

明 細 書

発明の名称：二次元コード

技術分野

[0001] 本発明は、二次元コードに関する。

背景技術

[0002] 二次元コードは、一次元コードと比較して、狭い面積に多くの情報を持つことができ、物品管理や、携帯電話を用いたウェブ誘導など、様々な用途で広く活用されている。二次元コードは色々な角度で撮影されるため、二次元コードの画像は、画面における大きさおよび向きが一定ではなく、画像の歪みおよび画像のぼけ等も発生する。また、二次元コード自体が汚れている場合には、一部の画像が判別できないという事態も発生する。

[0003] 二次元コードは、一般にほぼ正方形または長方形の形状であり、元の形状に変換した上で、各セルの画素値に基づいて解析（デコード）処理を行う。二次元コードは、撮影された画像の位置関係を検出するため、所定形状の位置検出パターンを有する。解析時は、撮影した二次元コードの画像において、位置検出パターンを検出し、位置検出パターンの位置関係に基づいて変換を行う。位置検出パターンの検出は、解析の基礎となる処理であり、認識精度や、解析時間への影響が大きいため、位置検出パターンの検出処理は、解析処理の重要な要素である。

[0004] 現在普及している一般的な二次元コードは、位置検出パターンを高速で解析できるように、パターンの中心をあらゆる角度で横切る走査線において同じ周波数成分比が得られる位置検出パターンを使用し、3個の位置検出パターンを二次元コードの四隅のうち3箇所に配置している。これにより、二次元コードがどのような回転位置で撮影されても、位置検出パターンの中心を走査する場合は、暗：明：暗：明：暗＝1：1：3：1：1の長さ（周波数）比率が得られる。そのため、横方向の走査でこの明暗の比率が得られれば、次に縦（上下）方向に走査して比率を調べるという簡単な画像処理で、位

置検出パターンを高速に検出可能となる。

[0005] 上記の位置検出パターンの検出は、二次元コードがある部分の対象領域全体の画素を走査して行い、二次元コードをVGA画像で撮影した場合には、 $640 \times 480 = 307200$ 画素が処理対象領域であり、全画素を走査するのに要する時間は長くなる。そのため、位置検出パターンの検出が、解析処理時間のうち無視できないウェイトを占める。

[0006] さらに、一度に複数のコードを同時に認識させるというニーズがある。例えば、棚に二次元コードを貼った同じ向きのダンボールがあり、これの棚卸をする場合、コードをダンボールの近くに行き一つ一つ認識させるよりも、遠くから一括して認識ができると、作業能率が向上する。このような一括認識は、例えばRFIDなどを利用して行うことが知られているが、RFIDはメディア側にアンテナ等が必要で、メディアが高コストになる。これに対して、二次元コードは、紙に印字するだけで使えるため、低コストである。

[0007] 複数の二次元コードを一括して認識する場合、1画像内に複数の位置検出パターンが写り込むことになる。上記の3個の同じ位置検出パターンを有する二次元コードを使用した場合、同一の位置検出パターンが多数画像内に存在し、多数の位置検出パターンの組合せでコードの位置を検出することになる。そのため、確認する位置検出パターンの組合せ数が増加し、その分処理時間が長くなるため、素早い認識という利点が損なわれてしまう。

[0008] また、位置検出パターンを1個のみ有する二次元コードも知られており、例えば、L字型の枠と、対面に点線となるタイミングパターンの四辺から、位置と回転角度を検出するコードが知られている。このような二次元コードであれば、1画像内に複数の二次元コードが存在しても、位置検出パターンを認識できれば対応する二次元コードが認識可能である。しかし、位置検出パターンは1個であっても、位置検出パターンが複雑であるため、上記の3個の同じ位置検出パターンの認識の場合より計算量が多くなる。

[0009] また、位置検出パターンを1個のみ有する二次元コードでは、位置検出パターンを画像に写り込むノイズ情報と明確に切り分ける必要がある。特に、

汚れや画像ブレのために、1個の位置検出パターンが識別できなければ、二次元コードの解析不良ということになる。

[0010] さらに、カラー二次元コードも知られており、特許文献2は、2種類の形状の位置検出パターンを有し、それぞれに異なる色を付した二次元コードを記載している。しかし、カラー二次元コードは、撮影時の照明条件などが影響するという問題がある。

以上の通り、各種の二次元コードが提案されており、それぞれ利点および欠点を有する。

[0011] 一方で、通常の二次元コードはデータ効率を重視したため、汚れや、撮影ぶれなどに悪影響を受け易い。例えば、上記の複数の二次元コードが1つの画面に写し込まれる棚卸しの例では、すべての二次元コードに撮影のフォーカスを合わせるのは困難であり、ぼけが生じやすい。ぼけ・ぶれが生じた二次元コードの画像に対しては補正が困難なため、ソフトウェアでの改善に限界があり、オートフォーカス機能、シャッタースピードを上げるなどのハードウェアの改善で解決する必要がある、そのためにコスト増になるという問題がある。最近では、携帯電話に設けられたカメラで二次元コードを認識するというニーズがあるが、撮影時のカメラの傾き、ぶれ、フォーカスが不十分であるために起きるぼけなど、二次元コードの解析においては問題の生じやすい撮影環境であり、コスト・サイズの関係からも、携帯電話のカメラ等の改善は困難な状況である。

[0012] また、一般的な二次元コードには誤り訂正機能があるが、これはデータ部に対しての誤り訂正であり、位置検出パターン（位置決め用パターン）は訂正できないため、位置検出パターンに汚れ・欠けが発生すると認識困難で検出不可になる。

先行技術文献

特許文献

[0013] 特許文献1：特開平7-254037号公報

特許文献2：特開2004-234318号公報

特許文献3：特開2007-241328号公報

特許文献4：特開2009-163720号公報

特許文献5：特開2009-075873号公報

発明の概要

発明が解決しようとする課題

[0014] 以上説明したように、各種の二次元コードが使用・提案されているが、複数のコードを同時に撮影する場合や、ぼけ・ブレが発生する環境での撮影など、いろいろな撮影条件で撮影された場合にも正確な認識が短時間で行える二次元コードが要望されている。

[0015] 本発明は、複数のコードを同時に撮影した撮影画像に対しても短時間での認識が可能であり、さらに二次元コードの汚れ、撮影ブレなどの影響を受けにくく、各種の撮影条件で撮影された場合にも正確な認識が可能な、二次元コードの実現を目的とする。

課題を解決するための手段

[0016] 上記目的を実現するため、本発明の二次元コードは、二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置した二次元コードであって、4個以上の異なる位置検出パターンを備えることを特徴とする。ここで、位置検出パターンが異なるというのは、形状、大きさ、二次元コード上での配置方向が異なる場合であり、線幅の比率が異なる場合も含まれる。ただし、撮影条件等の影響を受けるため、色の違いは対象とせず、明暗の二値画像を対象とする。

[0017] 上記の中心を走査する場合は、暗：明：暗：明：暗＝1：1：3：1：1の比率が得られる位置検出パターンは、画素を順に走査するという簡単な画像処理で容易に検出できるという利点を有する。しかし、近年二次元コードの解析装置を形成する組込用コンピュータ（プロセッサ）の計算処理能力および記憶容量が向上し、さらに光学機器の性能が向上したため、複雑な画像処理も高速に行えるようになっている。そのため、このような性能の向上した解析装置で解析する場合に、これまで提案されている二次元コードの位置

検出パターンが、位置検出パターン検出の高速化、および検出した位置検出パターンに基づく二次元コードの画像の射影変換という点で、かならずしも好ましいとは言えない。近年、一度に複数の二次元コードを認識ことも可能になったが、これまで提案されている二次元コードは、位置検出パターンがすべて同じであるため、その組合せの検討回数が増加するなどの理由で、それに適した構造になっていない。

[0018] 本発明の二次元コードは、4個以上の異なる位置検出パターンを有する。そのため、明暗画素の配列比率から位置検出パターンを識別することはできない。しかし、画像処理の高速化により、正方形や長方形の枠など所定の形状の検出は従来に比べて容易になっている。例えば、横方向に走査を行い、暗画素に突き当たると連続した暗画素の組を検出し、形状の外周を検出することで、外形形状を容易に識別することができる。更に、内周も同様に検出すれば枠の検出が容易に行える。このような形状検出は、検出しようとするパターンがいずれの方向でも所定以上の大きさであれば、走査を間引いて行うことができ、走査数が減少する。さらに、処理済みの領域を探索対象領域から除外することもできる。そのため、形状認識の処理量が増加しても、実際の処理量の増加を抑制または減少させることが可能である。

[0019] もちろん、上記の暗：明：暗：明：暗＝1：1：3：1：1の比率が得られる位置検出パターンについて、上記のような画像処理技術を適用することも可能であるが、この位置検出パターンは、全面走査を前提として形状配置を考案されており、そのために、大きな位置検出パターンとなっている。そのために、この位置検出パターンに上記の画像処理技術を適用した場合には、位置検出パターンの利点を活かすことができず、上記の画像処理技術を適用する場合には、効率のよい位置検出パターンとは言えないという問題がある。

[0020] さらに、1画面に多数の二次元コードを有する場合、すべての位置検出パターンの組合せについて正規の組合せであるか判定していたが、各二次元コードの複数の位置検出パターンが異なる場合には、組合せ数を大幅に減少さ

せることが可能であり、その分正規の組合せであるか判定する処理量を減少させることができる。

[0021] また、位置検出パターンが異なる場合には、1個の位置検出パターンを検出した場合には、残りの位置検出パターンの方向・位置の予測が容易になり、上記の探索方法を適用する場合に走査数および連続した暗画素の組の検出処理数を減少させることも可能である。

[0022] 特に、4個以上の異なる位置検出パターンの内、少なくとも3つの位置検出パターンの内部に識別スペースを設けた枠形状とし、識別スペースが二次元コードを構成する最小セルの2倍以上のスペースであるようにすることが望ましい。これは後述する手法により、本発明の二次元コードは、例えば4つの位置検出パターンの内、3つの位置検出パターンの検出ができさえすれば、必ず二次元コードの回転角度情報が把握できて配置情報を取得できるためである。

[0023] 内部の識別スペースの有無で、位置検出パターンとその他の写り込みを区別するというのは、容易に実施でき、ぼけ・ぶれに対して効果がある。この場合、識別スペースは、2セル以上であることが望ましい。その理由は、最小セルが1ビットを表す二次元コードでは、1セル以上のぼけ・ぶれが発生する場合は、位置パターンの検出ができて、データの取得ができなくなるためである。通常の撮影条件であれば、1セル以上のぼけ・ぶれを生じることはないため、識別スペースが2セルあれば、1セル分のあらゆる方向のぼれ・ぼけに対応できるためである。これは、1画面で複数の二次元コードを撮影する場合に、1つ1つの二次元コードが小さくなり、ぼけ・ぶれの影響が大きくなり、二次元コードの画像が不鮮明になることに対しても有効である。

[0024] また、位置検出パターンを、正方形、長方形の組合せからなるようにすることが望ましい。

また、1画面に複数の二次元コードが一括して写し込まれる場合に、位置検出パターンの組合せが増加し、複数の組合せについて調べるための処理量

が増加するという課題、および1つ1つの二次元コードの画像が小さくなり、二次元コードの画像が不鮮明になるため、検出が困難になるという課題がある。上記のように、位置検出パターンを、内部に識別スペースを設けた枠形状とし、さらに正方形、長方形の組合せとすれば、位置検出パターンをあまり大きくせずにすべて異ならせることができ、しかも黒潰れが発生しにくいので、上記の2つの課題を解決することができる。

[0025] 4個以上の位置検出パターンは、例えば、正方形の第1位置検出パターンと、第1位置検出パターンより小さな正方形の第2位置検出パターンと、長方形の第3位置検出パターンおよび第4位置検出パターンと、からなる4個の位置検出パターンであり、4個の位置検出パターンは、二次元のマトリックスの四隅に配置されている。この場合、第1位置検出パターンと第2位置検出パターンは、対角上に配置され、第3位置検出パターンと第4位置検出パターンは、対角上に配置され、第3位置検出パターンおよび第4位置検出パターンの長辺は、第1位置検出パターンの辺と同じ長さであり、第3位置検出パターンおよび第4位置検出パターンの短辺は、第2位置検出パターンの辺と同じ長さであり、第3位置検出パターンおよび第4位置検出パターンの短辺の一方は、第1位置検出パターンの辺の延長上に配置され、第3位置検出パターンおよび第4位置検出パターンの長辺の一方は、第2位置検出パターンの辺の延長上に配置されることが望ましい。

[0026] 上記のような4個の位置検出パターンは、ある程度以上の大きさであれば、位置検出パターン以外に同様の形状が二次元コード中に出現する可能性が小さく、検出が比較的容易であり、色々な撮影方向から撮影されて二次元コードが歪んだ場合でも、歪みの方向の特定が容易である。

[0027] また、第1位置検出パターンおよび第2位置検出パターンは、それぞれ一重の正方形枠を有し、第3位置検出パターンおよび第4位置検出パターンは、それぞれ一重の長方形枠を有してもよい。

枠であれば、位置検出パターン以外に同様の形状が二次元コード中に出現する可能性がさらに小さくなり、検出が比較的容易である。

[0028] また、第1位置検出パターンは、一重の正方形枠と、正方形枠内に配置された正方形を有し、第2位置検出パターンは、一重の正方形枠を有し、第3位置検出パターンおよび第4位置検出パターンは、それぞれ一重の長方形枠を有してもよい。

[0029] これは、第1位置検出パターンを上記の暗：明：暗：明：暗＝1：1：3：1：1の長さ比率が得られる位置検出パターンと類似のパターンとするものであり、面積が増加し、検出処理に要する時間が長くなる。しかし、1個であるので面積の増加が小さく、検出処理に要する時間は若干長くなるが、位置検出パターン以外に同様の形状が二次元コード中に出現する可能性はほとんどなく、検出精度を向上できる。

[0030] さらに、二次元のマトリックスの4個以上の位置検出パターン以外の領域は、複数の同じ大きさのブロックに分割され、二次元のマトリックスは、隣接するブロック間に、明または暗の分離スペースを有することが望ましい。したがって、分離スペースは、1セル幅のブロック間隔の格子状パターンになる。この場合、4個以上の位置検出パターンは、ブロックより大きな面積を有することが望ましい。

[0031] これにより、位置検出パターンと同じパターンが、二次元コード中に出現することはなくなり、位置検出パターンの検出が容易になる。さらに、各ブロックにおけるセルの明暗の判定精度が向上する。ブロックは、例えば、3×3のセルを有し、分離スペースは、例えば、1セルの幅を有し、4個以上の位置検出パターンは、例えば、4セル以上の横幅と、4セル以上の縦幅と、を有することが望ましい。

バージョン情報ブロックおよびフォーマット情報ブロックは、位置検出パターンの近傍に設けることが望ましい。

[0032] 上記の4個以上の異なる位置検出パターンを有する二次元コードを撮影した画像から、二次元コードを解析する解析方法は、4個以上の異なる位置検出パターンを識別し、識別した4個以上の異なる位置検出パターンの位置関係を確認した後、確認した位置関係に基づいて二次元コードの画像を射影変

換し、射影変換した二次元コードの画像を解析するが、4個以上の異なる位置検出パターンのうち識別した位置検出パターンが2個または3個の時は、識別した2個または3個の位置検出パターンの位置関係から、残りの位置検出パターンの位置を推測し、推測した位置検出パターンの位置を含めて少なくとも4個の位置検出パターンの位置関係に基づいて二次元コードの画像を射影変換するようにする。

[0033] これにより、位置検出パターンの一部が汚れ等で識別できない場合でも解析に成功する可能性が向上する。

また、二次元コードは、通常位置補正パターンを含むので、4個以上の異なる位置検出パターンの位置関係の確認および識別した位置検出パターンが2個または3個の時の残りの位置検出パターンの位置の推測に、位置補正パターンを合わせて利用することにより、解析精度が向上する。

[0034] 上記の解析方法を実行する解析装置およびコンピュータに上記の解析方法を実行させることも可能である。

[0035] また、本発明の二次元コードの作成システムは、3個の異なる位置検出パターンを、二次元のマトリックスの所定位置に配置する位置検出パターン配置手段と、二次元のマトリックスの3個の異なる位置検出パターンを配置する領域以外の領域に、二次元コードの解析に必要なベース情報を記録したベース情報領域を配置するベース情報配置手段と、メッセージの記録データを、二次元のマトリックスの3個の異なる位置検出パターンおよびベース情報領域を除いた領域に、順次配置するメッセージデータ配置手段と、を備えることを特徴とする。ここで、ベース情報は、例えば、バージョン情報、フォーマット情報、絵柄埋込情報などであり、位置補正パターンや誤り訂正符号などを含めてもよい。

発明の効果

[0036] 本発明の二次元コードを使用することにより、解析時間を長くすること無しに、各種の劣化が生じた二次元コードの撮影画像であっても、短時間で正確な解析が可能である。

特に、本発明の二次元コードを使用することにより、複数の二次元コードを一括で撮影した撮影画像を解析する場合であっても、膨大な数になる位置検出パターンの組合せを減らすことが可能な構成にすることで、短時間で解析が可能となる。

さらに、その場合に起こるぶれや、ぼけ等の劣化が生じた撮影画像であっても、解析時間を長くすること無しに、精度の高い解析が可能となる。

図面の簡単な説明

[0037] [図1]図1は、特許文献1に記載された二次元コードを示す図である。

[図2]図2は、第1実施形態の二次元コードを示す図である。

[図3]図3は、第1実施形態の二次元コードにおける複数のブロックの配置を示す図である。

[図4]図4は、分離スペースを有する二次元コードと、分離スペースを有さない二次元コードにおいて、データに応じてセルを明または暗にした場合のパターンの例を示す図である。

[図5]図5は、第1実施形態におけるデータブロックのデータ種別を説明する図である。

[図6]図6は、20パターン表現のデータ種別を説明する図である。

[図7]図7は、第1実施形態の二次元コードにしたがって作成した二次元コードの例を示す図である。

[図8]図8は、第1実施形態の二次元コードにしたがって作成した二次元コードの別の例を示す図である。

[図9]図9は、第2実施形態の二次元コードを示す図である。

[図10]図10は、第2実施形態の二次元コードにしたがって作成した二次元コードの例を示す図である。

[図11]図11は、第2実施形態の二次元コードの位置検出パターンを、サイズおよび形状の異なる二次元パターンに適用した変形例を示す図である。

[図12]図12は、変形例の4個の位置検出パターンの例を示す図である。

[図13]図13は、依頼に基づいて二次元コードを作成して提供する作成シス

テムのハードウェア構成を示す図である。

[図14]図14は、ユーザーが、ユーザーハードウェアを介してシステムハードウェアにアクセスし、所望の二次元コードを作成するエンコード処理の手順を示すフローチャートである。

[図15]図15は、ユーザーが、ユーザーハードウェアを介してシステムハードウェアにアクセスし、所望の二次元コードを作成するエンコード処理の手順を示すフローチャートである。

[図16]図16は、ユーザーが、ユーザーハードウェアを介してシステムハードウェアにアクセスし、所望の二次元コードを作成するエンコード処理の手順を示すフローチャートである。

[図17]図17は、第1実施形態の二次元コードを読み取り、二次元コードを解析する二次元コード解析装置のハードウェア構成を示す図である。

[図18]図18は、ユーザーが撮影した二次元コードを解析するデコード処理の手順を示すフローチャートである。

[図19]図19は、ユーザーが撮影した二次元コードを解析するデコード処理の手順を示すフローチャートである。

[図20]図20は、ユーザーが撮影した二次元コードを解析するデコード処理の手順を示すフローチャートである。

[図21]図21は、位置検出パターン候補の形状チェックを説明する図である。

発明を実施するための形態

[0038] 本発明の実施形態を説明する前に、現在広く使用されている一般的な二次元コードについて説明する。

図1は、特許文献1に記載された二次元コードを示す図である。

[0039] 図1に示すように、二次元コード1は、3つの隅に配置された3個の位置検出パターン（位置決め用シンボル）2と、データ領域3と、を有する。データ領域3は、二次元コード1の領域から、3個の位置検出パターン2およびその周囲のスペーサ部分を除いた部分である。各位置検出パターン2は、

暗部分である正方形の枠 2 a および正方形の枠 2 a 内に配置された正方形 2 c と、枠 2 a と正方形 2 c の間の明部分である正方形の枠 2 b と、を有する。撮影された位置検出パターン 2 は、走査線が中心を通過する時、走査線がいかなる方向であっても、その長さ比率（周波数成分比）は暗：明：暗：明：暗＝1：1：3：1：1 となる。そのため、二次元コードをいかなる方向に回転した状態で撮影しても、一定方向の走査処理のみで位置検出パターン 2 の持つ特定周波数成分比を検出することができる。したがって、位置検出パターン 2 の中心位置を容易に検出することができる。

[0040] データ領域 3 は、マトリックス状に密に配置されたビットを有し、ビットマトリックスは、データ領域内のドットの位置補正のための位置合わせパターン（タイミングセル）4 に加えて、二次元コードの形式情報およびその誤り訂正信号と、型番情報およびその誤り訂正信号と、データ部分と、に分けられる。位置合わせパターン 4、二次元コードの形式情報およびその誤り訂正信号、および型番情報およびその誤り訂正信号は、コード領域 3 のビットマトリックス上の所定の位置に配置され、残りの部分がコード化したデータを記録するデータ部で、必要なデータを記録して残った部分が残余ビットである。記録するデータビットと残余ビットは、データビットの最後のエンドパターンで区別される。残余ビットは、埋め草と称され、明（“1”）のビット列であるが、記録時には所定の処理を施されて明暗ビット列に変換されるのが一般的である。

[0041] 位置合わせパターン 4 は、データ領域内のドットの位置補正のための明暗のビット（ドット）列であり、図 1 に示すように、多数個が配置される。位置合わせパターン 4 は、二次元パターン全体の基準位置を検出する位置検出パターン 2 とは別のものである。

特許文献 3 および 4 は、埋め草であるデータ領域 3 のビットを利用して、ビット画像を設けることを記載している。

[0042] 二次元コードの認識では、まず位置検出が行えないとコードが認識できないため、位置検出パターンの認識精度を高めることが重要となる。そのため

の方策として、次の3つの手法が考えられる。

(1) コード全体を大きくする。

(2) 位置検出パターンを認識しやすくする。

(3) 位置検出パターンの訂正できるように、言い換えれば位置検出パターンに冗長性を持たせる。

[0043] (1) のコード全体を大きくすることについては、二次元コードを設ける紙面のデザインや、媒体の制限（小さな部品にコードをつけたい）など、二次元コードの印字スペースは小さいことが求められ、面積が小さいことが二次元コードの利点の1つである。よって、(1) の手法は望ましくない。

[0044] (2) の認識しやすい位置検出パターンについては、二次元コードのサイズに対して位置検出パターンを相対的に大きくして認識しやすくすることが考えられる。しかし、二次元コードのサイズが同じ場合、位置検出パターンを大きくすると、その分データ領域が狭くなり、データ量が少なくなるため、望ましくない。

[0045] 図1の二次元コード1の位置検出パターン2は、画像の走査線がいかなる方向であっても所定の周波数成分比が得られるように、正方形とその周囲に設けた正方形枠とを有する二重形状になっている。この形状自体は、他のビットパターンとの区別が容易であるという利点があるが、小さいパターンの場合には、正方形と正方形枠の間の明の部分がつぶれて認識できないため、画像にぼけ・ぶれがあっても容易に認識できるようにするには、ある程度の大きさにする必要があり、その分データ領域が狭くなるため、データ効率が低下する。

[0046] さらに、図1の位置検出パターン2は、全面走査を前提として形状配置を考案されており、必ずしも位置補正パターンの検出時間が短く、効率的な解析を行うのに適しているとは言えない。

[0047] 位置検出パターンは、単純な（シンプルな）形状にすると相対的に大きくしてもセルサイズを少なくできるが、その場合周囲の写り込みを拾いやすくなり、位置検出パターンを誤検出する可能性が高くなり、この場合確認する

必要のある位置検出パターンの組合せが増える。そのため、位置検出パターンは、単純ではあるが、ぼけ・ぶれの影響を受けにくく、他の写り込みと区別しやすい形状であることが望ましい。

[0048] (3) の位置検出パターンに冗長性を持たせるということは、位置検出パターンを複数個設け、そのうちの 1 個または 2 個以上の位置検出パターンが識別できない場合でも、識別した残りの位置検出パターンで二次元コード内の位置検出パターンの基準位置を訂正可能にすることである。

[0049] 例えば、図 1 の二次元コード 1 は、3 個の位置検出パターンを 3 箇所の隅（コーナー）に設け、1 箇所の隅には設けないようにすることで二次元コードの回転位置を特定できるようにしている。しかし、3 個の位置検出パターンのうち 1 個でも識別できないと二次元コードの回転位置を特定できず、認識に失敗する。位置検出パターンの補完を行えたとしても、二次元コードの検出処理時間が大幅に増加し、ななめからの撮影や、歪み、ぼけなどで正確な座標が得にくいため、データ抽出の精度に大きく影響する。

[0050] 位置検出パターンの個数を増加させれば、位置検出パターンの基準位置は訂正可能になる。例えば、位置検出パターンが 5 個以上あれば、1 個欠けても、4 個以上の位置検出パターンを識別できれば、データの座標のマッピング（画像とコードの位置関係の決定）が精度よく行える。

[0051] 位置検出パターンが多ければ多いほど冗長性が増し、汚れなどへの耐性が増すが、検討すべき組合せがその分多くなってしまい、処理時間が長くなると共に、二次元コードにおける位置検出パターンの占める面積が増加し、データ効率が低下する。

[0052] 図 1 の位置検出パターン 2 は、画像の走査線がいかなる方向であっても所定の周波数成分比が得られるが、形状を変更すると所定の周波数成分比が得られないため、3 個の位置検出パターン 2 は、同じ形状にする必要がある。そのため、図 1 の二次元コード 1 の位置検出パターン 2 を異ならせるには、大きさを変える必要がある。位置検出パターン 2 は、正方形とその周囲に設けた正方形枠を有するため、元々大きなパターンであり、もし 3 個または 4

個のパターンの大きさを、画像認識で異なると判定できるように変える場合には、もっとも小さなパターンもある程度の大きさにする必要があり、もっとも大きなパターンは非常に大きくなる。このように、図 1 の位置検出パターン 2 は、冗長性を持たせるには適していない。

[0053] なお、冗長性を持たせる場合、1 個の位置検出パターンが識別できない場合だけでなく、2 個以上の位置検出パターンが識別できない場合も考慮することが望ましい。特許文献 2 は、正方形とその周囲に設けた正方形枠とを有する 1 個の大型位置検出パターンおよび正方形枠を有する 3 個の小型位置検出パターンを四隅に配置し、中央に 1 個の小型位置検出パターンを配置した二次元コードを記載している。ここで、中央の小型位置検出パターンは、データの座標のマッピングへの寄与が少ないので、使用しない場合を考える。この場合、大型位置検出パターンと 1 個の小型位置検出パターンが識別できず、2 個の小型位置検出パターンのみが識別できた場合には、データの座標のマッピングは行えない。

[0054] 以上のように、位置検出パターンに冗長性を持たせる場合、異なる位置検出パターンの個数が増加するほど冗長性が向上する。

以下、本発明の実施形態を説明する。

[0055] 図 2 は、第 1 実施形態の二次元コードを示す図である。

第 1 実施形態の二次元コードは、 35×35 セルの大きさを有し、 9×9 のブロックに分割され、隣接するブロック間に 1 セルの明の分離スペース 23 を有する。したがって、分離スペースは、 8×8 列の 1 セル幅のブロック間隔の格子状パターンになる。右下の 3×3 ブロックおよび 3×3 分離スペースの部分には第 1 位置検出パターン 12A が、左上の 2×2 ブロックおよび 2×2 分離スペースの部分には第 2 位置検出パターン 12D が、右上の 3（横） $\times$ 2（縦）ブロックおよび 3×2 分離スペースの部分には第 3 位置検出パターン 12B が、左下の 2（横） $\times$ 3（縦）ブロックおよび 2×3 分離スペースの部分には第 4 位置検出パターン 12C が、それぞれ設けられている。したがって、二次元コードの第 1 から第 4 位置検出パターン以外のブロ

ックに、第1から第4位置検出パターン以上のパターンが出現することはない。

[0056] 二次元コードのコード部分の最小単位はセルである。二次元コードは、通常正方形または長方形の形状をとる。他の形状も可能だが、二次元コードの多くは四角形であり、第1実施形態の二次元コードも正方形であるが、これに限定されるものではなく、長方形でも他の形状でもよい。

[0057] 一般に、二次元コードは明暗のセルでデータを表し、第1実施形態の二次元コードもコード部は明暗のセルでデータを表す。なお、特許文献2に記載されたような色によりセルを区別する二次元コードや、ミラー（左右反転）型の二次元コードも提案されており、そのような二次元コードに本発明を適用することも可能であるが、その場合にも、後述するように、4個以上の形状の異なる位置検出パターンを備える。

[0058] さらに、第1実施形態の二次元コードは、 35×35 セルの正方形であるが、これよりサイズの大きなコードや、長方形のような縦横の大きさを変えた形状にすることも可能である。 35×35 セルの正方形の二次元コードを、これをバージョン 2×2 と呼ぶ。ここで、最小のバージョンは 1×1 であり、この時のセルサイズは 27×27 となる。絵柄の埋込は、縦横のバージョンが共に2以上の場合に可能とする。バージョンが 1×1 の際に絵柄を埋め込まない理由は、データ量が少ないにも関わらず、絵柄を埋め込むと、さらにデータ量が少なくなり、認識してもほとんどデータが得られないことになるため、上記のような制限を加えた。第1実施形態の二次元コードは、縦方向および横方向にそれぞれ伸張できる。横バージョンが2、縦バージョンが3の場合、バージョン 2×3 と表記する。バージョンが1上がるごとに、セルサイズが8大きくなり、ブロック数が2大きくなる。よって、バージョン 3×4 では、セルサイズは 43×51 になる。バージョンは1から20まで選択可能であるが、これに限定されるものではない。

[0059] 第1位置検出パターン12Aは、暗セルの部分が、 8×8 セルの1セル幅の正方形枠と、正方形枠の中心の 2×2 の正方形と、を有する。第2位置検

出パターン 1 2 D は、暗セルの部分が、 4×4 セルの 1 セル幅の正方形枠を有する。第 3 位置検出パターン 1 2 B は、暗セルの部分が、 8×4 セルの 1 セル幅の長方形枠を有する、横長の長方形枠である。第 4 位置検出パターン 1 2 C は、暗セルの部分が、 4×8 セルの 1 セル幅の長方形枠を有する、縦長の長方形枠である。

[0060] 第 1 実施形態では、正方形の二次元コードの 4 隅に 4 個の異なる位置検出パターンを配置する。なお、位置検出パターンが異なるというのは、形状、大きさ、二次元コード上での配置方向が異なる場合であり、線幅の比率が異なる場合も含まれる。ただし、撮影条件等の影響を受けるため、色の違いは対象とせず、明暗の二値画像を対象とする。

[0061] 第 1 実施形態では、位置検出パターンは、正方形または長方形の各頂点付近に配置する。これは、1 箇所配置する場合より、できるだけ離れて配置する方が汚れの影響を分散できるためである。ただし、位置検出パターンの配置位置は、4 隅に限定されるものではない。第 1 実施形態のように、位置検出パターンとして、正方形だけでなく、長方形のものをを用いることで、形状・傾き・大きさの区別が容易にできるため、高速に認識できる。また中抜きした長方形は、他の写り込みと区別するのが容易であり、データの無駄が少ない。

[0062] 4 個の位置検出パターンを配置する理由は、解析において精度高くデータの座標のマッピングを計算するのに用いる二次元の射影変換では、コードのセルと、そのセルに対応する撮影画像の座標を計算するために、4 組の座標の組をパラメータとして与える必要がある。位置検出パターン数が 2 個または 3 個しかない場合でも、座標を補間することで、4 点を算出した上で射影変換を行うことは可能であるが、ななめからの撮影や、歪み、ぼけなどで正確な座標が得にくく、データ抽出の精度に大きく影響するため、第 1 実施形態では 4 個の位置検出パターンを配置している。

また、位置検出パターンが異なる理由は、見つからない位置検出パターンがあった場合でも、見つからなかった位置検出パターンが確実に特定でき、

見つからなかった位置検出パターンの座標の予測が可能になり、位置検出パターンの訂正が可能になる。すべて同じ位置検出パターンを用いた場合、見つからなかった位置検出パターンの特定は、すべて異なる位置検出パターンを用いた場合ほど容易ではなく、位置検出パターンの訂正が困難になる。

二次元コードの4個の位置検出パターン以外の56ブロックは、データ部であり、実データブロック、誤り訂正符号ブロック、バージョン情報ブロック、フォーマット情報ブロック、および絵柄埋込情報ブロックと、を含む。

[0063] 上記のように、データ部は 3×3 セルを1ブロックとし、ブロック単位でデータを埋め込む。データ部は、バージョン情報ブロック15および16と、フォーマット情報ブロック17と、絵柄埋込情報ブロック18A、18B、19Aおよび19Bと、を有し、残りのブロックが実データブロック22である。なお、後述するように、第1実施形態では、実データブロック22の一部を、誤り訂正符号と、位置補正パターンとして利用する。

[0064] したがって、位置検出パターン、バージョン情報ブロック、フォーマット情報ブロック、絵柄埋込情報ブロック、位置補正パターンを設けるブロック、さらに誤り訂正符号ブロックを除いた、残りのブロック22が、実データブロックである。もし、絵柄の埋め込みが行われる場合には、絵柄埋込情報ブロックが指示する絵柄埋め込みブロックも実データブロックから除く。実データは、実データブロックに属するブロックについて、左上から右に順にデータを敷き詰め、敷き詰め終わったら1ブロック下の左から敷き詰めるという具合に順に記録する。

[0065] ブロックは、 3×3 セルを有し、各セルは明と暗の1ビットを表すので、1ブロックは最大9ビットの情報を持つ（9ビット表現）が、ここでは画像のぼけ・ぶれに対する耐性を高めるため、1ブロック当たりのデータ量を9ビットのデータ量より小さく設定できるようにしている。なお、第1実施形態では、ブロックを 3×3 セルのサイズとしたが、ブロックサイズはこれに限定されるものではなく、最小である 1×1 セルのサイズや、縦と横のセルサイズが異なるなどのブロックサイズも可能である。

- [0066] バージョン情報ブロックは、二次元コードのサイズに関する情報を記録し、縦バージョン情報ブロック 15 と、横バージョン情報ブロック 16 の 2 種類存在し、それぞれ対応するバージョン情報が記録される。縦バージョン情報ブロック 15 および横バージョン情報ブロック 16 は、それぞれ同じ情報を記憶した 2 個のブロックを離れた 2 箇所配置することにより、汚れ等で欠損するリスクを分散させる。バージョン情報として、バージョン数（縦横各 1 から 20）までの数値を表すブロックを配置する。
- [0067] フォーマット情報ブロック 17 は、絵柄埋込フラグ（1 ビット）と、データブロック種別データ（2 ビット）からなるフォーマット情報を記録し、同じ情報を記憶した 3 個のブロックを離れた 3 箇所に配置する。絵柄埋込フラグは、絵柄埋込の有無、すなわち絵柄埋込領域が設けられているか否かを示す。絵柄埋込が有の場合、絵柄埋込情報ブロックが設けられる。データブロック種別データは、データブロックの種別を示し、データブロックの種別とは、1 ブロック当たりのデータ容量であり、6、7、9 ビット／ブロックのいずれかを選択する。データ量は 9 ビット > 7 ビット > 6 ビット表現の順に少なくなるが、データの読み取りのしやすさは、6 ビット > 7 ビット > 9 ビットの順であり、ぼけやぶれの発生具合と、データ量を勘案して、選択する。これについては後述する。
- [0068] 埋め込まれる絵柄は、ブロック単位で埋め込まれ、隣接した複数のブロックが絵柄埋込領域である場合には、その間の分離スペースも絵柄埋込領域に含まれる。埋め込まれる絵柄の外形は四角形であり、絵柄の画像は、セル単位の 2 値画像に限定されず、どのようなものでもよい。また、絵柄を埋め込む四角形のブロックは、複数個あってもよい。例えば、絵柄は、セルより高分解能の多値画像でも、カラー画像でもよく、もちろんセル単位の 2 値画像でもよい。第 1 実施形態では、絵柄埋め込みの制限として、コード上側ブロック 2 行、左側ブロック 2 列、下側ブロック 3 行、右ブロック 3 列には絵柄を埋め込まないものとする。この範囲は二次元コードの仕様により、絵柄の埋め込み範囲外の部分に画像の情報を持つ絵柄埋込情報ブロックを配置する

。

[0069] 絵柄埋込情報ブロックは、絵柄を埋め込む領域のサイズおよびブロック位置に関する情報を縦横それぞれで記録し、縦埋込情報ブロック 18 A および 18 B と、横埋込情報ブロック 19 A および 19 B の 2 種類存在し、それぞれ対応するバージョン情報が記録される。縦埋込情報ブロック 18 A および 18 B は、絵柄埋め込み高さ（3 ビットまたは 6 ビット）および絵柄埋込オフセット高さ（3 ビットまたは 6 ビット）を記録する。横埋込情報ブロック 19 A および 19 B は、絵柄埋込幅（3 ビットまたは 6 ビット）および絵柄埋込オフセット幅（3 ビットまたは 6 ビット）を指定する。なお、3 ビットまたは 6 ビットにするかは、後述するようにバージョン情報に関係し、バージョンが 2 ～ 4 の場合には 3 ビット、バージョンが 5 ～ 20 の場合には 6 ビット必要であり、バージョン 2 ～ 4 の場合には、縦埋込情報ブロック 18 B および横埋込情報ブロック 19 B は不要である。

[0070] 絵柄埋込情報ブロックは、フォーマット情報ブロックで絵柄埋込フラグが立っていた場合のみ存在する。第 1 実施形態は、埋込絵柄の個数が 1 つのみの場合を想定しており、複数の埋込絵柄を複数領域に埋め込むことを想定した場合には、フォーマット情報ブロックで絵柄埋込フラグのビット数を増加させて絵柄埋込領域数を示すようにすると共に、上記の絵柄埋込情報ブロックを絵柄枚数分設ける。絵柄のサイズと絵柄の埋込開始位置は、ブロック単位で指定する。ブロックの座標を左上隅のブロックを原点として (n, m) と表現し、横ブロック数を $b \mid W$ 個、縦ブロック数を $b \mid H$ 個とする。第 1 実施形態では、絵柄の埋め込み領域は、左上 $(2, 2)$ 、右下 $((b \mid W - 4, b \mid H - 4))$ からなる矩形の領域に限定される。絵柄の最大横ブロック数は $b \mid W - 5$ 個、最大縦ブロック数は $b \mid H - 5$ 個となる。

[0071] 絵柄の位置は、始点（左上）となるブロック座標と、ブロックサイズの情報を絵柄埋込情報ブロックとしてコード内に埋め込むことで、コード解析時にどの位置に絵柄が入っているかを認識できるようにする。バージョン情報ブロックと同様に、横と縦で別々に扱う。横は $b \mid W - 5$ が 8 未満の時（横

バージョン2～4)と、それ以外の場合(横バージョン5～20)でデータ量を分け、縦も同様にする。bIWと、bIHの値は、横と縦のバージョンを解析することで求められる。

[0072] bIWが13未満の時、横のブロックサイズは最小1、最大8ブロックであり、8通りは3ビットで表すことができ、同様に横のオフセットブロック数は2から8まで7通りあり、これも3ビットで表せるため、合計が6ビットとなる。そのため、ブロックを6ビット表現し、これらは1ブロックで表現可能である。縦も同様にする。bIWが13以上の時、横のブロックサイズは6ビットで表し、同様に横のオフセットブロック数も6ビットで表す。よって、2ブロックを使って表現することになる。縦も同様にする。

[0073] 横埋込情報ブロック19Aおよび19Bは、左上および右下の横バージョン情報ブロックの外側(上側及び下側)に、縦埋込情報ブロック18Aおよび18Bは、左下および右下の縦バージョン情報ブロックの外側(左側および右側)に配置する。

もし絵柄を埋め込まない場合、すなわちフォーマット情報ブロックの絵柄埋込フラグが「無し」を示す場合、絵柄埋込情報ブロックおよび絵柄埋め込みの領域が不要になるため、コードが効率的になる。

[0074] 位置補正パターンは、実データブロック、誤り訂正符号ブロック、絵柄が埋め込まれている場合には絵柄埋込情報ブロックの座標補正に用いる目的のパターンである。二次元コードにおける概ねの座標は位置検出パターンから取得できるが、紙のよれ、曲がり、レンズの歪み、位置検出パターン取得時のズレによりデータ部の座標に誤差がでるので、これを修正する目的で設ける。位置補正パターンは、位置検出パターンと異なり、他のノイズと区別する必要がなく、位置検出パターンが算出できれば位置補正パターンの位置は容易に判明する。したがって、位置補正パターンは細かな座標補正がしやすい形状が望ましい。

[0075] 第1実施形態では、位置補正パターンは、決められたセルの組合せからなるブロック(位置補正ブロック)を実データの代わりに等間隔に配置する。

ブロックの中心 1 セルを暗、その周囲を明にしたブロックを、ブロック座標 $(7n, 7m)$ (n, m は 0 以上の整数) がデータ部であれば配置する。ただし、その座標に、位置検出ブロック、バージョン情報ブロック、フォーマット情報ブロック、絵柄埋込情報ブロックおよび絵柄が存在すべき時は、配置しない。ここでは、ブロック座標は、ブロックの左上の座標基準座標を $(0, 0)$ とし、右隣のブロックを $(1, 0)$ と表現している。なお、位置補正パターンを設けるブロックは、上記に限られない。

[0076] 実データは、メッセージと、メッセージを修飾するヘッダー（メッセージ種類（メッセージのエンコード）・メッセージサイズ）を合わせたセグメントを、メッセージ数だけ並べる。また特殊なセグメントとして、メッセージを含まない、終端フラグだけが入ったセグメントを用意し、実データの容量が余った場合に終端フラグのセグメントを配置し、その後に埋め草を配置する。実データはデータブロック種別データが示す 1 ブロック当たりのデータ容量に従い、ブロック単位に分割する。同様に誤り訂正符号も、ブロック単位に分割する。

誤り訂正符号にリードソロモン符号を使った場合、誤り訂正はワード単位で行われるため、1 ワードを 1 ブロックとすることが望ましい。1 ワードが複数のブロックにまたがる場合、1 つのブロックに汚れが発生した場合でも、そのブロックが関係する全てのワードが誤り訂正の対象となり、訂正の効率が悪くなる。訂正原因となる汚れや、スポットライトによる色とびは、1 箇所集中することが多いが、ブロックにすることにより同時に訂正対象となるデータを 1 箇所にまとめる効果があり、効率的な訂正を可能にし、コードを認識させる可能性を高める。

[0077] 上記のように、第 1 実施形態の二次元コードは、ブロック間に設けられた 1 セル幅の分離スペース 2 3 を有する。したがって、位置検出パターンおよびブロックは、すべて周囲を明のセルで囲まれている。

図 3 は、第 1 実施形態の二次元コードにおける複数のブロックの配置を示す図である。各ブロックは 3×3 のセルを有し、周囲が 1 セル幅の分離ス

ース23で隔てられている。

[0078] 図4は、分離スペース23を有する二次元コードと、分離スペース23を有さない二次元コードにおいて、データに応じてセルを明または暗にした場合のパターンの例を示す図である。図4の(A)は、分離スペース23を有する二次元コードで、4個のブロックの4個の3×3セルを、中央および4隅が暗、それ以外を明として場合のパターン3つと、その3つのパターンと明暗が反転したパターン1つからなる二次元コードで、図4(B)は、(A)に示した二次元コードの分離スペース23がない場合を示したものである。このように分離スペース23がない場合は、ブロック間の境目が不明確になり、暗セルに囲まれた明セルは黒潰れしてしまう可能性がある。

[0079] 分離スペース23を有さない二次元コードに比べて、分離スペース23を有する二次元コードの方が大きくなり、データ効率は低下するが、撮影した画像にぼけ・ぶれがある場合でも、隣接するブロックのパターンの影響を受けにくくなるので、各セルの明暗判定の精度が向上する。実際に、分離スペース23を有する二次元コードと分離スペース23を有さない二次元コードの画像を二値化処理した場合、分離スペース23を有さない二次元コードは黒潰れが発生しやすい。撮影した画像にぼけ・ぶれがある場合、各セルは周囲のセルの色に影響を受ける。図4の(B)は周囲のセルが明暗どちらであるかはほぼランダムであり、周囲のセルが異なる色が大きければ大きな影響があるが、そうでなければ影響は少なく、一様でないため、色の判定は困難となる。図3のようにブロックを構成するセルをa~iとして説明すると、図4(A)は分離スペース23があることからa c g iは周囲8箇所中5箇所が分離スペースと接し、b d f hは周囲8箇所中3箇所が分離スペースと接するため、e以外のセルは図4の(B)より周囲の影響を受けにくくなったといえる。さらに後述する6ビット表現にすることで、影響を受けやすいeのセルを評価しなくなり、a c g iは周囲8箇所中6箇所が明セルと接し、b d f hは周囲8箇所中4箇所が明セルと接するため、よりぼけ・ぶれの影響を受けにくくなったといえる。

[0080] このように、分離スペース23を設けることで、セルがぼけ・ぶれにより正しく認識できなくなるリスクを低減することが可能となる。

[0081] なお、位置検出パターンは、最もサイズの小さい第2位置検出パターン12Dでも、4×4セルの正方形であり、3×3ブロックが分離スペース23で分離されているため、データブロック中に位置検出パターンと同じパターンが出現することはない。したがって、分離スペース23を設けることにより、位置検出パターンの検出処理も容易になる。

[0082] 第1実施形態では、1ブロック当たりのデータ容量を、データ劣化の耐性に応じた種別に分けて選択可能にしている。

[0083] 図5は、第1実施形態における1ブロックのデータ種別を説明する図である。

図5の(A)は、1ブロックの3×3のセルのそれぞれa～iのビットを割り当て、9ビット表現可能にしたデータ種別を示す。この場合、1ブロックは、9ビットの情報量を有し、 $2^9=0\sim511$ までの値を表現できる。

図5の(A)の解析を行う際は、入力された撮影画像から作成された二値画像に対して、各セルが対応する画像内の座標を計算し、その座標の画素の値が明(“1”)と暗(“0”)のいずれかを判断し、ブロックあたり9ビットとなる値にする。撮影画像を二値画像に変換する方法は後述する。

[0084] 図5の(B)は、1ブロックの3×3のセルの中心を“1”(明)に固定し、周囲の8個のセルj1～j8は、4個の明(“1”)と4個の暗(“0”)を含むように設定するデータ種別を示す。この場合、1ブロックは、 ${}_8C_4=70$ パターン、およそ6ビットの情報量を有する。(6ビット表現)

[0085] 図5の(C)は、1ブロックの3×3のセルの中心を“0”(暗)に固定し、周囲の8個のセルj1～j8は、4個の明(“1”)と4個の暗(“0”)を含むように設定するデータ種別を示す。この場合も、1ブロックは、 ${}_8C_4=70$ パターン、およそ6ビットの情報量を有する。(6ビット表現)

データ種別が6ビット表現である時、(B)と(C)のいずれも使うことができるが、エンコード側とデコード側で一致させる。

図5の(B)(C)の解析を行う際は、入力された撮影画像から作成されたグレースケール画像に対して、 $j1 \sim j8$ のセルの撮影画像内での座標を計算し、各セルのグレースケールの画素値を取得し、画素値を降順にソートする。このとき、画素値の高いセル4つが明であり、低いセル4つが暗である。撮影画像をグレースケール画像に変換する方法は後述する。

二次元コードは様々な撮影環境下での認識が求められる。その際に、光は撮影画像に対して、局所的な色の変化、段階的な色の変化など、悪い影響を与え、これは撮影画像に対する適切な二値化画像の作成が困難である要因となる。

この方法の利点は、二値化閾値が不要な点である。二値化閾値を導出しなくても、各セルの相対的な画素値の関係より、セルの明暗を判定することを可能とする効果がある。また、スポットライトのような局所的な強い色の変化に対しても、ブロックにすることで関係するセルが密集するため、全セルが影響を一樣に受ける確率が高くなり、セル全体が白くなった場合でも、相対的な画素値の大小関係は維持されるため、認識できる可能性が高くなる。

[0086] なお、図5の(B)および(C)で、中心のセルは“1”(明)または“0”(暗)に任意に設定可能で、周囲の8個のセル $j1 \sim j8$ を、4個の明(“1”)と4個の暗(“0”)を含むように設定するデータ種別も可能であり、この場合は、1ブロックは、 ${}_8C_4 \times 2 = 140$ パターン、およそ7ビットの情報量を有する。(7ビット表現)

この解析方法について、周囲のセルの解析の方法までは6ビットの場合と同じである。これに加え、中心画素の解析は、画素値が高い4点のうち画素値が最小となるセルの画素値を $G1$ 、画素値が低い4点のうち画素値が最大となるセルの画素値を $G2$ とし、ブロックの中心のセルの撮影画像内での座標を計算し、得られた画素値が $(G1 + G2) / 2$ 以上であれば中心のセルを明セル、それ以外であれば中心のセルを暗セルとする。

この方法では、中心のセルの評価に隣接するセルの解析から得られた閾値を用いており、中心セルと近接するセルは撮影環境から受ける影響がほぼ同

じである可能性が高いため、精度が高く認識できる。

[0087] 図5の(D1)～(D8)は、1ブロックにおいて、中心セルを明か暗に固定し(ここでは明(“1”)に固定)、周囲の8個のセルを明暗4セルずつにするが、時計回りに明暗暗明明明暗暗に固定し、孤立した明セルの位置でデータを取り出す。この場合、8パターン取り出せるので、3ビットのデータ量になる。(3ビット表現)このパターンで、孤立した明セルの位置を検出するには、開始位置のセルから時計回りに、各セルの輝度値を、加算、減算、減算、加算、加算、加算、減算そして減算する演算処理を、開始位置を順に回転しながら行い、演算結果が最大になる開始位置を求める。これが孤立した明セルの位置である。同様に、明暗反転して行うこともでき、この場合は演算結果が最小になる開始位置を求める。

[0088] 前述のように、第1実施形態では、フォーマット情報ブロック17内のデータブロック種別データ(2ビット)により、上記の9ビット表現、7ビット表現、および6ビット表現のいずれかが選択される。必要があれば、3ビット表現を選択できるようにしてもよい。

[0089] なお、第1実施形態では、重要な情報を記録するバージョン情報ブロックおよびフォーマット情報ブロックは、次に説明する20パターン表現のデータ種別を利用する。

[0090] 図6は、20パターン表現のデータ種別を説明する図である。

図6において、参照番号31は、20パターン表現のデータブロックを、32は第1から第4のいずれかの位置検出パターンを示す。図6の(A)および(B)に示すように、位置検出パターン32の下側または上側に配置されるデータブロック31は、上下方向の中央の列のセルを、暗明暗(010)に固定し、両側の2列のセルm1からm6の6ビットでデータを表現する。さらに、6ビットのうち3ビットを明、3ビットを暗にすることにより、 $C_3=20$ パターンを表現する。図6の(A)および(B)に示したブロックのセルデータの検出は、上下方向に走査することにより行う。また、図6の(C)および(D)に示すように、位置検出パターン32の右側または左側

に配置されるデータブロック 31 は、横方向の中央の列のセルを、暗明暗（010）に固定し、両側の 2 列のセル m1 から m6 の 6 ビットでデータを表現する。さらに、6 ビットのうち 3 ビットを明、3 ビットを暗にすることにより、 ${}_6C_3 = 20$ パターンを表現する。図 6 の（C）および（D）に示したブロックのセルデータの検出は、横方向に走査することにより行う。

このデータ種別により、解析の際にコードのバージョン、およびセルサイズが不確かな状態でも、位置検出パターンの撮影画像内での位置が判明していれば、データブロックからデータを得ることができる。バージョン、およびセルサイズが不明である場合、位置検出パターンの撮影画像内での座標がわかっているにもかかわらずブロックの座標が算出できない。しかし、位置検出パターンから、隣接する位置検出パターン（第 1 位置検出パターンと第 2 位置検出パターンは、第 3 位置検出パターンと第 4 位置検出パターン隣接する）の方向を走査することで、データブロックが見つかる。図 6 の（A）では、データブロックの中央の列に向かい、参照番号 32 の位置検出パターンの下側の辺に対して垂直に走査を行う。位置検出パターンのすぐ下から走査を行い、中央の列の上側のセルである暗セルにぶつかる。このまま進めると、明セルにぶつかり、そして暗セルにぶつかり、さらに分離スペースの明セルにぶつかる。これにより中央の列のセルの座標と、データブロックの 1 セルあたりの画素数が算出できる。これにより、データブロックの他のセルの座標も容易に判明することから、データブロックが表すデータが得られる。

[0091] バージョン情報ブロックおよびフォーマット情報ブロックは、解析処理において重要なデータブロックであり、位置検出パターンに対して図 6 に示すような位置関係になるように配置して、20 パターン表現を行う。情報ブロックは 2 箇所、フォーマット情報は 3 箇所に同じ情報を保持することで、汚れや、スポットライトなどの局所的なノイズへの対策を行っている。

[0092] 図 7 は、第 1 実施形態の二次元コードにしたがって作成した二次元コードの例を示す図である。図 7 の二次元コードは、 43×43 セルで、 11×11 ブロックの二次元コードであり、バージョンは 3×3 となる。図 7 の二次

元コードは、絵柄領域を有さず、絵柄も埋め込まれていない。したがって、絵柄埋込情報ブロックも不要である。

[0093] 図8は、第1実施形態の二次元コードにしたがって作成した二次元コードの別の例を示す図である。図8の二次元コードは、59×59セルで、15×15ブロックの二次元コードであり、バージョンは5×5となる。図8の二次元コードは、ブロック座標(2, 5)と(11, 8)を対角とする長方形の絵柄領域を有し、IDコードと物品名が記載された画像が付されている。これにより、ユーザーは、コードを認識する前でも認識対象の概要を判別することが可能である。

[0094] 位置検出パターンは、各種の形状のものを使用することが可能であり、二次元コードのサイズ等も色々な値が取り得る。

図9は、第2実施形態の二次元コードを示す図である。第2実施形態の二次元コードの第1実施形態の二次元コードとの相違点は、次の通りである。

[0095] (1) 第1位置検出パターン52Aが、暗セルの正方形枠のみで内側に正方形を有さない。また、第2位置検出パターン52Bおよび第3位置検出パターン52Cは、正方形に近い長方形である。第4位置検出パターン52Dは、第1実施形態の12Dと同じ形状である。

[0096] (2) ブロックサイズおよびブロック構成が異なり、特に横方向および縦方向が位置検出パターンに挟まれた外側部分のブロック構成が、異なる。この部分に配列されるブロックは1列で、その部分に、2個の縦バージョン情報ブロック55、2個の横バージョン情報ブロック56、3個のフォーマット情報ブロック57、2個の縦絵柄埋込情報ブロック58、2個の横絵柄埋込情報ブロック59、および位置補正パターン54Aおよび54Bを設ける。残りのブロック60が、実データブロック、誤り訂正符号ブロック、位置補正パターンとなるブロック、および絵柄埋込領域である。

[0097] 他の部分は、第1実施形態と同じであるので、説明は省略する。

図10は、第2実施形態の二次元コードにしたがって作成した二次元コードの例を示す図である。

[0098] 図10の(A)は、 36×36 セルで、 8×8 ブロックの二次元コードである。図10の(A)の二次元コードは、ブロック座標(2, 2)と(5, 5)を対角とする長方形の絵柄領域を有し、高分解能の多値画像が付されている。これにより、二次元コードが与える印象を改善し、注目を集めることができる。

[0099] 図10の(B)は、 36×36 セルで、 8×8 ブロックの二次元コードである。図10の(B)の二次元コードは、ブロック座標(2, 3)と(5, 4)を対角とする長方形の絵柄領域を有し、数字の画像が付されている。

[0100] 二次元コードのサイズおよび形状は各種あり、そのいずれにも本発明の二次元コードを適用することが可能である。

[0101] 図11は、第2実施形態の二次元コードの位置検出パターンを、サイズおよび形状の異なる二次元パターンに適用した変形例を示す図である。参照番号53は、データ領域を示す。

[0102] 図11の(A)は、正方形の小型の二次元コードの例を示す。

図11の(B)は、第2実施形態と類似の二次元コードの例を示す。

図11の(C)は、長方形の二次元コードの例を示す。

[0103] 図11の(D)は、長方形の二次元コードで、中心部に第5位置検出パターン52Eを追加した二次元コードの例を示す。第5位置検出パターン52Eは、正方形枠と、その内部に正方形を有する第1実施形態の第1検出パターン12Aに類似の形状を有する。

[0104] ここで、位置検出パターンとして好ましい形状について検討する。

本発明での位置検出パターンは、(1)4種の位置検出パターン同士で、形状・大きさ・二次元コード上での配置方向のいずれかで区別が可能であること、(2)位置検出パターン同士で、種類の区別が簡易なアルゴリズムで実現可能であること、(3)区別した位置検出パターンについて、組合せの絞込みが容易であること(高速化に貢献)、(4)位置検出パターンのセルサイズを抑えること、(5)二次元コード以外の写り込んだ画像と区別しやすいこと、および(6)二次元コード内に他に位置検出パターンが表れない

ことが望まれる。

図2および図9の位置検出パターンは、以上の事項を満足するといえる。

[0105] ここで、正方形および長方形の位置検出パターンで、枠の幅および識別スペースの幅、すなわちセルサイズについて、色々な組合せの位置検出パターンを作成して、ぼけ・ぶれを生じるように撮影した画像で二値化を行い、位置検出パターンの区別が行えるか実験を行った。その結果、枠の内部の識別スペースの有無で位置検出パターンとその他の写り込んだ画像とを区別するというのは、容易に実施できること、1セルのスペース、その中の1セルの正方形は、ぼけに影響を受けやすいこと、外枠の太さはぼけの耐性に大きく影響しないこと、が判明した。

[0106] このような結果から、正方形または長方形の枠で、内部に識別スペースを有し、識別スペースが2セルであるものの有用性が判明した。この理由は、次のように考えられる。最小セルが1ビットを表す二次元コードでは、1セル以上のぼけ・ぶれが発生する場合は、位置検出ができて、データの取得ができなくなる。通常の撮影条件であれば1セル以上のぼけ・ぶれが生じることは少ないため、識別スペースが2セルあれば、1セル分のあらゆる方向のぼけ・ぶれに対応できることとなる。

[0107] 第1の実施形態に示した、セルサイズが4×4の正方形である第2位置検出パターンが、上記のスペースを持つ、最小セルサイズの構成となる。

第1位置検出パターンについては、1セルの枠と2セルのスペースと2セルの正方形を重ねたものとなる。この位置検出パターンであれば、その向きによらず、中心を走査すると暗明暗明暗のパターンが得られ、第2位置検出パターンや、ノイズとの違いが明確である。

[0108] 第3位置検出パターンおよび第4位置検出パターンの形状は、位置検出パターンの枠の延長線に位置検出パターンの枠が存在することが望ましい。これにより、位置検出パターンが欠けた際に、欠けた位置検出パターンの座標を予測する精度が向上する効果が得られる。

これらの条件に当てはまる形状は、第1実施形態の第3位置検出パターン

、および第4位置検出パターンとなる。それぞれ90度傾いており、第1位置検出パターン、および第2位置検出パターンとは縦横比が異なることから区別が容易である。

[0109] 第2実施形態は、最も位置検出パターンが占めるセルサイズが少ない構成であり、第1位置検出パターンと第2位置検出パターンは、同じ形状であり、大きさが異なり、第2位置検出パターンを 5×5 セルにし、上記と同様の理由で第3位置検出パターンと、第4位置検出パターンを作成する。

これら位置検出パターンのセルサイズは、第1実施形態では、 $16 + 64 + 32 + 32 = 144$ セルとなり、第2実施形態では、 $16 + 25 + 20 + 20 = 81$ セルとなる。

図1に示し二次元コードは、3箇所には位置検出パターンが無いにもかかわらず、 $7 \times 7 \times 3 = 147$ セルで、本実施例よりセルサイズが大きい。

[0110] 第1実施形態は、第2実施形態と比べ、セルサイズが大きくなるが、第2実施形態では全ての位置検出パターンの面積が近く、斜めからの撮影で精度への影響を受けやすいのに比べ、第1実施形態は、第1位置検出パターンと、第2位置検出パターンは形状が異なり、第3位置検出パターンと、第4位置検出パターンは向きが異なり、第1検出パターン・第2位置検出パターンと、第3検出パターン・第4位置検出パターンは縦横比が異なる。これらは斜めからの撮影でも、充分区別可能であり、位置検出パターンの区別に有用な構成となる。

[0111] また、図2の位置検出パターンは周囲2セルに余白として明セルを配置している。図1の二次元コードの認識の際は、コードを印字した周囲にクワイエットゾーンが必要であり、これはコードの一部と考えられる。これに対して、第1実施形態の二次元コードは、コードの周囲はどのような印刷が行われようが精度に影響しない。また、図9の第2実施形態のように、余白は図1のコード同様にコードの外に配置することも可能である。

[0112] しかし、本発明の主旨からは、他の形状の位置検出パターンも使用でき、各種の変形例が可能である。

図 1 2 は、変形例の 4 個の位置検出パターンの例を示す図である。

図 1 2 の (A) は、大きな円枠、方向の異なる 2 個の楕円枠および小さな円枠が、4 個の位置検出パターンである例を示す。

[0113] 図 1 2 の (B) は、内部に小さな円を有する大きな円枠、方向の異なる 2 個の楕円枠および小さな円枠が、4 個の位置検出パターンである例を示す。

図 1 2 の (C) は、大きな十字形、延長方向が異なる一方のみ延長した 2 個の十字形および小さな十字形が、4 個の位置検出パターンである例を示す。

[0114] 図 1 2 の (D) は、大きな L 字形、延長方向が異なる一方のみ延長した 2 個の L 字形および小さな L 字形が、4 個の位置検出パターンである例を示す。ここで、L 字は同じ方向であり、角の位置から回転方向を判定することが可能である。なお、L 字の方向を 90 度ずつ回転したものにしてもよい。

[0115] 図 1 2 の (E) は、大きな三角枠、延長方向が異なる一方のみ延長した 2 個の三角枠および小さな三角枠が、4 個の位置検出パターンである例を示す。

図 1 2 に示した例は、一部の例であり、他にも各種の形状の位置検出パターンが可能である。

[0116] 次に、第 1 および第 2 実施形態の二次元コードを作成する処理（エンコード処理）について説明する。

図 1 3 は、依頼に基づいて二次元コードを作成して提供する作成システムのハードウェア構成を示す図であり、クライアント・サーバ構成の例を示す。

[0117] 作成システムは、仕様を決めて二次元コードの作成を依頼するユーザーが操作するユーザーハードウェアと、依頼された二次元コードを作成して提供するシステムハードウェアと、を有する。

[0118] ユーザーハードウェアは、コンピュータ等のユーザー処理装置 6 1 と、磁気ディスク等の記憶装置 6 2 と、を有する。

システムハードウェアは、コンピュータ等のシステム処理装置 6 5 と、磁

気ディスク等の記憶装置 66 と、を有する。

[0119] ユーザー処理装置 61 とシステム処理装置 65 は、通信回線等で接続され、通信可能に構成されている。

ここでは、印刷は、ユーザー側で行うが、システム側や他の印刷所で行うようにしてもよい。二次元コードを印刷する媒体はどのようなものでもよく、例えば、紙、樹脂板、筐体表面などである。ここで、媒体として、埋め込む絵柄があらかじめ印刷されているものでもよく、印刷されている埋込絵柄が二次元コードの絵柄領域に入るようにセッティングした上で二次元コードを印刷する。

[0120] 印刷装置は、これらの媒体に二次元コードを印刷可能なものであればよく、例えば簡易プリンタ、精密プリンタ、印刷装置等であり、モノクロ印刷だけでなく、カラー印刷可能なものでもよい。また、作成した二次元コードは、印刷せずに、通信回線を介してユーザーに二次元コードのデータとして送信されてもよい。ユーザーは、必要に応じて第三者のディスプレイ等に、作成した二次元コードを表示するようにデータを送信する。

[0121] なお、図 13 はクライアント・サーバ構成の作成システムの例を示すが、作成システムはこれに限定されず、例えば、クライアント PC 上のエンコードソフトウェアで発行し、USB 接続したプリンタから発行する構成や、ハンディタイプの端末・プリンタから発行する構成など、各種の変形例があり得る。

[0122] 図 14 から図 16 は、ユーザーが、ユーザーハードウェアを介してシステムハードウェアにアクセスし、所望の二次元コードを作成するエンコード処理の手順を示すフローチャートである。

[0123] ステップ S10 では、ユーザーがエンコードのメイン処理を開始する。

ステップ S11 では、ユーザーが二次元コードに記録するメッセージを入力する。

ステップ S12 では、ユーザーが二次元コードに絵柄の埋め込みを行うか否かを決定し、絵柄の埋め込みを行う場合にはステップ S13 に進み、行わ

ない場合にはシステム側で行うステップS 1 5に進む。この際、ユーザー処理装置6 1は、システム処理装置6 5に入力されたメッセージ、および絵柄の埋め込みを行わないこと、を通知する。

[0124] ステップS 1 3では、ユーザーが、絵柄を埋め込むブロック位置（絵柄埋込オフセット幅、絵柄埋込オフセット高さ、絵柄埋込幅および絵柄埋込高さ）を入力する。ユーザー処理装置6 1は、システム処理装置6 5に、入力されたメッセージ、絵柄の埋め込みを行うこと、および絵柄埋込ブロック位置を通知する。

[0125] ステップS 1 4では、システム側で、送信されたブロック位置に基づいて絵柄のブロックサイズを決定する。

ステップS 1 5では、送信されたメッセージを記録するブロックサイズを決定する。

[0126] ステップS 1 6では、すべての横バージョンに関して最小縦バージョンを計算する。これは、縦および横のブロック数を乗じたものから、位置検出パターン、バージョン情報ブロック、フォーマット情報ブロック、絵柄を埋め込む場合には絵柄埋込情報ブロックおよび絵柄埋込領域に応じたブロック、位置補正パターンのブロックを除いた残りのブロックを計算する。さらに、このブロック数に対して、一定の誤り訂正率になるような誤り訂正符号ブロックの数を除いた残りのブロックに、メッセージを記録する実データブロックが収容できるサイズをテーブルの形で提示するものである。システム処理装置6 5は、計算したテーブルのデータを、ユーザー処理装置6 1に送信する。

[0127] ステップS 1 7では、ユーザーに縦・横バージョンのテーブルを提示し、ユーザーがいずれを使用するかを決定する。ユーザー処理装置6 1は、決定した縦・横バージョンを、システム処理装置6 5に送信する。

ステップS 1 8では、システム側で、送信された縦・横バージョンと、絵柄埋込領域に応じたブロックサイズから、実データのブロックサイズを決定する。

[0128] ステップS 19では、システム側で、送信された縦・横バージョンと、絵柄埋込領域に応じたブロックサイズから、誤り訂正符号ブロックサイズを決定する。

ステップS 20では、システム側で、バージョン情報ブロック、フォーマット情報ブロック、絵柄を埋め込む場合には絵柄埋込情報ブロックおよび絵柄埋込領域ブロックの位置を決定する。

[0129] ステップS 21では、システム側で、絵柄埋込を行うか否かが判定され、絵柄埋込を行う場合にはステップS 22に進み、行わない場合にはステップS 23に進む。

ステップS 22では、絵柄埋込情報ブロックを、コード画像部分の所定ブロック位置に配置する。

[0130] ステップS 23では、位置検出パターン・位置補正パターンを画像部分に配置する。

ステップS 24では、バージョン情報ブロック・フォーマット情報ブロックを画像部分に配置する。

[0131] ステップS 25では、メッセージごとにメッセージヘッダーを付与する。

ステップS 26では、まだデータを入れる部分があるか判定し、あればステップS 27に進み、無ければステップS 29に進む。

ステップS 27では、終端メッセージヘッダー（終端フラグ）をメッセージデータの末尾に付与する。

[0132] ステップS 28では、埋め草を残りの実データ部に付与する。

ステップS 29では、実データをブロックにして、画像部分に配置する。

ステップS 30では、誤り訂正符号ブロックを画像部分に配置する。

[0133] ステップS 31では、システム処理装置65が、上記のようにして作成したコード画像を出力し、ユーザー処理装置61に送信する。

ステップS 32では、ユーザー側で、絵柄埋め込みを行うか否かを判定し、絵柄埋め込みを行う場合にはステップS 33に進み、行わない場合にはステップS 34に進んで終了する。

[0134] ステップS 3 3 では、送信されたコード画像で、絵柄埋込領域のブロックに埋め込む絵柄データを合成して全体の画像を生成する。絵柄の埋め込みは、例えばデータ添付という形でシステム処理部において行ってもよい。

ステップS 3 4 で、エンコードのメイン処理が終了する。

上記の例では、ユーザー側が絵柄データを有していたが、システム側が絵柄データを有しており、それをユーザーが指定するようにしてもよい。

[0135] 図 1 7 は、第 1 実施形態の二次元コードを読み取り、二次元コードを解析する二次元コード解析装置のハードウェア構成を示す図である。

[0136] 二次元コード解析装置は、読み取り部 7 0 と、コンピュータ（二次元コード解析処理部） 7 4 と、ディスプレイ 7 5 と、通信インターフェース 7 6 と、を有する。読み取り部 7 0 は、レンズ 7 1 と、イメージセンサ 7 2 と、アナログ・デジタル変換器（A D） 7 3 と、を有し、撮影した二次元コードのデジタル画像データをコンピュータ 7 4 に出力する。図 1 7 の二次元コード解析装置は、広く使用されており、近年は携帯端末も二次元コード解析装置と同様の機能を実現している。

[0137] 図 1 8 から図 2 0 は、ユーザーが撮影した二次元コードを解析するデコード処理の手順を示すフローチャートである。このデコード処理は、1 画面に複数の二次元コードが写し込まれている場合を想定している。デコード処理は、解析メイン処理と、情報取り出し処理からなる。まず、解析メイン処理を説明する。

[0138] ステップS 1 0 1 では、解析のメイン処理を開始する。

ステップS 1 0 2 では、二次元コードの撮影画像を入力する。

[0139] ステップS 1 0 3 では、入力された撮影画像の二値画像を作成する。二値化の方法は、入力された撮影画像が R G B 画像などのカラー画像であれば、一旦グレースケール画像に変換し、画像内の最大輝度値と最小輝度値の平均を閾値とし、閾値以上であれば明、未満であれば暗とする。カラー画像からグレースケール変換するには、各画素の R G B の値を用いて、輝度(brightness) = $0.299R + 0.587G + 0.114B$ の変換式にしたがって変換

する。カラー画像へのグレースケール画像への変換方法、さらに二値化画像への変換方法は、様々な方法が提案されており、上記変換方法に限られない。

[0140] ステップS104では、位置検出パターン候補を検出する。具体的には、画像の左上の点を始点とし、画面をX方向に走査し、右端にたどり着いたら、数ピクセル下の左端からX方向に走査する。言い換えれば、数ピクセル分間引くことになる。これを画像全体に対して繰り返す。間引くピクセル数は、 $(\text{最大間引き量}) = (\text{位置検出パターンの最小セル高さ}) \times (\text{撮影時のセルあたりのピクセル数})$ である。位置検出パターンの最小高さとは、コードの回転も考慮し、位置検出パターンの回転で起こり得るY方向の最小セルサイズであり、実施形態では左上の第2位置検出パターンの回転しない状態での4セルが最小となる。撮影時のセルあたりのピクセル数は、1セル当たり1ピクセルを下回ると、理想画像でもセルを明と暗を判別することが困難になる。実際の撮影画像でのセルのずれや回転を考慮すると、少なくとも1セル当たり3ピクセル以上あるのが望ましい。3ピクセルとすると、上記の場合最大間引き量は12ピクセルとなり、画像走査回数は全画素数の $1/12$ となり、走査数を大幅に削減できる。

[0141] ステップS105では、位置検出パターン候補を検査する。具体的には、走査中に明から暗の境界を見つけた場合、暗部分の周囲を走査する。その際に、最小・最大のX座標、Y座標をそれぞれ求める。これにより、暗の固まりの外接する長方形が取得できる。これの左上座標を (x_1, y_1) 、右下座標を (x_2, y_2) と表す。1度周囲を走査すれば、以後走査はしなくてよい。

[0142] なお、周囲探索を行っても、十分に画像走査回数を減らすことができる。周囲探索でもっとも処理時間が長くなる最悪ケースの一例は、1ピクセルの縦縞を1ピクセルごとに配置するようなケースであるが、そもそもそのようなケースは発生しない上、発生しても周囲長の探索は全画素を1回走査するのと同じ回数で抑えることができ、走査回数が増加することはない。これは

他の最悪ケースでも同様である。

[0143] 図1の二次元コードの場合、暗明暗明暗が1:1:3:1:1の比率になる部分を探すことにより位置検出パターンを検出でき、1回の走査で位置検出パターンを見つけることができる。ここで、撮影したコードが45度傾いている際の走査で、1:1:3:1:1の比率が得られるのは、対角線上の走査だけである。多くの場合、撮影の向きが自由であり、そのため、撮影するコードの向きが360度のいずれの回転位置もあり得る。このような場合、間引くことができる量は少なく、実際の回転位置は分からないので、間引き量を大きくすることはできない。

[0144] 図21は、位置検出パターン候補の形状チェックを説明する図である。

次に、位置検出パターン候補の形状をチェックする。第1実施形態の二次元コードの位置検出パターンは、いずれも枠、すなわち穴形状を有する。そこで、まず穴の有無を確認する。始点を外接長方形の左側中点($x_1, (y_1 + y_2) / 2$)、終点を($x_2, (y_1 + y_2) / 2$)とするx軸方向の走査を行う。ここで、次の(1)または(2)の条件にあてはまることを確認する。

[0145] (1) 暗明暗となる並びが存在することを確認する。1つ目の暗の左端($l \times 1$)から2つ目の暗の右端($r \times 1$)までの中心となる画素が明であることを確認する。

(2) 暗明暗明暗というパターンを探す。1つ目の暗の左端($l \times 2$)から3つ目の暗の右端($r \times 2$)までの中心となる画素が暗であることを確認する。

[0146] (1)の場合はさらに、図21の(A)に示すように、中心座標($(l \times 1 + r \times 1) / 2, (y_1 + y_2) / 2$)から上下を走査し、上が明暗明となっており(終点: $((l \times 1 + r \times 1) / 2, y_1 - 1)$)、下が明暗明(終点: $((l \times 1 + r \times 1) / 2, y_2 + 1)$)であり、上側走査の暗と明の境界のy座標 $t y_1$ と下側走査の暗と明の境界のy座標 $b y_1$ の中点が、明の中心座標とほぼ一致することを確認する。

[0147] (2) の場合もさらに、図 21 の (B) に示すように、中心座標 $((l \times 2 + r \times 2) / 2, (y_1 + y_2) / 2)$ から上下を走査し、上が暗明暗明となっており (終点: $((l \times 2 + r \times 2) / 2, y_1 - 1)$)、下が暗明暗明 (終点: $((l \times 2 + r \times 2) / 2, y_2 + 1)$) であり、上側走査の 2 つめの暗と明の境界の y 座標 $t y_2$ と下側走査の 2 つめの暗と明の境界の y 座標 $b y_2$ の中点が明の中心座標とほぼ一致することを確認する。

[0148] 以上の処理により、枠内に暗部を有する第 1 位置検出パターン (正方形枠内に正方形) と、枠のみの第 2 から第 4 位置検出パターン (正方形枠または長方形枠: 口型) と、に区別できた。

[0149] (1) または (2) に当てはまらないものは、位置検出パターンではない。

次に、各位置検出パターン候補の頂点を探索する。外接する長方形の左上の座標 (x_1, y_1) から X 軸方向に (x_2, y_1) まで走査する。一番初めに接触した暗点の座標を $vertex 1: (v x_1, v y_1)$ とする。次に右上の座標 (x_2, y_1) から Y 軸方向に (x_2, y_2) まで走査する。一番初めに接触した暗点の座標を $vertex 2: (v x_2, v y_2)$ とする。右下の座標 (x_2, y_2) から X 軸の反対方向に (x_1, y_2) まで走査する。一番初めに接触した暗点の座標を $vertex 3: (v x_3, v y_3)$ とする。次に左下の座標 (x_1, y_2) から Y 軸の反対方向に (x_1, y_1) まで走査する。一番初めに接触した暗点の座標を $vertex 4: (v x_4, v y_4)$ とする。 $vertex 1 \sim 4$ は長方形 (正方形を含む) となるはずである。そうでない場合は、位置検出パターンではない。図 21 の (B) の表記は省略する。

[0150] 次に長方形の縦横比を見る。 $((v x_1 - v x_2) \times (v x_1 - v x_2) + (v y_1 - v y_2) \times (v y_1 - v y_2))^{1/2}$ で 1 辺の長さが、 $((v x_2 - v x_3) \times (v x_2 - v x_3) + (v y_2 - v y_3) \times (v y_2 - v y_3))^{1/2}$ でもう 1 辺の長さが、求められる。辺の比率が、(長辺) : (短辺) = 1 : 1 でかつ形状が (1) であれば第 2 位置検出パターン、(2) であ

れば第1位置検出パターン、(長辺) : (短辺) = 1 : 2で形状が(1)であれば第3または第4位置検出パターンとなる。ここまでの、3種類の位置検出パターンの区別が可能となった。

[0151] 次に面積を求める。ここでの面積は、外側の長方形枠の面積で、内側のスペースの面積も含む。第1位置検出パターンの場合は、枠の中の正方形も含む。位置検出パターンの面積は、外側の長方形の面積から4隅を含む4つの直角三角形の面積を減じれば求まる。面積 = $(x_2 - x_1) \times (y_2 - y_1) - (v_{x1} - v_{x4}) \times (v_{y4} - v_{y1}) / 2 - (v_{x2} - v_{x1}) \times (v_{y2} - v_{y1}) / 2 - (v_{x2} - v_{x3}) \times (v_{y3} - v_{y2}) / 2 - (v_{x3} - v_{x4}) \times (v_{y3} - v_{y4}) / 2$ で求まる。

[0152] 次に、回転角度を求める。第1および第2位置検出パターンは、正方形なので90度以上の傾きを検知できないので、90度まで求める。角度は時計回りとする。

$$\theta = \arctan((v_{y2} - v_{y1}) / (v_{x2} - v_{x1}))$$

θ が回転角度となる。

[0153] 第3および第4位置検出パターンは、0から180度まで求める。第3位置検出パターンは、2点間座標の距離を求める関数 $\text{length}(x_1, y_1, x_2, y_2) = ((x_2 - x_1)^2 + (y_2 - y_1)^2)^{1/2}$ を定義し、0~45度までは $\text{length}(v_{x1}, v_{y1}, v_{x2}, v_{y2}) > \text{length}(v_{x4}, v_{y4}, v_{x1}, v_{y1})$ かつ $(x_2 - x_1) > (y_2 - y_1)$ であり、 $\theta = \arctan((v_{y2} - v_{y1}) / (v_{x2} - v_{x1}))$ で求まる。45~90度までは $\text{length}(v_{x1}, v_{y1}, v_{x2}, v_{y2}) > \text{length}(v_{x4}, v_{y4}, v_{x1}, v_{y1})$ および $(x_2 - x_1) < (y_2 - y_1)$ であり、 $\theta = \arctan((v_{y2} - v_{y1}) / (v_{x2} - v_{x1}))$ で求まる。90~135度までは $\text{length}(v_{x1}, v_{y1}, v_{x2}, v_{y2}) < \text{length}(v_{x4}, v_{y4}, v_{x1}, v_{y1})$ および $(x_2 - x_1) < (y_2 - y_1)$ であり、 $\theta = \arctan((v_{y2} - v_{y1}) / (v_{x2} - v_{x1})) + 90$ 度で求まる。135~180度までは

$length(vx1, vy1, vx2, vy2) < length(vx4, vy4, vx1, vy1)$ および $(x2 - x1) > (y2 - y1)$ であり、 $\theta = \arctan((vy2 - vy1) / (vx2 - vx1)) + 90$ 度で求まる。第4位置検出パターンについても同様に回転角度 θ を求める。

[0154] 次に、位置検出パターン候補の中心座標を検出する。中心座標は外接長方形の中心 $((x1 + x2) / 2, (y1 + y2) / 2)$ とする。

以上で、位置検出パターン候補の検査が終了する。

[0155] ステップS106では、既に3種類に区別されている位置検出パターンを、区別ごとに、位置検出パターンを面積で降順にソートする。

[0156] ステップS107では、4種の位置検出パターンの組合せを作成する。まず、第1位置検出パターン候補の中で最も面積の大きな位置検出パターン候補を選択する。第1位置検出パターンの面積を S とすると、面積が $S / 4$ である第2位置検出パターン候補で、かつ回転角度が同じ組合せを探す。これが見つかり、第3および第4位置検出パターン候補で面積が $S / 2$ となる候補を探し、さらに角度が同じ候補と、90度異なる候補の組合せを探す。

[0157] ここで、1画面に複数の二次元コードが写し込まれている場合、位置検出パターン候補のあらゆる組合せを考えると、莫大な組合せになってしまう。本発明のように位置検出パターンを全て変えることにより組合せを大きく減らせ、さらに事前に得られた形状・面積・回転などの特徴を用いることで、簡単な比較演算で、さらなる組合せの絞り込みが可能となっている。これについては後述する。

[0158] ステップS107では、検討していない4種の位置検出パターンの組合せが残っているか判定し、残っていればステップS108に進み、残っていなければステップS112に進む。

[0159] ステップS108では、組合せた位置検出パターンについて、同一の二次元コードの位置検出パターンであるとして情報の取り出しを行う。この処理については、図19および図20を参照して後述する。

[0160] ステップS109では、情報取り出しに成功したかの結果に応じて、成功

した場合にはステップS 1 1 0に進み、失敗した場合には失敗した位置検出パターンの組合せを除外する処理を行った後、ステップS 1 0 7に戻る。

[0161] ステップS 1 1 0では、取り出しに成功した情報のメッセージを保存する。

ステップS 1 1 1では、データの取り出しに成功した二次元パターンに使われた候補を位置検出パターン候補から除外し、ステップS 1 0 7に戻る。なお、データの取り出しに成功した二次元パターンの範囲に、使用していない位置検出パターン候補がある場合には、それも候補から除外する。

[0162] ステップS 1 0 7からS 1 1 1を繰り返すことにより、写し込まれた二次元コードの解析が終了する。ここで、除外されていない位置検出パターン候補が残っている場合には、3種または2種の組合せについて、ステップS 1 0 8の情報の取り出し処理を行うようにしてもよい。これについては後述する。

[0163] ステップS 1 1 2では、解析が終了したとのメッセージを呼出元に送信する。

ステップS 1 1 3で、解析メインの処理を終了する。

[0164] 次に図19および図20を参照してステップS 1 0 8の情報取り出し処理を説明する。

ステップS 2 0 0で、情報取り出し処理を開始する。

ステップS 2 0 1で、位置検出パターン4種の組合せの関係をチェックする。チェックは以下の(a)(b)の条件を満たすか判定することで行い、満たさない項目があれば、問題ありとして処理を終了する。

[0165] まず、前記第1位検出パターンの中心座標をp t 1、前記第2位検出パターンの中心座標をp t 2、前記第3位検出パターンの中心座標をp t 3、前記第4位検出パターンの中心座標をp t 4とし、p t 1、p t 3、p t 2、p t 4、p t 1と順番に各点を結ぶ。位置検出の4種の組合せが正しければ、長方形が得られる。

[0166] (a) 全ての頂点の角度が直角になる。

- [0167] (b) 向かい合う辺の長さが 1 : 1 になる。
- [0168] ステップ S 2 0 2 では、以上のチェック結果で問題があるか判定し、あればステップ S 2 0 3 に進み、情報取り出しに失敗したことを示すフラグを立て、問題が無ければステップ S 2 0 4 に進む。
- [0169] ステップ S 2 0 4 では、いずれかの位置検出パターンから、1 セルあたりのピクセル数を求める。第 2 位置検出パターンの場合、面積は $4 \times 4 = 16$ セル分の面積であり、1 セルあたりのピクセル数 = $((\text{面積}) / 16)^{1/2}$ の式で求まる。
- [0170] ステップ S 2 0 5 では、隣接する位置検出パターンの中心座標の距離を縦横それぞれで計測し、ステップ S 2 0 4 で求めた値で割ることにより、二次元コードの縦・横のセルサイズをそれぞれ求めることができる。得られた中心座標の横の距離を d ピクセル、1 セルあたりのピクセル数を p ピクセル毎セルとすると、コードの横のセルサイズ = $d / p + (4 + 6)$ が得られる。縦も同様の方法。
- [0171] ステップ S 2 0 5 では、ステップ S 2 0 5 で求めたセルサイズから二次元コードのバージョンを仮決定する。バージョンは 8 セルずつ段階的に大きくなる。コードの横のセルサイズ = 横バージョン $\times 8 + 19$ で求められるため、この式を変形させ横バージョン = $(\text{セルサイズ} - 19) / 8$ で求められる。ここで得られた横バージョンの値が、横仮決定バージョンとなる。セルサイズが一致するバージョンがない場合は、最もセルサイズに近いバージョンを指定する。縦も同様の方法。仮とする理由は、斜めから撮影や、印字した紙の曲がりなどで、正確な値が得られない可能性を考慮しており、バージョン情報ブロックもあわせて利用することで、正しい値を得るようにしている。
- [0172] ステップ S 2 0 6 では、位置検出パターンから、隣接する位置検出パターンの方向を走査することで、バージョン情報ブロック、フォーマット情報ブロックを構成するセルの撮影画像での座標を検出する。
- [0173] ステップ S 2 0 7 では、バージョン情報ブロックの解析（縦横）を行う。

これにより、縦横のバージョンが判明する。解析に失敗した場合、処理を終了する。前述のように、バージョン情報ブロックおよびフォーマット情報ブロックは、1ブロックで20通りの値を示す。バージョン情報ブロックは、縦・横それぞれブロックが2箇所配置されており、縦・横2箇所ずつのバージョン情報ブロックから読み取った値と、コード幅・高さ位置検出パターンのサイズから得られる仮決定バージョン情報、これら3つの値のうち2つ以上同じ値があれば、その値を採用する。

[0174] ステップS208では、フォーマット情報ブロックの解析を行う。絵柄埋め込みの有無と、ブロックごとのデータ埋め込み方が決定する。解析に失敗した場合、処理を終了する。フォーマット情報ブロックは、3箇所は位置されており、2箇所以上同じであれば採用する。

[0175] ステップS209では、位置補正パターンの座標を計算し、各セルの二次元コードの画像内での座標を高精度に求める。各セルの座標は既に導出できているが、位置補正パターンにより、レンズの歪みの影響を軽減することが可能となる。既に位置補正パターンの画像内での座標は判明しているので、その点が暗であることを確かめる。そして、この点を始点とし、右方向に走査を行い、暗と明の画素の境界を見つけ、S105同様の方法で周囲を走査し、外接する長方形の左上座標を (x_1, y_1) 、右下座標を (x_2, y_2) とし、中心座標は $((x_1 + x_2) / 2, (y_1 + y_2) / 2)$ が得られた。

[0176] ステップS210では、フォーマット情報ブロックで絵柄埋め込みが有りであるか否かを判定し、絵柄ありと判定された場合、ステップS211に進み、絵柄なしと判定された場合ステップS212に進む。

[0177] ステップS211では、絵柄埋込情報ブロックを参照することにより、絵柄埋め込みブロックの縦横それぞれのオフセット幅と高さ、および絵柄のブロックの幅と高さが判明する。

ステップS212では、実データブロックと、誤り訂正符号ブロックに関して、最寄りの、位置検出パターンの中心座標、または位置補正パターンの

中心座標、いずれか4点を選び、射影変換より、撮影画像内でのセルの座標を求める。

[0178] ステップS 2 1 3では、実データブロックと、誤り訂正符号ブロックについて、フォーマット情報ブロックのデータ種別に従い、複数の明（1）と暗（0）のデータを検出する。

ステップS 2 1 4では、実データブロックと、誤り訂正符号ブロックの個数から、誤り訂正符号ブロックの個数を計算する。これにより、実データと、誤り訂正符号が判明する。

[0179] ステップS 2 1 5では、誤り訂正符号を用いて、データ部と誤り訂正符号に関して、誤りの数と位置を検出する。誤りが0であれば、ステップS 2 1 9に進む。そうでなければ、ステップS 2 1 6に進む。

ステップS 2 1 6では、訂正可能であるか否かを判定し、訂正可能であればステップS 2 1 8に進み、訂正ができなければステップS 2 1 7に進む。

[0180] ステップS 2 1 7では、情報取り出しに失敗したのでメッセージが得られず、処理を終了する。

ステップS 2 1 8では、誤り訂正符号により、実データの誤り訂正を行う。

[0181] ステップS 2 1 9では、得られた実データは、ヘッダー（メッセージ種類（メッセージのエンコード）・メッセージサイズ）、メッセージ、終端フラグ、埋め草からなる。ヘッダーとメッセージを合わせたものをセグメントとする。終端フラグはメッセージを含まない、特殊なセグメントである。セグメントは複数の場合がある。終端フラグが見つかった際、その後は全て埋め草となる。データからメッセージを復元する。

ステップS 2 2 0では、情報取り出し処理を終了する。

[0182] 射影変換では、4組以上の座標が必要である。上記の解析処理では、4個の位置検出パターンが検出できた場合の例を説明したが、汚れやぼけ・ぶれなどのために、3個または2個の位置検出パターンのみが検出されることも起こり得る。本発明では、そのような場合、検出できた3個または2個の位

位置検出パターン形状が異なるので、3個または2個の位置関係から検出できなかった位置検出パターンの種類が判明することから、位置を容易に予測することができる。例えば、第1実施形態の二次元コードで、3個の位置検出パターンが検出できた場合には、形状から対角に位置する2個の位置検出パターンが識別できるので、検出できなかった1個は、対角の2個の位置検出パターンを結ぶ直線に対して、もう1個の位置検出パターンのミラー対称の位置にあると推測できる。また、さらに、2個の位置検出パターンが検出できた場合には、形状から2個の位置関係が判明する。例えば、隣接する2個の位置検出パターンであれば、2個の位置検出パターンを結ぶ直線から垂直な方向にこの直線と同じ長さ離れた位置に残りの2個の位置検出パターンが存在すると予測される。また、対角の2個の位置検出パターンであれば、2個の位置検出パターンの辺の延長線上の交点に残りの2個の位置検出パターンが存在すると予測される。例えば、第3位置検出パターンと第4位置検出パターンが見つかった場合に、それぞれの短辺2本を延長することで交点が4点できるが、この交点は第1位置検出パターンの頂点となり、同様に長辺を延長してできる交点4点は、第2位置検出パターンとなる。第1位置検出パターンと第2位置検出パターンが見つかった場合も同様の方法で予測できる。もちろん、4個の位置検出パターンの場合に比べて、3個の場合は精度が低下し、2個の場合はさらに精度が低下する。いずれにしても上記のようにして4個の位置検出パターンの位置を仮決定し、それを射影変換し、位置補正パターンの位置情報を取得する。そして、位置補正パターンを利用して座標を補正し、検出した位置検出パターンの座標と、位置補正パターンで補正した座標に基づいて4組の位置検出パターンの座標を最終決定する。

[0183] 位置検出パターンの個数は、汚れなどで検出できないものが発生することを考慮すると、個数が多い方が望ましいが、その分位置検出パターンの示す面積比率が増加してデータ効率が低下するだけでなく、位置検出パターンの検出処理量が増加するので、二次元コードの使用形態等に応じて適宜決定することが望ましい。

[0184] 次に、1画面に複数の二次元コードが写し込まれた場合について、本発明の二次元コードと図1の二次元コードを使用した場合を比較する。例えば、1画面に6個の図1の二次元コードが写し込まれた場合、同じ位置検出パターンが18個写し込まれる。前述のように、18個の位置検出パターンを検出し、それらのうち3個の組合せすべてについて同一の二次元コードのものであるかチェックするとする。この場合、組合せは ${}_{18}C_3 = 816$ 通りとなる。もし、1個の二次元コードに4個の同じ位置検出パターンがある場合には、画面の位置検出パターンの個数は24個になり、その組合せは、 ${}_{24}C_4 = 10624$ 通りとなる。このように、組合せの数が膨大となりその分処理時間が増加する。

[0185] これに対して、本発明のように位置検出パターンが異なる場合、3個の異なる位置検出パターンでは、 ${}_6C_1 \times {}_6C_1 \times {}_6C_1 = 216$ 通りとなり、4個の異なる位置検出パターンでは、 ${}_6C_1 \times {}_6C_1 \times {}_6C_1 \times {}_6C_1 = 1296$ 通りとなり、組合せが大幅に減る。

[0186] さらに、上記の解析処理では、画面を間引きながら走査して位置検出パターンを検出したが、1個の位置検出パターン候補を発見できれば、その形状から他の位置検出パターンが存在する方向および距離範囲をある程度予測することが可能であり、走査をその領域に限定することで位置検出パターンの検出効率を改善することができる。例えば、第1および第2実施形態の二次元コードでは、長方形の第3および第4位置検出パターンの長辺の延長方向に第2位置検出パターンが存在し、長辺に垂直な短辺の延長方向に第1位置検出パターンが存在すると予測することが可能である。2個の位置検出パターンが検出された場合および3個の位置検出パターンが検出された場合には、残りの位置検出パターンが存在すると予測される領域は更に限定される。

[0187] なお、上述した二次元コードは4個の異なる位置検出パターンが配置されているが、3個の異なる位置検出パターンと、1個以上の位置補正パターン、という構成でも精度の高いデータ抽出が可能である。この場合の位置検出パターンの配置は、4個の位置検出パターンの配置から、位置検出パターン

が1個欠けた配置となる。

[0188] 解析する際は、3個の位置検出パターンを検出した後、位置検出パターンが欠けた部分の位置検出パターンの座標を、他の位置検出パターンの位置関係から補完する。ただし、斜め上から撮影する場合は、補完された座標に誤差があることが懸念される。そこで、まずは位置検出パターン3組の座標と、補完した座標1組の、計4組の座標から、欠けた位置検出パターンの最寄りの位置補正パターンの座標を、射影変換により取得する。

[0189] そして、位置補正パターンを走査することにより位置補正パターンの座標補正を行う。次に、3個の位置検出パターンと、1個の位置補正パターンの、計4組の座標を用い、データの座標のマッピングを行う。位置補正パターンには、座標の誤差を軽減する効果があり、精度の高い座標の取得ができることから、位置検出パターンが3個と、1個以上の位置補正パターンからでも精度の高いデータの抽出が可能となる。

位置補正パターンが既に配置されている場合には、位置検出パターンが1つ減るため、コードの効率がよくなる。一方で、位置検出パターンの冗長性が減り、汚れなどに対する耐性が下がる。

[0190] また、入力される撮影画像は1枚とは限らず、複数画像の入力から1枚の画像を合成し、解析を行うことも考えられる。例えば、動画のような時系列の連続した画像から1枚の画像を生成して解析する場合や、コードの画像が分割され、複数枚に分けて入力される場合などである。

さらに、動画で異なるコードを連続して表示し、これらコードを全て認識することで、はじめて一つのデータが得られるといった活用も考えられる。この場合、動画でなく、紙に複数のコードを配置してもよい。

このような場合でも本発明の二次元コードを適用することは可能である。

[0191] 以上、本発明の実施形態を説明したが、記載した実施形態は発明を説明するためのもので、当業者には、特許請求の範囲において各種の変形例があり得ることが容易に理解可能である。

符号の説明

- [0192]
- 1 2 A 第 1 位置検出パターン
 - 1 2 B 第 3 位置検出パターン
 - 1 2 C 第 4 位置検出パターン
 - 1 2 D 第 2 位置検出パターン
 - 1 5 バージョン情報ブロック (縦)
 - 1 6 バージョン情報ブロック (横)
 - 1 7 フォーマット情報ブロック
 - 1 8 A – 1 8 B 絵柄埋込情報ブロック (縦)
 - 1 9 A – 1 9 B 絵柄埋込情報ブロック (横)

請求の範囲

- [請求項1] 二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置した二次元コードであって、
- 4個以上の異なる位置検出パターンを備えることを特徴とする二次元コード。
- [請求項2] 前記4個以上の異なる位置検出パターンの内、少なくとも3つは、内部に識別スペースを設けた枠形状であり、前記識別スペースは、二次元コードを構成する最小セルの2倍以上のスペースであることを特徴とする請求項1記載の二次元コード。
- [請求項3] 前記4個以上の異なる位置検出パターンは、正方形、長方形の組み合わせからなることを特徴とする請求項1または2記載の二次元コード。
- [請求項4] 前記4個以上の位置検出パターンは、
- 正方形の第1位置検出パターンと、
- 前記第1位置検出パターンより小さな正方形の第2位置検出パターンと、
- 長方形の第3位置検出パターンおよび第4位置検出パターンと、からなる4個の位置検出パターンであり、
- 前記第3位置検出パターンの長方形の長辺と前記第4位置検出パターンの長方形の長辺は、伸びる方向が異なり、
- 前記4個の位置検出パターンは、前記二次元のマトリックスの四隅に配置されている請求項1から3のいずれか1項記載の二次元コード。
- [請求項5] 前記第1位置検出パターンと前記第2位置検出パターンは、対角上に配置され、
- 前記第3位置検出パターンと前記第4位置検出パターンは、対角上に配置され、
- 前記第3位置検出パターンおよび前記第4位置検出パターンの長辺

は、前記第 1 位置検出パターンの辺と同じ長さであり、

前記第 3 位置検出パターンおよび前記第 4 位置検出パターンの短辺は、前記第 2 位置検出パターンの辺と同じ長さであり、

前記第 3 位置検出パターンおよび前記第 4 位置検出パターンの短辺の一方は、前記第 1 シンボルの辺の延長上に配置され、

前記第 3 位置検出パターンおよび前記第 4 位置検出パターンの長辺の一方は、前記第 2 位置検出パターンの辺の延長上に配置される請求項 4 記載の二次元コード。

[請求項6] 前記第 1 位置検出パターンおよび前記第 2 位置検出パターンは、それぞれ一重の正方形枠を有し、

前記第 3 位置検出パターンおよび前記第 4 位置検出パターンは、それぞれ一重の長方形枠を有する請求項 4 または 5 記載の二次元コード。

[請求項7] 前記第 1 位置検出パターンは、一重の正方形枠と、前記正方形枠内に配置された正方形を有し、

前記第 2 位置検出パターンは、一重の正方形枠を有し、

前記第 3 位置検出パターンおよび前記第 4 位置検出パターンは、それぞれ一重の長方形枠を有する請求項 4 または 5 記載の二次元コード。

[請求項8] 前記二次元のマトリックスの前記 4 個以上の位置検出パターン以外の領域は、複数の同じ大きさのブロックに分割され、

前記二次元のマトリックスは、隣接する前記ブロック間に、明または暗の分離スペースを有する請求項 1 から 7 のいずれか 1 項記載の二次元コード。

[請求項9] 前記 4 個以上の位置検出パターンは、前記ブロックより大きな面積を有する請求項 8 記載の二次元コード。

[請求項10] 前記ブロックは、 3×3 のセルを有し、
前記分離スペースは、1 セルの幅を有し、

前記 4 個以上の位置検出パターンは、4 セル以上の横幅と、4 セル以上の縦幅と、を有する請求項 9 記載の二次元コード。

[請求項11] バージョン情報ブロックおよびフォーマット情報ブロックが前記位置検出パターンの近傍に設けられていることを特徴とする請求項 1 から 10 のいずれか 1 項記載の二次元コード。

[請求項12] 二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置した二次元コードであって、

3 個の異なる位置検出パターンと、1 個以上の位置補正パターンを備えることを特徴とする二次元コード。

[請求項13] 二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置し、4 個以上の異なる位置検出パターンを備える二次元コードを撮影した画像から、前記二次元コードを解析する二次元コードの解析方法であって、

前記 4 個以上の異なる位置検出パターンのうち識別した前記位置検出パターンが 2 個または 3 個の時は、識別した 2 個または 3 個の前記位置検出パターンの位置関係から、残りの前記位置検出パターンの位置を推測し、推測した前記位置検出パターンの位置を含めて少なくとも 4 個の前記位置検出パターンの位置関係に基づいて前記二次元コードの座標を変換することを特徴とする二次元コードの解析方法。

[請求項14] 前記二次元コードは、位置補正パターンを含み、

前記 4 個以上の異なる位置検出パターンの位置関係の確認および識別した前記位置検出パターンが 2 個または 3 個の時の残りの前記位置検出パターンの位置の推測に、前記位置補正パターンを利用する請求項 13 記載の二次元コードの解析方法。

[請求項15] 二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置し、3 個の異なる位置検出パターンと 1 個以上の位置補正パターンを備える二次元コードを撮影した画像から、前記二次元コードを解析する二次元コードの解析方法であって、

前記 3 個の異なる位置検出パターンのうち識別した前記位置検出パターンが 2 個の時は、識別した 2 個の前記位置検出パターンの位置関係と位置補正パターンを利用し、残りの前記位置検出パターンの位置と、位置検出パターンが設けられていない第 4 の位置を推測し、推測した前記位置検出パターンの位置と第 4 の位置を含めた少なくとも 4 点の位置関係に基づいて前記二次元コードの座標を変換することを特徴とする二次元コードの解析方法。

[請求項16]

二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置し、4 個以上の異なる位置検出パターンを備える二次元コードを撮影した画像から、前記二次元コードを解析するようにコンピュータを制御するプログラムであって、

コンピュータが、

前記 4 個以上の異なる位置検出パターンのうち識別した前記位置検出パターンが 2 個または 3 個の時は、識別した 2 個または 3 個の前記位置検出パターンの位置関係から、残りの前記位置検出パターンの位置を推測し、推測した前記位置検出パターンの位置を含めて少なくとも 4 個の前記位置検出パターンの位置関係に基づいて二次元コードの座標を変換するように制御することを特徴とするプログラム。

[請求項17]

二進コードで表されるデータをセル化して、二次元のマトリックス状にパターンとして配置し、3 個の異なる位置検出パターンと 1 個以上の位置補正パターンを備える二次元コードを撮影した画像から、前記二次元コードを解析するようにコンピュータを制御するプログラムであって、

コンピュータが、

前記 3 個の異なる位置検出パターンのうち識別した前記位置検出パターンが 2 個の時は、識別した 2 個の前記位置検出パターンの位置関係と位置補正パターンから、残りの前記位置検出パターンの位置と位置検出パターンが設けられていない第 4 の位置を推測し、推測した前

記位置検出パターンと第4の位置を含めた少なくとも4個の位置関係に基づいて二次元コードの座標を変換するように制御することを特徴とするプログラム。

[請求項18]

請求項1記載の二次元コードの作成システムであって、

前記4個以上の異なる位置検出パターンを、二次元のマトリックスの所定位置に配置する位置検出パターン配置手段と、

前記二次元のマトリックスの前記4個以上の異なる位置検出パターンを配置する領域以外の領域に、前記二次元コードの解析に必要なベース情報を記録したベース情報領域を配置するベース情報配置手段と、

メッセージの記録データを、前記二次元のマトリックスの前記4個以上の異なる位置検出パターンおよび前記ベース情報領域を除いた領域に、順次配置するメッセージデータ配置手段と、を備えることを特徴とする二次元コードの作成システム。

[請求項19]

請求項1記載の二次元コードの作成システムであって、

前記3個の異なる位置検出パターンを、二次元のマトリックスの所定位置に配置する位置検出パターン配置手段と、

前記二次元のマトリックスの前記3個の異なる位置検出パターンを配置する領域以外の領域に、前記二次元コードの解析に必要なベース情報を記録したベース情報領域を配置するベース情報配置手段と、

メッセージの記録データを、前記二次元のマトリックスの前記3個の異なる位置検出パターンおよび前記ベース情報領域を除いた領域に、順次配置するメッセージデータ配置手段と、を備えることを特徴とする二次元コードの作成システム。