置コードが回転されるとき、各ドットの変位の方向は、しかしながら、変化され、それが、今度は、変化されるドットの変位によってコード化されたビットの対に導く。どのように変化が生ずるかは、変位とビットの対の間の変換に依存する。以下においては、上述されたように変換が実施されると仮定されている。三つの異なるケースが存在する。

【0134】

1) 位置コードのイメージの 180 度の回転によって、非回転位置コードにおける x -および y -位置をコード化する循環主列は、イメージに基づく位置をデコードする試みがなされるとき、後方を読み取るであろう。上述された変位の組み合わせおよびビットの対によって、デコードされたビットは、位置コードが上下逆に読み取られることから、反転されるであろう。

【0135】

2) 時計回り方向における 90 度の回転によって、非回転行列における x -位置をコード化するビット列は、回転行列における y -位置をデコードする試みがなされるとき、後方を読み取るであろう、そして前記ビットは反転されるであろう。

【0136】

3) 時計回り方向における 270 度の回転によって、非回転行列における y -位置をコード化するビット列は、回転行列における x -位置をデコードする試みがなされるとき、後方を読み取るであろう、そして前記ビットは反転されるであろう。

【0137】

このことは、もしも、正しい向きで正立状態にある非回転行列における部分列が、反転されることなく且つ循環主数列において後方にあるならば、90, 180 および 270 度の回転が検出され得ることを意味する。もしもそのような条件が前記主数列について満たされるならば、このことは、その長さがずいぶん減小され、今度はより少ない位置がコード化され得る。

【0138】

この問題を解決するため、我々は、その代わりに、前記循環主数列のより長い部分列について、条件が満たされるようにさせる。このことは、実際の位置決定のために要求されるよりも長い部分列が、読み取られなければならないことを意味する。ある場合において

は、そのようなより長い部分列は、読み取りの間に既に利用できる。もしも、位置についての座標をコード化する位置コードの最小のサブセットが二次式であるならば、この例における場合にそうであるように、読み取りデバイスが、位置コードに関係して45度にわたって回転されたときでさえも、それが最小のサブセット全体をも含むように大きな位置コードの一部がいつも読み取られなければならない。それ故、位置決定のために要求されるものよりも長い少なくとも一つの部分列がいつも含まれる。このことは、図8に図解されており、そこでは、ドットに対応する各正方形50およびグレーに影付けされた正方形が、二つの余分のドットと共に少なくとも一つの行および一つの列を常に読み取ることができることを図解している。

【0139】

現在の例において、そこでは位置決定のための主数列の部分列は、長さ6を有し、長さ8の部分列が回転の検出のために使用され得る。これら8ビット長の部分列は、したがって、主数列内において後戻りおよび反転を生じることがないという特性を有する。この特性は、64ビット長の主数列について達成され得ず、それが前記長さがその代わりに63として選択される理由である。

【0140】

上述された回転のうちの二つ、すなわち90および270度において、正しい方向に回転する一列、そして誤った方向に回転する一列が読み取られるであろう。回転検出のための上述された方法によれば、第一の二つの読み取り方向(xおよびy)だけによって、回転エラーが検出されるであろう。該第二の読み取り方向において、正しく回転されたイメージの第一の読み取り方向における位置をコード化する循環移動がみられるであろう。

【0141】

時計回り方向における90度回転の場合において、正確に回転されたイメージのy-座標をコード化する循環移動は、x-方向についてデコーディングするときにみられるであろう。したがって、それは、みられるy-方向の一次差分数列の部分列であるが、誤った方向に回転される(左から右への代わりに右から左へ移動する、そしてそれはx-方向の一次差分数列の部分列の場合である)。

【0142】

対応する方法において、x-方向の一次差分数列の部分列は、270度回転の場合にy-方向にデコーディングするときに、誤った方向に回転するであろうことがわかる。

【0143】

なぜ部分列が誤った方向に回転するよう見えるかという理由は、x-およびy-方向の一次差分数列の相対的な方向性である。全体の位置コーディングパターンの縁部から始まり、y-方向の一次差分数列が、反時計回り方向に移動する間に、x-方向の一次差分数列が、時計回り方向に移動する。もしもそれらが同一の方向に走っていたら、それらは誤った方向には回転しないが、上述のケースにおいて正しい方向に回転する。

【0144】

一次差分数列を反対方向に走らせることには多数の利点がある。一次差分数列の部分列が正しい回転をしているケースにおいては、「正しい」座標がデコードされるが、誤った方向においてである。この座標は、デコーディングに要求される一次差分数列からの六つの循環移動に対応するだけでなく、周囲の移動がデコードされる座標が正しいことを確認するであろう。一次差分数列が互いに反対方向に走っているケースにおいては、一次差分数列からの六つの循環移動もデコードされるが、誤った順序で、結果として誤って回転された一次差分数列の周囲部分に統計的に対応しないであろう座標を生ずる。この事実は、結果としてエラー検出についてより良い可能性を生ずる。

【0145】

他の利点は、もしも一次差分数列が、互いに反対の方向に走っていれば、連続的なエラー、すなわち複数の連続するイメージが隣接する位置にデコードされるエラー、の危険が少ないとある。より詳細には、誤った方法で回転される一次差分数列における六つの循環移動の異なる隣接する組は、隣接する座標をコード化しない傾向がある。連続的なエ

ラーは、そこでは正しくない座標の広がりが大きく、他のエラーよりも検出することが一層困難である。

【0146】

回転の検出のために使用されるのと同一の基本原理が、エラー訂正のために使用され得る。主数列は、例えば、位置決定のために要求されるよりも長いいくつかの所定長さの部分列が、主数列内でビット反転されて生じないようにして選択され得る。もしも、そのようにより長い部分列における一を除く全てのビットが確実性をもって検出され得るならば、不正なビットは、訂正され得る。

【0147】

位置コードのそのようなエラー検出およびエラー訂正特性は、主数列の理性的な選択によってかなり改善され得る。さらなる改善が、二次差分数列の選択によって達成され得る。

【0148】

上述の例においては、主数列および二次差分数列が、次のような方法で選択されている。

【0149】

主数列 M :

```
0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 1 , 1 , 1 , 1 , 1 , 0 , 1 , 0 , 0 , 1 , 0 ,
0 , 0 , 0 , 1 , 1 , 1 , 0 , 1 , 1 , 1 , 0 , 0 , 1 , 0 , 1 , 0 , 1 , 0 , 1 , 0 , 0 , 0 ,
1 , 0 , 1 , 1 , 0 , 1 , 1 , 0 , 0 , 1 , 1 , 0 , 1 , 0 , 1 , 1 , 1 , 1 , 1 , 0 , 0 ,
0 , 1 , 1
```

二次差分数列 :

```
A 1 = 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 2 , 0 , 1 , 0 , 0 , 1 , 0 , 1 , 0 ,
, 0 , 2 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 1 , 2 , 0 , 0 , 1 , 0 , 2 , 0 , 0 , 2
, 0 , 2 , 0 , 1 , 1 , 0 , 1 , 0 , 1 , 1 , 0 , 2 , 0 , 1 , 2 , 0 , 1 , 0 , 1 , 1 , 2
, 0 , 2 , 1 , 0 , 0 , 1 , 1 , 1 , 0 , 1 , 1 , 1 , 0 , 2 , 1 , 0 , 1 , 0 , 1 , 0 , 2
, 1 , 1 , 0 , 0 , 1 , 2 , 1 , 0 , 1 , 1 , 2 , 0 , 0 , 0 , 2 , 1 , 0 , 2 , 0 , 2
, 1 , 1 , 1 , 0 , 0 , 2 , 1 , 2 , 0 , 1 , 1 , 1 , 2 , 0 , 2 , 0 , 0 , 1 , 1 , 2
, 1 , 0 , 0 , 0 , 2 , 2 , 0 , 1 , 0 , 2 , 2 , 0 , 0 , 1 , 2 , 2 , 0 , 2 , 0 , 2
, 2 , 1 , 0 , 1 , 2 , 1 , 2 , 1 , 0 , 2 , 1 , 2 , 1 , 1 , 0 , 2 , 2 , 2 , 1 , 2 , 1
, 2 , 0 , 2 , 2 , 0 , 2 , 2 , 2 , 0 , 1 , 1 , 2 , 2 , 2 , 1 , 1 , 0 , 1 , 2 , 2 , 2
, 2 , 1 , 2 , 0 , 0 , 2 , 2 , 1 , 1 , 2 , 1 , 2 , 2 , 1 , 0 , 2 , 2 , 2 , 2 , 2
, 0 , 2 , 1 , 2 , 2 , 2 , 1 , 1 , 1 , 2 , 1 , 1 , 2 , 0 , 1 , 2 , 2 , 1 , 2 , 2
, 0 , 1 , 2 , 1 , 1 , 1 , 1 , 2 , 2 , 2 , 0 , 0 , 2 , 1 , 1 , 2 , 2 , 2
A 2 = 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 2 , 0 , 1 , 0 , 0 , 1 , 0 , 1 , 0 ,
, 1 , 1 , 0 , 0 , 0 , 1 , 1 , 1 , 1 , 0 , 0 , 0 , 1 , 1 , 0 , 1 , 0 , 0 , 2 , 0 , 0
, 0 , 1 , 2 , 0 , 1 , 0 , 1 , 2 , 1 , 0 , 0 , 0 , 2 , 1 , 1 , 1 , 0 , 1 , 1 , 1
, 0 , 2 , 1 , 0 , 0 , 1 , 2 , 1 , 2 , 1 , 0 , 1 , 0 , 2 , 0 , 1 , 1 , 0 , 2 , 0
, 0 , 1 , 0 , 2 , 1 , 2 , 0 , 0 , 0 , 2 , 2 , 0 , 0 , 1 , 1 , 2 , 0 , 2 , 0 , 0
, 2 , 0 , 2 , 0 , 1 , 2 , 0 , 0 , 2 , 2 , 2 , 1 , 1 , 0 , 0 , 2 , 1 , 0 , 1 , 1 , 2
, 1 , 0 , 2 , 0 , 2 , 2 , 1 , 0 , 0 , 2 , 2 , 2 , 1 , 0 , 1 , 2 , 2 , 0 , 0 , 2
, 1 , 2 , 2 , 1 , 1 , 1 , 1 , 2 , 0 , 0 , 1 , 2 , 2 , 2 , 1 , 2 , 0 , 1 , 1 , 1
, 2 , 1 , 1 , 2 , 0 , 1 , 2 , 1 , 1 , 2 , 2 , 0 , 2 , 2 , 2 , 0 , 1 , 1 , 2 , 2
, 2 , 2 , 1 , 2 , 1 , 2 , 2 , 0 , 1 , 2 , 2 , 2 , 0 , 2 , 0 , 2 , 1 , 1 , 2 , 2
, 1 , 0 , 2 , 2 , 0 , 2 , 1 , 0 , 2 , 1 , 1 , 0 , 2 , 2 , 2 , 2 , 0 , 1 , 0 , 2
, 2 , 1 , 2 , 2 , 2 , 1 , 1 , 2 , 1 , 2 , 0 , 2 , 2 , 2 , 2 ,
A 3 = 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 1 , 1 , 1 , 1 , 0 , 0 ,
, 1 , 0 , 1 , 0 , 1 , 1 , 0 , 1 , 1 , 0 , 1 ,
A 4 = 0 , 0 , 0 , 0 , 0 , 1 , 0 , 2 , 0 , 0 , 0 , 0 , 2 , 0 , 0 , 2 , 0 , 1 , 0
, 0 , 0 , 1 , 1 , 2 , 0 , 0 , 1 , 2 , 0 , 0 , 2 , 1 , 0 , 0 , 0 , 2 , 1 , 1
```

```
, 2 , 0 , 1 , 0 , 1 , 0 , 0 , 1 , 2 , 1 , 0 , 0 , 1 , 0 , 0 , 2 , 2 , 0 , 0 , 0
, 2 , 2 , 1 , 0 , 2 , 0 , 1 , 1 , 0 , 0 , 1 , 1 , 1 , 0 , 1 , 0 , 1 , 1 , 0 , 1
, 2 , 0 , 1 , 1 , 1 , 1 , 0 , 0 , 2 , 0 , 2 , 0 , 1 , 2 , 0 , 2 , 2 , 0 , 1 , 0
, 2 , 1 , 0 , 1 , 2 , 1 , 1 , 0 , 1 , 1 , 1 , 2 , 2 , 0 , 0 , 1 , 0 , 1 , 2 , 2
, 2 , 0 , 0 , 2 , 2 , 2 , 0 , 1 , 2 , 1 , 2 , 0 , 2 , 0 , 0 , 1 , 2 , 2 , 0 , 1
, 1 , 2 , 1 , 0 , 2 , 1 , 1 , 0 , 2 , 0 , 2 , 1 , 2 , 0 , 0 , 1 , 1 , 0 , 2 , 1
, 2 , 1 , 0 , 1 , 0 , 2 , 2 , 0 , 2 , 1 , 0 , 2 , 2 , 1 , 1 , 1 , 2 , 0 , 2 , 1
, 1 , 1 , 0 , 2 , 2 , 2 , 0 , 2 , 0 , 2 , 2 , 1 , 2 , 1 , 1 , 1 , 1 , 2 , 1
, 2 , 1 , 2 , 2 , 2 , 1 , 0 , 0 , 2 , 1 , 2 , 2 , 1 , 0 , 1 , 1 , 2 , 2 , 1 , 1
, 2 , 1 , 2 , 2 , 2 , 1 , 2 , 0 , 1 , 2 , 2 , 1 , 2 , 0 , 2 , 2 , 0 , 2 , 2 , 1
, 1 .
```

【0150】

二次差分数列は、以下に述べる特別なエラー訂正特性を有している：

主数列の部分列の一つが、正しくなくデコードされ、そしてそれはデコードされた主数列における正しくない場所に導くと仮定されたい。各場所は二つの隣接する差分数を計算するために使用されるので、これらは、両方とも正しくないデコーディングによって影響されるであろう。もしも、差分数のいずれか一方が、使用される差分数範囲(5, 58)の外側にあれば、エラーは直ちに検出されるであろう。もしも、これがそのようなケースでなければ、しかしながら、一次差分数列の部分列が結果として生ずる四つの二次差分数列の部分列の組合せにおける少なくもと一つの部分列が、二つの隣接する位置にゆがめられるであろう。二つの第一の二次差分数列A1およびA2の両方がベース3を有し、且つ使用されない差分数の数は、 $9 = 3 \times 3$ であるので、これら二つの列のいずれか一つにおけるゆがみは、二つの影響された数の合計は、3を法としたいつも同一の値を有するという特性を有している。

【0151】

二つの二次数列A1およびA2は、単一の不適正な場所のデコーディングによって引き起こされる部分列の14の可能性のある歪みの多くとも一つにおける長さ7の各部分列について二次数列内に見出されるであろうという特性を有している。第三の二次数列A3は、加えて、単一の不適正な場所のデコーディングによって引き起こされる部分列の13の可能性のある歪みの多くとも一つにおける長さ7の各部分列について、二次数列の中に見出されるであろうという特性を有している。第四の二次数列A4は、28の可能性のある歪みの多くとも七つについての同一の特性を有している。検出される主数列の部分列の単一の正しくないデコーディングの可能性は大きい。

【0152】

選択された列によれば、410815348の異なる位置の合計が、セクションの各次元にコード化され得る。コード化され得る異なるセクションの数は、上述された通り 63^2 である。したがって、コード化され得る位置の合計数は、 $63^2 \times 410815348^2 = 6.7 \times 10^{20}$ 位置である。

【0153】

もしも、各位置が、 $0.3 \times 0.3 \text{ mm}^2$ の面に対応するならば、これは6000万km²の面上にコード化され得る特有の位置に対応する。位置コードによって理論的にコード化され得る全ての特有の点によって作られるこの面は、仮想面と呼ばれ得る。位置コードがコード化される座標は、それ故仮想面上の点についての絶対座標である。位置コードの部分は物理的ベースまたは面上に適用され得る。それから、位置コードは、このベース上で位置をコード化する。しかしながら、座標は、通常、物理ベース上の位置についての絶対座標に関連しないが、仮想面上の点について関連している。

【0154】

〔位置デコーディングの実際の例〕

位置コードは、位置コードを読み取るためのセンサ、および何らかの適切なタイプのプロセッサユニットを有し得るデコーディングデバイスにおいてデコードされ得る。プロセ

ツサユニットは、実際のプロセッサ、作業メモリおよび、位置コードをデコーディングするための命令群を有するコンピュータプログラムがそこに格納される、プログラムメモリを備えている。前記プロセッサユニットは、典型的なパーソナルコンピュータに、手持ち読み取りデバイスにまたは何らかの他の適切なデバイスに組み込まれ得る。前記デコーディングデバイスは、その代わりに、ASICまたはFPGAまたはこの特定のタスクのために適するように適用され得る同様のユニット、またはディジタルおよび／またはアナログ回路またはそれらの何らかの適切な組み合わせからなるような特定目的のハードウェアからなることもできる。

【0155】

デコーディングデバイスの特定の例が以下において説明されるであろう。

【0156】

以下の記述は図9のフローチャートに言及している。

【0157】

デコーディングデバイスへの入力信号は、位置コードの検出されたサブセットのイメージまたは他の何らかのディジタル表現からなっており、そのサブセットは、少なくとも一点についての座標をコード化している（ステップ700）。この例においては、位置コードは、上述されたように、ラスタにおけるラスタ点に関連する所定の方法にて変位された、ドットによりグラフィカルに表現される。

【0158】

デコーディングの最初のステップにおいては、デコーディングデバイスが、例えば閾値弁別を用いることによって、イメージにおけるドットを識別し、ラスタをドットに合わせる（ステップ705）。ラスタは、例えば、異なるドットの対の間ににおける距離の利用によって、本出願人の国際特許出願WO01/26033に記述された方法によって、または代わりにフーリエ変換の利用による国際特許出願WO01/75783に記述された方法で決定され得る。各ラスタ点について、関連付けられたドットの変位が決定され、変位に依存する値0～3が与えられる。少なくともサイズが $(n_1 + 1) \times (n_1 + 1)$ の行列が選択され、ここで、 n_1 は、位置決めのために使用される主数列の部分列の長さであり、その場所は、主数列内に明確に決定される。当該例に関連して、 8×8 行列が選択される $((n_1 + 2)) \times (n_1 + 2)$ 。この行列は、変位値をビットの対に変換するテーブルO/Iにおいて検索されることによって、ビットの対に変換される変位値によって、x-行列（x-位置コードとも呼ばれる）およびy-行列（y-位置コードとも呼ばれる）に分離される（ステップ710）。各ビット対における最初のビットはx-行列を形成し、且つ各ビット対における第二のビットは、y-行列を形成する。

【0159】

次のステップにおいて、x-およびy-行列の回転の四つの可能性のあるタイプ（0, 90, 180および270度）は、調査され且つ現在の回転が決定される（ステップ715）。主数列において失われた長さ8の部分列が生じ、行列の不完全な回転の場合には、先に述べられた事実が用いられる。

【0160】

同時に、ある度合いのエラー訂正が長さ8の部分列に基づいて実行される。

【0161】

正しい回転が決定されたとき、次のステップにおいて、中間にあり且つ正しい方法で回転されそして $n_1 \times n_1$ なるサイズ、すなわちこの例においては二つの 6×6 行列、を有する、x-およびy-行列のみが使用される。x-行列およびy-行列が、図9における並列フローによって示される、xに先だってyまたはその逆を用いて並列的にまたは逐次的にデコードされる。

【0162】

x-行列のデコーディングの第一のステップ720において、一次差分数列の部分列が決定され、それは図9において、一次部分列と呼ばれる。より詳細には、x-行列において、行列内の列におけるビットが、主数列の部分列を構成する。主数列における対応する

場所は、部分列を場所に変換するテーブル M I 内が参照されることにより決定される。その後に、差分数は、主数列の部分列によってコード化された差分数が決定される。この例においては、それらは数において 5 である。より詳細には、隣接する部分列の場所の間の差分 d として差分数が決定され、前記差分は主数列の長さを法として決定される。

【0163】

そのようにして取得される差分数の列は、一次差分数列の部分列であり、それは x - 方向についてのコーディングのために使用されるが、そのサイズのために、デコーディングデバイスには格納されない。その代わりに、各部分列が二次差分数列の各々からの部分列の組み合わせに変換され得る一次差分数列の特性の使用がなされる。ステップ 725 において、一次差分数列が、そのように二次部分列に変換される。より詳細には、各差分数は次のように書き換える：

【0164】

$$d = d_1 + b_1 \times d_2 + \dots + b_{n-1} \times b_n \dots \times b_{n-1} \times d_n$$

ここで、 n は二次差分数列の数であり、そして b_i は、二次差分数列 i 内におけるベースである。これに関する例は、次のようになる：

【0165】

$$d = 5 + d_1 + 3 \times d_2 + 9 \times d_3 + 18 \times d_4$$

ここで、 $d_1 \sim d_4$ は、整数除算またはテーブル参照によって決定され得る。

【0166】

一次差分数列のデコードされた部分列における五つの数の各々は、それ故結果として四つの数 d_1, d_2, d_3, d_4 となる。したがって、第一の二次差分数列の部分列を形成する五つの d_1 数が得られ、第二の二次差分数列の部分列を形成する五つの d_2 数が得られ、第三の二次差分数列の部分列を形成する五つの d_3 数が得られ、そして第四の二次差分数列の部分列を形成する五つの d_4 数が得られる。例が、下に与えられる：

| | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|
| d | 2 | 3 | 4 | 5 | 5 | 1 | 9 | 3 | 7 |
| d ₁ | 2 | 0 | 0 | 0 | 0 | 1 | | | |
| d ₂ | 1 | 0 | 2 | 0 | 0 | | | | |
| d ₃ | 0 | 1 | 1 | 1 | 1 | 0 | | | |
| d ₄ | 1 | 2 | 2 | 0 | 2 | | | | |

【0167】

二次差分数列の部分列のそれぞれの場所は、それぞれの二次差分数列の部分数列をそれぞれの二次差分数列における場所に変換する四つのテーブル D C I_i を参照することによって決定される（ステップ 730）。

【0168】

各部分列について、場所 p_i がそのように得られる。これらの場所について、次のような場合がある：

$$P = p_1 \pmod{11}$$

$$P = p_2 \pmod{12}$$

•

•

$$P = p_m \pmod{1m}$$

但し、 P は、座標 x に対応する一次差分数列における場所、 p_i は、二次差分数列における場所、 l_i は、二次差分数列 i の長さ、そして m は、二次差分数列の数である。

【0169】

この式のシステムは、よく知られた中国の余剰定理を用いて解決され得る、それについては、例えば Niven, Suckerman に非常に詳細に説明されており、数の理論への導入は Springer Verlag に説明されている。こうして、ステップ 735 において、 x が決定される。

【0170】

```
D e f i n e L = p r o d ( i = 1 , m ) l i および q i × ( L / l i )
= 1 ( m o d l i )
```

一次差分数列における場所 P は、次のようにして得られる：

```
P = ( s u m ( i = 1 , n ) ( ( L / l i ) ) × p i × q i ) ( m o d L )
```

関連する例において

```
L = 1 1 × 1 2 × 1 3 × 1 4 = 2 3 6 × 2 3 3 × 3 1 × 2 4 1 = 4 1 0 8 1 5 3 4 8
```

および

```
q 1 = 1 3 5
```

```
q 2 = 1 4 5
```

```
q 3 = 1 7
```

```
q 4 = 6 2
```

が得られる。

【0171】

もしも、二次差分数列の部分列について例えば、場所 $p_1 = 97$; $p_2 = 176$; $p_3 = 3$ および $p_4 = 211$ が得られるならば、一次差分数列における対応する場所は、中国の余剰定理に従って 170326961 となるであろう。

【0172】

P が決定されているとき、次のステップは、場所が属する x - セクションを決定することである。これは、最初に $y = 0$ という仮定のもとに、そして $s(x, 0)$ の計算によって実行され、それは $x s = 0$ についての列 x のための部分数列の主数列における場所である。

【0173】

$s(x, 0)$ は、位置 x についての位置コードの決定に関連して上述されたのと同様に二次差分数列を用いて計算される。もしも、 $s(x, 0)$ が、X 方向についての位置決めのために使用される六つの部分数列の第一の部分数列の場所から引かれると、 $y = 0$ について $x s$ が得られる。

【0174】

y - 行列は、 x - 行列に先だって、それと並列的にあるいはそのデコードの後にデコードされ、一次差分数列における場所は、 x - 方向についての上述された対応する方法にて、図 9 におけるステップ 740 ~ 755 に従って、 y - 方向について決定される。加えて、関連する y - セクション $y s$ が x - セクションについて上述されたのに対応する方法で計算されるが、 $x = 0$ の仮定のもとである。最後に、ステップ 760 において、 $x s$ は、 y を考慮して決定され、且つ $y s$ は、 x を考慮して y (主数列の長さを法として) を減算することによって、 $x s$ から $y = 0$ について決定され、そして x (主数列の長さを法として) を減算することによって $y s$ から $x = 0$ について決定される。

【0175】

次のテーブルは、したがって、位置コードをデコードするために使用され得る：

M I : それは、主数列の部分列を主数列の場所に変換する、

D C I_i : それは n の二次差分数列の各々の部分列を、n の二次差分数列における場所に変換する、そして

O I : それは変位値をビット対に変換する。

【0176】

図 9 におけるフローチャートは位置デコーディングが、例えばコンピュータプログラムにおいて、どのように実現され得るかの粗い概略的な例であることは強調されるべきである。

【0177】

付録 B (APPENDIX B)

次に、図 10 のフローチャートを参照して、確率計算の手段によりコーディングパターンをデコーディングする方法の例について説明する。デコーディングするコーディングパ

ターンは、同出願者によるWO01/26033に記載されているタイプのものである。

【0178】

図11aは、光学的に読み出し可能な位置コード17の形式の位置コーディングパターン（以下、コーディングパターン）を含んだ表面16を備える一枚の紙15を示す。位置コードはマーク18で構成され、そして明瞭性のために大幅に拡大してある。図11bは、図11aの位置コード17をさらに拡大した部分19を示す。デバイスは、位置コードの部分領域の画像を記録し（ステップA）画像中の複数のマーク18を識別し（ステップB）、また、ラスターポイント22にて交差しているラストーライン21を備えたラスター形式の基準システムを画像に一致させる（ステップC）ように構成されている。この一致は、各マーク18がラスターポイント22と関連する方法で実行される。例えば、マーク23はラスターポイント24と関連している。その結果、ラスターの一致により、各マークがどのラスターポイントに属するかの決定が可能となる。この例では、ラスターは四角格子の形態を有するが、これ以外の形態を使用することも可能である。本願明細書中で参照により援用している、同出願人の明細書WO01/75783、WO01/26034、SE0104088-0は、画像中におけるラスターのマークへの一致をより詳細に開示している。

【0179】

「理想的」なコーディングパターンにおいて、各々のラスターポイントと関連するマークは唯一つである。コーディングパターンの画像における変形および不完全性のために、コーディングパターンの画像中で、どのマークがコーディングパターンに属し、複数のマークのどれが特定のラスターポイントと関連するべきものかを決定することは困難である。この理由から、この例では、複数のマークを、データのデコーディングにおける同一の一個のラスターポイントに関連付けすることが可能である。或るラスターポイントと、これに関連したマークとが、共にラスターポイントに属するマーク要素を形成する。

【0180】

この例におけるコーディングパターンでは、マーク18の値が、関連付けされたラスターポイント22に対するその変位によって定義される。より詳細には、これは、マークの値を定義するラスターポイントに対するマークのポイントの位置である。一般に、このポイントはマークの主要ポイントである。この例におけるコーディングパターンでは、各マークにつき四つの理想的な位置が存在する。これらの位置は四本のラストーライン21の各々の上にあり、四本のラストーラインは一個のマークに関連したラスターポイント22から伸びている。また、これらの位置は、ラスターポイントから等距離で離れた所に位置している。図12a～図12dに、或るマークのための理想的な位置25を拡大図にて示している。これらは、図12aでは値“0”を、図12bでは値“1”を、図12cでは値“2”を、図12dでは値“3”を有する。したがって、各マークは四つの異なる値“0～3”を表すことができる。

【0181】

様々な理由から、記録した画像中で識別されたマークは、理想的な位置を有していないことが多い。そのため、多くの場合、一個のマークにつき行う一個の値を明確に決定することが困難になる可能性がある。このために、デバイスは、識別した各マークについて、マークが定義する各値“0～3”的関連する値確率を計算するよう構成されている（ステップD）。各マークについての値確率は、図13に見られるようにマーク27からその理想的な位置25までの、あるいはより詳細には、マーク27の主要ポイントから理想的な位置までの距離26の増加関数である。通常、複数のマークは理想的な位置の周辺に分布していると予想できる。

これはつまり、値確率P(d_i)が公式P(d_i) = k e x p(- (d_i)² / V)により計算することができ、ここで、k = 定数、d_i = 或るマークから理想的な位置までの距離、V = 定数、この例では距離の変数である。この変数は経験的に計算される。したがって、各マークについて四つの値確率を計算することができる。時に、ラスターポイントの中心にマークが見つかることもある。その場合は、マークから各々の理想的な位置までの

距離の大きさが等しいため、一個のマークについての四つの値確率は等しくなる。ラスターポイントに関するマークが存在しない場合、つまり、関連する要素がマークを含んでいない場合には、ラスターポイントの中心にマークが存在する、すなわち値確率が等しいものとして扱われる。

【0182】

一個のラスターポイントに関するマークが一個以上、例えば三個ある場合、つまり、関連するマーク要素内に三個のマークがある場合には、そのラスターポイントまたはマーク要素について合計で 3×4 の値確率があることになる。したがって、本デバイスは、各ラスターポイントまたはマーク要素につき、また各値につき、この値を共に定義するラスターポイントに関するマーク値確率を決定するように構成されている（ステップE）。これらのラスターポイントについての値確率は、ラスターポイント確率とも呼ばれる。ラスターポイントについて値確率を計算することで、記録した画像内の全てのマークを位置決定に考慮することができ、また、情報損失の危険を最小化することができる。上述は、各マーク要素について、マーク要素が各々の値を定義する値確率が計算されるため、これ以降の説明では、値“0”～“3”をマーク要素値と呼ぶ。

【0183】

或るマーク要素についての値確率は、比較されるマーク要素内のマークについての値確率により決定することができ、また、最も高い値確率が、使用可能なマーク要素値の各々について選択される。あるいは、要素についての値確率は、マーク要素内のマークの使用可能な各マーク要素値についての値確率の加重和となることができる。当然ながら、マーク要素についての値確率は、上述した方法以外の方法によっても決定することができる。

【0184】

この例における、一個のマーク要素についての値確率の計算を、図14の数字による例で示している。図14aは、ラスターポイント22を、二個の関連するマーク28、29と共に示す。マーク28、29は共に、ラスターポイント22に属するマーク要素を構成している。図14b中の表30、31は、各々のマークの使用可能な値についての値確率 P_1 を含んでいる。図14c中の表32は、該要素の使用可能なマーク要素値について結果として得られた値確率 P_2 を含んでいる。この例では、マーク要素についての値確率は相対的である。あるいは、代わりに、これらを適切な方法で正規化することができる。ラスターポイントに関するマークが一個しかない場合、つまり、関連するマーク要素内にマークが一個しかない場合には、そのマークおよびマーク要素についての値確率は明らかに同一である。

【0185】

画像の記録時に、デバイスから表面までの距離が、記録する位置コードの大きさ、さらに、これにより、画像に一致させるラスターの大きさに影響を及ぼす。画像を位置に変換するためには、所定数のマーク要素、この例では 8×8 マーク要素を使用する。 8×8 よりも多くのラスターポイントを画像に一致させる場合には、過剰なマーク要素が識別される。したがって、本デバイスは全ての識別したマーク要素から、表面上での位置に関する情報を最も多く提供する一組のマーク要素を選択するようにさらに構成される（ステップF）。しかし、この一組の要素は連続したものである必要はない。この一組のマーク要素に含まれるマーク要素は、この例においては、ラスターポイント行列に関連し、ラスターポイント画像に一致された状態にあるが、これは必須条件ではない。換言すれば、目的は、記録した画像の情報分量を最小化する、関連した値確率を含んだマーク要素を各マーク要素値について選択することである。この目的のために、識別したマーク要素の各々についてエントロピーが計算される。その後、エントロピーの和が最小の 8×8 のマーク要素が選択されるが、これは、記録した画像の最大の情報分量に関連している。マーク要素に対する値の確率が以下の式のように正規化されると、

$$\sum_i P_{2,i} = 1$$

【0186】

マーク要素のエントロピーHは、以下の式で計算される。

$$H = -\sum_i P_{2,i} \log_2(P_{2,i})$$

ここで、 $P_{2,i}$ は、マーク要素値*i* (*i* = 0, 1, 2, 3)についてのマーク要素の値の確率であり、 \log_2 は、二つの対数である。そのため、一個を除く全ての値確率がゼロである際の、ある値確率が等しく高く最小である場合に、該要素についてのエントロピーは最大となる。エントロピー計算の手段により 8×8 マーク要素を選択する代替案が、各マーク要素に最高の値確率を情報値として使用するために用いられる。この場合、 8×8 マーク要素についての情報値の和によって構成される情報分量を最大化する、連続した 8×8 要素が選択される。

【0187】

この例では、コーディングパターンが、表面4上の一つのマークについて二個の座標をコーディングする。これらの座標は個別にコーディングされる。そのため、これらを二つの次元におけるデータと呼ぶことができる。コーディングパターン内の各マークが、より詳細には、第一座標のデコーディングに使用される第一ビットをコーディングし、第二座標のデコーディングに使用される第二ビットをコーディングする。

【0188】

記録した画像内のコーディングパターンをデコーディングする際に、したがって、一個の要素について使用可能な各マーク要素“0”～“3”が、この例では二進数である第一、第二デコーディング値に変換される。そのため本デバイスは、一組のマーク要素内の 8×8 マーク要素について、マーク要素値“0”～“3”を四個の異なるビットの組み合わせ(0, 1)、(0, 0)、(1, 0)、(1, 1)に変換するように構成されている。これらのビットの組み合わせは、各マーク要素についての、マーク要素値に属する値確率を有するが、これについては、図15a中の表における先行の数字による例を参照のこと。ビットの組み合わせにおいて、第一ビット、つまり第一デコーディング値が第一の次元に関連し、第二ビット、つまり第二デコーディング値が第二の次元に関連する。関連するマーク要素値についての値確率 P_2 が第一、第二デコーディングに関連している。そのため、一組のマーク要素を、第一の次元のための関連する値確率を有する第一デコーディング値の第一組と、第二の次元のための関連する値確率を有する第二デコーディング値の第二組とを作成するべく使用することができる(ステップG)。表33は、一組のマーク要素内の一組のマーク要素について説明している。表33'、表33''は、関連する値確率を有する第一組内における第一デコーディング値と、関連する値確率を有する第二組内における第二デコーディング値とを含んでいる。第一、第二デコーディング値の各々は、上述の説明から明白であるように、0または1のいずれかである。

【0189】

本デバイスは、マーク要素の第一組内の各マーク要素について、異なる使用可能な各々の第一デコーディング値を一個の値確率と、また、第二組内の異なる使用可能な第二デコーディング値の各々を一個の値確率と関連付けるように構成されている。この例では、使用可能な第一、第二デコーディング値は0および1であるため、上述の結果、一組のマーク要素内の各マーク要素につき、第一組、第二組において、デコーディング値0に一個の値確率、また、デコーディング値1に一個の値デコーディング値が得られる。これ以降、デコーディング値0についての値確率を0確率、デコーディング値1についての値確率を1確率と呼ぶ。

【0190】

表33'を参照すると、この例では、上述は、0である第一デコーディング値に関連し

た第一組内の値確率を比較することにより、一組のマーク要素内の各々のマーク要素について実行される。次に、最も高い値確率が0確率として選択され、第1行列34内に保存される。この方法では、第一デコーディング値に関連した第一組内の1である値確率が比較される。次に、最も高い値確率が1確率として選択され、やはり第1行列34内に保存される。表33”を参照すると、上述の方法が、その後、第二組内の値確率、第二符号化値、第2行列35についても繰り返される。そのため、0確率と1確率を含んだ第1、第2行列を作成するために、第一組、第二組が使用される(ステップH)。その結果を、図15b中の連続した、数字による例に示す。あるいは、一組の要素内における 8×8 要素についての、0確率と1確率を含んだ第1、第2行列が、加算され、0である、第一デコーディング値に関連した第一組内における値確率によって、また、加算され、1である、第一数に関連した第一組内における値確率によって作成され、ここで、第一の値確率の和が0確率として記録され、第二の値確率の和が1確率として記録される。

【0191】

そのため、一組のマーク要素内における 8×8 マーク要素が、各々が 8×8 の行列要素を備えた二個の行列34、35と関連し、ここで、各行列要素が一個の0確率と一個の1確率を含んでいる。これら第1、第2行列の手段により、位置の座標を決定することができる。

【0192】

記録した画像内の全ての要素について値確率を決定した後に一組のマーク要素を選択する代わりに、識別された全てのマーク要素について、行列34、35に関連する行列が決定されるまで待つこともできる。この場合には、各行列内における 8×8 の行列要素を、関連する0確率と1確率に基づいて選択する。これを行うための一つの方法は、0確率および1確率のうち一方が高く、他方が低い 8×8 の行列要素を選択するものである。この場合、両座標を決定するために、同一のマーク要素に関連した行列要素を選択する必要はなく、また、二個の座標について、関連した異なるマーク要素の計算が続行される。

【0193】

この例では、位置コードが、第一の循環主数列に基づいた第一次元内にある。これにより、マーク要素のマーク要素値間における関係についての条件が与えられる。第一の循環主数列は、該数列内における、所定の長さの各部分数列のための位置が明確に決定されるという特性を有する。この例では、所定の長さは6である。そのため、第一の循環主数列内の任意の位置において六個の連続した数字を取った場合、これら六個の数字は、この連続における第一の循環主数列において一度だけ現れる。この特性は、第一の主要数列の最後が第一の主要数列の最初と連結している場合にも適応される。したがって、第一の主要数列は循環と呼ばれる。この例では、二進主要数列を用いる。六個の数字を有する部分数列のための位置を明確に決定する場合には、第一の主要数列は最大である $2^6 = 64$ の長さを有することができ、長さ6の部分数列は、第一の循環主数列内の0~63の位置を有することができる。しかし、長さ6の第一の主要数列を選択した場合には、以降の説明から明らかになるように、改善されたエラー修正特性を設けることが可能である。したがって、以降の説明では、第一の主要数列の長さは63であり、そのため0~62の範囲内における固有の位置が定義される。

【0194】

図16は、位置コーディングに関連して使用できる第一の循環主数列の例を示す。部分数列0、0、0、0、0は、第一の主要数列内に、例えば明確な位置0を、部分数列1、1、1、1、1、0は明確な位置9を、部分数列1、1、1、0、1、0は明確な位置11を有する。表面上の位置を決定するために、記録した画像内で 6×6 マーク要素を識別する必要がある。しかし、上述しているように、位置の決定には 8×8 マーク要素を使用するものであり、その理由は以降の説明より明らかになる。上述したように、第一次元において位置コードが基づく第一の循環主数列が、長さ6の相互的に固有な部分数列のみを含むという特性を有する。その結果、さらに、長さ8の各部分数列についての、第一の循環主数列内における位置が明確に決定される。この事実は、表面上の位置についての

座標決定においても用いられる。

【0195】

本デバイスは、第一の循環主数列内における長さ8の固有の部分的数列の各々に、第1行列34内における列の各々を一致させるように構成されている（ステップI）。この方法を図17に示す。同図は、長さ8の二進数の部分数列36と、第1行列34内における列37（図15b）の一例を示しており、該列は行列要素を備えているが、この行列要素の各々は、それぞれ0および1である第一デコーディング値に関連した0確率と1確率を含んでいる。各行列要素について、部分数列36内の関連する数に基づき、0確率と1確率のうち一つが選択される。部分数列36内の1番目の数は例えば0であり、これはつまり、列37内の第1行列要素について0確率が選択されることを意味する。部分数列内の2番目の数は1であり、これはつまり、列37内の第2行列要素について1確率が選択されることを意味する。第一の主要数列内の各部分数列、第1行列34内の各列について、本デバイスはさらに、行列要素について関連的に選択された0確率と1確率を乗算することで、第一数列確率を計算するように構成されている（ステップJ）。図17では、部分数列36および列37に関連した第一数列確率38を算出した。この動作の後に、第1行列34内の各列について、関連する固有の数列値を各々含んだ63個の第一数列確率が存在することになる。これらの数列値は、第一の循環主数列内における関連する部分数列の位置によって定義される。デバイスは、各列につき、最も高い第一数列確率と、関連する数列値を選択し、これらを保存するように構成されている。

【0196】

ここでは、第二の次元中の位置コードは、この例では第一の循環主数列と同一の特性を有する第二の循環主数列に基づいている。

【0197】

デバイスはさらに、上述に関連した方法において、第二の循環主数列内における長さ8の固有の部分数列の各々を、第2行列35内の行と一致させるように構成されている。行列35内の行は、行列34内の列と全く同様に、各々0および1である第二デコーディング値に関連した0確率と1確率の対応する一つを含んでいる。第二の循環主数列における部分数列内の関連数に基づき、各行列要素について0確率および1確率の一つが選択される。第2行列35内の各行についての、第二の主要数列内の各部分数列につき、デバイスはさらに、行列要素について、関連的に選択された0確率および1確率を乗算することで、第二数列確率を計算するように構成されている（ステップJ）。この動作後には、第2行列35内の各列につき、各々関連する固有の数列値を含んだ63個の第二数列確率が存在することになる。これらの数列値は、第二の循環主数列内における関連する部分数列の位置によって定義される。デバイスはさらに、最も高い第二数列確率と、関連する数列値とを各行につき選択し、これらを保存するように構成されている。

【0198】

この例で使用している位置コードは、循環主数列の異なる回転または循環移動の使用に基づいている。例えばx方向において位置をコーディングするためには、第一の主要数列を印刷するか、または、これ以外の何らかの方法によって回転させるか、あるいは、表面と交差する列内において、すなわち、位置をコーディングする方向に対して直行するy方向において、上から下に向かって、異なる方法で円形シフトさせる。主要数列は同一の列内で繰り返し印刷されるが、これは、主要数列長さに関連した位置よりも多くの位置をy方向にコーディングする場合に必要である。次に、主要数列の同一の回転が全ての繰り返しに使用される。これはつまり、異なる回転を異なる列に使用できることを意味する。

【0199】

近接した列の各対が差分数Dを定義する。差分数Dは、各列内における第一部分数列のための主要数列内における位置間の差によって得られる。部分数列の位置間の差を列内で一段階下げて取った場合にも、これらの位置が同胞の方法で相殺されるため、同一の結果が得られる。したがって、差分数Dは、列内のどの「高さ」が主要数列内の部分数列の位置と比較されるかに關係なく、常に同じである。そのため、各対の列について、差分数D

はy方向において一定である。近接した列間の差分数が、第一の次元において表面上の位置の座標を得るべく使用できるのではなく、むしろ、一組の差分数を形成する。

【0200】

第二方向、この場合は例えばy方向における位置コードは、第一の次元における位置コードと同じ原理に基づくことができる。次に、第二の主要数列を、表面上の列内において、つまりx方向において、左から右に向かい、異なる円形シフトで構成することができる。近接した列間で差分数が定義され、これらの差分数が、第二の次元における表面上の位置の座標を得るために使用可能な一組の差分数を形成する。

【0201】

そのため、位置コードは、第一方向についての一個の部分位置コードと、第二方向についての部分位置コードとで構成されている。

【0202】

上述から明白であるように、部分数列は明示値ではなく、図形コーディングで書かれたものである。図形コーディングでは、マークが部分位置コードの重ね合わせを定義する。

【0203】

位置コードは、表面上の所定の方向に配置された主要数列に基づいているため、修正すべき位置決定のために、マークをこれらの方向においてデコーディングしなくてはならない。適正なデコーディング方向は、上述したように、上から下、および左から右へ向かう。

【0204】

画像の記録時に、デバイス1を、異なる位置で、表面および位置コードに対して回転した状態に維持することができる。図18中に矢印40として示す四つの使用可能な記録回転が存在する。位置コードは0、90、180または270度の角度で回転した場合、その外観が本質的に同一となるため、記録した位置コードの画像自体が位置コードとデバイス間の相対回転を示すことはない。しかし、位置コードを回転すると、各マークの、これに関連したラスターポイントに対する変位の方向が変更される。その結果、これが如いては、変更されているマークの変位をコーディングするビットの組み合わせ（第一デコーディング値、第二デコーディング値）となる。位置コードの「適正な」回転の状態では、マークは、列内で上から下、また行内で左から右に向かう適正なデコーディング方向に配置される。位置の適正な回転が0である場合、不適正な回転に以下が適用される：

【0205】

・右回りに90度：上から下へ向かって配置された、「適正な」回転のマークを伴う列が、右から左へ向かって、つまり不適正なデコーディング方向に配置されたマークを伴う行となり、また、左から右へ向かって配置された、「適正な」回転のマークを伴う行が、上から下へ向かって、つまり適正なデコーディング方向において配置されたマークを伴う列となる。

【0206】

・180度：「適正な」回転のマークを伴う列が、下から上へむかって、つまり不適正なデコーディング方向に配置されたマークを伴う列となり、また、「適正な」回転のマークを伴う行が、右から左へ向かって配置された、つまり不適正なデコーディング方向において配置されたマークを伴う行となる。

【0207】

・右回りに270度：「適正な」回転しているマークを伴う列が、左から右方向へ向かう、つまり適正なデコーディング方向において配置されたマークを伴う行となり、また、「適正な」回転されているマークを伴う行が、下から上へ向かって、つまり不適正な方向において配置されたマークを伴う列となる。

【0208】

列および行内のマークが不適正なデコーディング方向において配置されている場合には、デコーディング時に、各要素について0確率および1確率が逆転される。

【0209】

したがって、デバイスは、以下に説明するように、記録した画像内における位置コードの部分領域の異なる回転をテストするように構成されている。第1行列34、第2行列35上で実行された各々の動作、つまり、循環主数列内の部分数列の、該行列内における列および行の各々への一致(ステップI)、数列確率の計算(ステップJ)、列および行の各々についての、関連する数列値を伴った最も高い数列確率の選択が、180度回転され「逆」された第1行列34、第2行列35上でも実行され、そのため、図15c中の行列にそれぞれ参照符号34'、35'を付している。回転され、逆転されたこれらの行列34'、35'は、記録した画像における位置コードの部分領域の逆転バージョンに関連している。この理由を、記録した画像における位置コードの部分領域の一例を示す図19にて説明している。同図では、明瞭性のために九個のマークのみを使用しているが、例証のために、これらのマークはそれぞれ一個のラスターポイントに関連している。位置コード45は画像内に記録されたものである。位置コード45'は、同じ位置コードを逆転したものである。行列46、47は、右方向に進行方向を変更した位置コード45について、行列34、35にそれぞれ関連し、また、行列48、49は、逆転させた位置コード45'について、行列34'、35'にそれぞれ関連している。逆転させた位置コードについての行列48、49を180度回転させて反転させた場合、右方向に進行方向を変更した位置コードについて、行列46、47が得られる。この内容においては、反転とは、各行列要素内の0確率および1確率が位置を変更することを意味する。

【0210】

上述の手順の実行後には、行列34、34'内の各列について、また、行列35、35'内の各行について、関連する数列値を伴う最も高い数列確率が存在する。そこで、本デバイスは、行列34、34'および35、35'の各々について、関連する最も高い数列確率を乗算することより、回転確率を計算する(ステップK)ように構成されている。最も高い回転確率に関連した行列34、34'の数列値についての、最も高い数列確率に関連した数列値、および、最も高い回転確率に関連した行列35、35'の数列値についての、最も高い数列確率に関連した数列値に基づいて、該位置の座標を計算することができる。

【0211】

上述したように、四つの回転全てを調べる必要はない。これは、例証の手段によって単純に説明される。次に、図19に示すように、記録した画像における位置コードの部分領域が1(45)であると推測する。さらに、位置コードの「適正な」回転が、記録されたものに対して、右方向に90度回転すると推測する。上述の説明によれば、適正な回転は0度の回転に関連すると推測され、これはつまり、記録した画像内の位置コード45が、「適正な」回転に対して右方向に270度回転することを意味する。その結果、位置コード45'が、「適正な」回転に対して右方向に90度回転する。上述した方法で、図19において位置コード45、45'がデコーディングされる。0度の「適正な」回転におけるマークを伴う列が、上述したように、位置コード45内に、適正なデコーディング方向において配置される。適正な方向によって、これらの行に関連する、数列確率と、如いては回転確率とが上昇する。適正な回転におけるマークを伴う行は、上述のように、位置コード45内に不適正なデコーディング方向において配置された列である。不適正な方向と反転を組み合わせると、これらの列に関連した値確率と、如いては回転確率とが低下する。この関係は、位置コード45'についての関係とは逆のものである。0度の適正な回転におけるマークを伴う列は、上述したように、位置コード45'内に不適正なデコーディング方向において配置された行である。不適正な方向により、これらの行に関連した、数列確率と、如いては回転確率とが低下する。適正な回転におけるマークを伴う行は、上述したように、位置コード45'内に適正なデコーディング方向において配置された列である。この適正な方向により、これらの列に関連する、数列確率と、如いては回転確率とが上昇する。

【0212】

位置コードの「適正な」回転を記録する際に、列および行が、上述したように画像内の

「適正な」方向へと伸びる。これはつまり、図15中の例について、行列34、35の両方についての回転確率が、行列34'、35'についての回転確率よりも高くなることを意味する。これは、位置コードの「適正な」回転が記録されたことを表す。これにより、行列34についての最も高い数列確率に関連した数列値に基づいて、第一座標を計算することができ、また、行列35についての最も高い数列確率に関連した数列値に基づいて、第二座標を計算することができる。

【0213】

「適正な」回転に対して180度回転した位置コードを記録する際に、列および行が、画像内で「不適正な」方向に伸びる。これはつまり、例えば図15では、行列34'、35'の両方についての回転確率が、行列34、35についての回転確率よりも高くなることを意味する。これは、位置コードが、「適正な」回転に対して180度の回転において記録されたことを表す。これにより、行列34'についての最も高い数列確率に関連した数列値に基づいて、第一座標を計算することができ、また、行列35'最も高い数列確率に関連した数列値に基づいて、第二座標を計算することができる。

【0214】

「適正な」回転に対して右方向に90度または270度回転した位置コードの記録が、同一の回転に属していない最も高い回転確率によって表される。図15中の行列34、35が、適正な方向に対して右方向に90度される位置コードより派生していると推測した場合、行列34についての回転確率、行列34'についての回転確率、行列35についての回転確率が、行列35'についての回転確率よりも低くなる。この場合、行列34についての最も高い数列確率に基づいて第二座標が計算され、また、行列35'についての最も高い数列確率に関連した数列値に基づいて第一座標が計算される。図15b中の行列34、35が、適正な方向に対して右方向に270度回転する位置コードから派生していると推測すると、第二座標は、行列34'についての最も高い数列確率に関連した数列値に基づいて計算され、第一座標が、行列35についての最も高い数列確率に関連した数列値に基づいて計算される。

【0215】

したがって、記録した画像内における位置コーディングの回転方向により、或る行列についての回転確率が、その行列が90度、180度、または270度の角度で回転した際に変更されるという事実が得られる。位置コードの回転が0ではない場合、つまり、記録した画像内における位置コードの回転が不適正である場合には、上述したように回転確率が低下する。これは、循環主数列内における長さ8の固有の部分数列が、主要数列内では反転または逆転しないという事実に依存する。このような条件は、長さ6の部分数列の主要数列について満たされなければならないが、これはつまり、主要数列が大幅に低下することを意味し、如いては、より少ない位置がコーディングされることを含意している。したがって、これは、理論的に必要なのは 6×6 であるが、位置決定について 8×8 マーク要素を使用できる一つの理由である。

【0216】

回転検出に使用されるものと同一の基本原理を、エラー修正に用いることができる。例えば、位置決定に必要なものよりも長い所定の長さの部分数列が、主要数列内で一個のビットが反転した状態で生じないように、主要数列を選択することが可能である。次に、このようなより長い部分数列内において、一個を除く全てのビットを適正に検出できる場合には、不適正なビットを修正することが可能である。これは、理論的には 6×6 要素しか必要でないのに、位置決定について 8×8 マーク要素が使用される別の理由である。

【0217】

そのため、主要数列を賢明に選択することで、コーディングパターンのエラー検出特性およびエラー修正特性を相当に向上させることが可能である。

【0218】

しかし、長さ8の部分数列が反転または逆転しない循環主数列の特性は、64ビット長の主要数列には与えられず、これが、主要数列の長さが63に選択されている理由である

。

【0219】

そのため、デコーディングでは、エラー修正特性を得るべく冗長情報が使用される。上述した例では、 6×6 のラスターポイントに基づいて位置情報を抽出することができるにも関わらず、デコーディングに 8×8 のマーク要素を使用しているが、これはつまり、位置を決定するために 56 ビットの冗長情報があることを意味し、これが $[(8^2 - 6^2) \times 2]$ と表される。デコーディングでは、現在画像に属する値確率を使用しながら、列および行によって、現在画像内の情報が、位置コード内で生じた異なる部分数列と一致される。冗長情報と、確率と、マーク要素の値どうし間ににおける関係のための従来の条件とを組み合わせることで、現在画像内のインターフェースに対する優れた無感応性が得られる。これにより、各々の独立したマークの値の重要性が、最も高い確率を有する部分数列の重要性よりも低下する。

【0220】

エラー修正の必要性がさらに制限されると、デバイスを、第一間行列内の各列について、また、第二行列内の各行について、各マーク要素についての 0 および 1 の確率のうち最も高い確率に関連して、数列、如いては数列値を直接選択するように構成することができる。

。

【0221】

記録した位置コードの、「適正な」回転に関連した回転 40° が確立されている場合には、位置の第一、第二座標を決定することができる（ステップ L）。これは、上述したように、図 18において、第一座標について $Sx_1 \sim Sx_8$ (41)、第二座標について $Sy_1 \sim Sy_8$ (42) と示される数列値に基づいて実行される。

【0222】

デバイスは、数列値 $Sx_1 \sim Sx_8$ および $Sy_1 \sim Sy_8$ について、近接した数列値間の差分を計算するように構成されており、その結果、符号 43、44 で示す二組の、七個の差分数 $Dx_1 \sim Dx_7$ 、 $Dy_1 \sim Dy_7$ が各々得られる。次に、これらの差分数を用いて、第一座標と第二座標を生成する。

【0223】

しかし、第一座標を計算するためには、上述のように、数列値 $Sx_1 \sim Sx_8$ のうち六個のみ、すなわち差分数 $Dx_1 \sim Dx_7$ のうち五個のみが必要である。この例によれば、数列値 $Sx_2 \sim Sx_7$ と、如いては差分数 $Dx_2 \sim Dx_6$ が使用される。これは、次に、数列値 $Sy_2 \sim Sy_7$ 、如いては差分数 $Dy_2 \sim Dy_6$ から算出される第二座標についても同様である。あるいは、各方向について、六個のみの数列値 $Sx_2 \sim Sx_7$ 、 $Sy_2 \sim Sy_7$ が決定される。

【0224】

差分数から座標への変換は多くの方法で実行することができるが、これには例えば、本願明細書中で援用により参照している同出願人による明細書 WO 01 / 26033 に記述されている方法が含まれる。

【0225】

上述した例では、記録した画像内において、データデコーディングのために 8×8 マーク要素が識別されている。しかし、これほど多くのマーク要素を識別できない場合が生じる可能性もある。次に、 8×8 マーク要素を得るために、画像内で識別できるマーク要素に「空白」の追加的マーク要素が加えられる。先に述べたように、「空白」マーク要素についての値確率は全て等しい。

【図面の簡単な説明】

【0226】

【図 1】図 1 は、本発明が使用され得るシステムの概略図である。

【図 2】図 2 は、一枚の紙の上の部分領域のデコーディングおよびマッチングを概略的に図解している。

【図 3】図 3 は、シンボルを用いて部分領域を概略的に示している。

【図4】図4は、 \times 方向についてのコーディングがどのようにして実行されるかを概略的に示している。

【図5】図5は、一次数列PDが四つの二次数列A1～A4によってどのように作られるかを概略的に示している。

【図6】図6は、位置コーディングパターンのグラフィカルコーディングの一例を示している。

【図7】図7は、拡大された位置コーディングパターンによる製品を示している。

【図8】図8は、余分なシンボルがエラー訂正の目的のために用いられるべくどのように読み取られ得るかを示している。

【図9】位置コーディングパターンがどのようにしてデコードされ得るかの一例を示すフローチャートである。

【図10】図10は、位置コーディングパターンの部分領域の画像がどのようにして位置にデコードされ得るかの一例を示すフローチャートである。

【図11a】図11aは、位置コーディングパターンを備える一枚の紙を示している。

【図11b】図11aにおける位置コーディングパターンの一部分の拡大である。

【図12】は、位置コーディングパターンにおけるマークの四つの理想的な位置を示している。

【図13】図13は、一つのマークの値確率がどのようにして計算されるかを示している。

【図14】図14a～図14cは、一つの要素についての値確率がどのようにして計算されるかを示している。

【図15】図15a～図15cは、第一および第二の行列の生成を図解している。

【図16】図16は、循環主数列を示している。

【図17】図17は、列確率の計算を図解している。

【図18】図18は、位置コーディングパターンのデコーディングのステップを図解している。

【図19】図19は、位置コーディングパターンの回転を図解している。

【図20】図20は、位置デコーディングを示すフローチャートである。

【図21】図21は、位置デコーディングを示す概略的なブロック図である。

【図22】図22は、位置デコーディングが実行され得るデバイスを概略的に示している。

【誤訛訂正3】

【訂正対象書類名】図面

【訂正対象項目名】図20

【訂正方法】変更

【訂正の内容】

【図 20】

