

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5125548号
(P5125548)

(45) 発行日 平成25年1月23日 (2013. 1. 23)

(24) 登録日 平成24年11月9日 (2012. 11. 9)

(51) Int. Cl.	F 1
G 0 6 F 3/042 (2006. 01)	G O 6 F 3/042 4 2 1
G 0 6 F 3/038 (2013. 01)	G O 6 F 3/038 3 1 O Y
G 0 6 T 1/00 (2006. 01)	G O 6 T 1/00 5 O O A
G 0 6 K 19/06 (2006. 01)	G O 6 K 19/00 E

請求項の数 8 (全 29 頁)

(21) 出願番号	特願2008-16819 (P2008-16819)	(73) 特許権者	000005496
(22) 出願日	平成20年1月28日 (2008. 1. 28)		富士ゼロックス株式会社
(65) 公開番号	特開2009-176250 (P2009-176250A)		東京都港区赤坂九丁目7番3号
(43) 公開日	平成21年8月6日 (2009. 8. 6)	(74) 代理人	100104880
審査請求日	平成22年12月17日 (2010. 12. 17)		弁理士 古部 次郎
		(74) 代理人	100118108
			弁理士 久保 洋之
		(72) 発明者	園田 隆志
			神奈川県足柄上郡中井町境430 グリー
			ンテクなかい 富士ゼロックス株式会社内
		(72) 発明者	大西 健司
			神奈川県足柄上郡中井町境430 グリー
			ンテクなかい 富士ゼロックス株式会社内
		審査官	田中 純一
			最終頁に続く

(54) 【発明の名称】 画像処理装置及びプログラム

(57) 【特許請求の範囲】

【請求項 1】

媒体上の位置を表す特定ビット列を構成する複数の部分列であって、当該媒体上の所定位置を表す m ($m \geq 1$) 個の部分列と当該 m 個の部分列に隣接する n ($n \geq 1$) 個の部分列とからなる複数の部分列にそれぞれ対応する複数のパターン画像を取得する画像取得手段と、

前記複数のパターン画像のうちの前記 m 個の部分列にそれぞれ対応する m 個のパターン画像から検出できた p ($1 \leq p \leq m$) 個の部分列と、前記複数のパターン画像のうちの前記 n 個の部分列にそれぞれ対応する n 個のパターン画像から検出できた q ($1 \leq q \leq n$) 個の部分列とを取得する部分列取得手段と、

前記 p 個の部分列により前記所定位置が特定されない場合に、前記特定ビット列を構成する複数の部分列と当該複数の部分列が表す前記媒体上の位置とを対応付けた対応情報を、前記画像取得手段により前回に取得された複数のパターン画像に基づいて特定された位置を検索開始位置として、当該 p 個の部分列と前記 q 個の部分列とを含む検索用部分列に基づいて検索することにより、当該所定位置を特定する特定手段とを備えたことを特徴とする画像処理装置。

【請求項 2】

前記特定ビット列は、 k 次の M 系列であり、

前記特定手段は、前記 p 個の部分列の長さ p と前記 q 個の部分列の長さ q との和が k 以上であれば、任意の位置を検索開始位置として、前記対応情報を前記検索用部分列に基づいて

検索し、前記 p 個の部分列の長さと同前記 q 個の部分列の長さの和が k 未満であれば、前記対応情報の前記検索用部分列に基づく検索を実行しないことを特徴とする請求項 1 記載の画像処理装置。

【請求項 3】

前記複数のパターン画像のうちの対応する部分列を取得できないパターン画像である誤りパターン画像を検出する検出手段を更に備え、

前記部分列取得手段は、前記複数のパターン画像のうち、前記誤りパターン画像以外の p 個のパターン画像から前記 p 個の部分列を取得し、前記誤りパターン画像以外の q 個のパターン画像から前記 q 個の部分列を取得することを特徴とする請求項 1 記載の画像処理装置。

10

【請求項 4】

前記特定手段は、前記対応情報において前記所定位置に対応付けられた複数の部分列に基づいて、前記複数のパターン画像から取得できなかった部分列を補完することにより、前記 m 個の部分列と同前記 n 個の部分列とを取得することを特徴とする請求項 3 記載の画像処理装置。

【請求項 5】

前記複数のパターン画像は、 p 個のパターン画像と同 q 個のパターン画像とからなり、

前記部分列取得手段は、前記複数のパターン画像のうち、前記 p 個のパターン画像から前記 p 個の部分列を取得し、前記 q 個のパターン画像から前記 q 個の部分列を取得することを特徴とする請求項 1 記載の画像処理装置。

20

【請求項 6】

前記特定手段は、前記対応情報において前記所定位置に対応付けられた複数の部分列に基づいて、前記複数のパターン画像から誤って取得した部分列を訂正することにより、前記 m 個の部分列と同前記 n 個の部分列とを取得することを特徴とする請求項 5 記載の画像処理装置。

【請求項 7】

コンピュータに、

媒体上の位置を表す特定ビット列を構成する複数の部分列であって、当該媒体上の所定位置を表す m ($m \geq 1$) 個の部分列と同当該 m 個の部分列に隣接する n ($n \geq 1$) 個の部分列とからなる複数の部分列にそれぞれ対応する複数のパターン画像を取得する機能と、

30

前記複数のパターン画像のうちの前記 m 個の部分列にそれぞれ対応する m 個のパターン画像から検出できた p ($1 \leq p \leq m$) 個の部分列と、前記複数のパターン画像のうちの前記 n 個の部分列にそれぞれ対応する n 個のパターン画像から検出できた q ($1 \leq q \leq n$) 個の部分列とを取得する機能と、

前記 p 個の部分列により前記所定位置が特定されない場合に、前記特定ビット列を構成する複数の部分列と同当該複数の部分列が表す前記媒体上の位置とを対応付けた対応情報を、前記複数のパターン画像を取得する機能により前回に取得された複数のパターン画像に基づいて特定された位置を検索開始位置として、当該 p 個の部分列と同前記 q 個の部分列を含む検索用部分列に基づいて検索することにより、当該所定位置を特定する機能とを実現させるためのプログラム。

40

【請求項 8】

前記特定ビット列は、 k 次の M 系列であり、

前記特定する機能では、前記 p 個の部分列の長さと同前記 q 個の部分列の長さとの和が k 以上であれば、任意の位置を検索開始位置として、前記対応情報を前記検索用部分列に基づいて検索し、前記 p 個の部分列の長さと同前記 q 個の部分列の長さの和が k 未満であれば、前記対応情報の前記検索用部分列に基づく検索を実行しないことを特徴とする請求項 7 記載のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

50

本発明は、画像処理装置、プログラムに関する。

【背景技術】

【0002】

表面上の複数の位置を符号化する位置コードを提供する方法は知られている（例えば、特許文献1参照）。この特許文献1では、循環数字列を表面に沿って複数回印刷する。その際、循環数字列の異なるローテーションを、隣り合う数字列間で所定のずれが起こるように使用し、表面を分割した複数のコードウィンドウが、少なくとも3つの循環数字列を備えると共に、隣り合うコードウィンドウの1つの数字列と重なり合う1つの数字列を有するものであり、また、コードウィンドウの位置を、そのコードウィンドウに属している循環数字列間のずれによって符号化している。

10

【0003】

論理的に順位付けられたデジタル量子の機械読取り可能なエンコーディングを記録した記録媒体からのエンコーディングの空間的に非同期的な読取りを可能にする技術も知られている（例えば、特許文献2参照）。この特許文献2では、本質的に同一であるエンコーディングの多数のコピーを形成し、機械認識可能な空間同期指標をエンコーディングのコピーの各々の中のプリント位置に組み合わせて、エンコーディングの空間的に同期可能な多数の事例を提供し、それらの事例をレイアウト規則に従って記録媒体における空間的に周期的な中心の格子に書き込んでいる。

【0004】

取り込まれた画像の位置をより大きい画像から決定するためのシステム及び方法も知られている（例えば、特許文献3参照）。この特許文献3では、非反復的な系列を、所定のサイズのサブウィンドウ毎に固有とされた非反復的な配列に折り畳み、その取り込まれた位置を非反復的な配列内のサブウィンドウ毎に決定することによって、取り込まれた画像の位置を、より大きいサブウィンドウの画像から決定している。

20

【0005】

光学的に読み取り可能に記録されるマルチメディア情報の長時間記録及び繰り返し再生を可能とする技術も知られている（例えば、特許文献4参照）。この特許文献4において、記録装置は、プリンタシステムや印刷用製版システムにより、オーディオ情報、映像情報、デジタルコードデータ等を含めた所謂マルチメディア情報を光学的に読み取り可能なドットコードとして、画像や文字と一緒に紙等の媒体上に記録する。ペン型の情報再生装置は、ドットコードの手動走査に応じてそのドットコードを順次取り込んで、元のオーディオ情報を音声出力器にて、元の映像情報を表示装置にて、元のデジタルコードデータをページプリンタ等にて出力する。

30

【0006】

タブレットを使用せずに、媒体上の座標を精緻に検出できるようにする技術も知られている（例えば、特許文献5参照）。この特許文献5において、ペン型の座標入力装置は、媒体上に形成されその媒体上の座標を示すコードシンボルを光学的に読み取り、この読み取ったコードシンボルをデコードし、読み取った画像中におけるコードシンボルの位置、向き及び歪み量を検出する。そして、デコード後の情報と、コードシンボルの位置、向き及び歪み量とにより、先端部の媒体上での位置の座標を検出する。

40

【0007】

【特許文献1】特表2003-511762号公報

【特許文献2】特開平9-185669号公報

【特許文献3】特開2004-152273号公報

【特許文献4】特開平6-231466号公報

【特許文献5】特開2000-293303号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

ここで、一般に、特定ビット列を構成する部分列と媒体上の位置とを対応付けた対応情

50

報をパターン画像から取得した部分列に基づいて検索することにより、媒体上の位置を特定する場合、パターン画像から部分列を取得できないことや、パターン画像から誤った部分列を取得すること等があるので、媒体上の位置を特定できる可能性を高めるのには限界があった。

本発明の目的は、パターン画像から取得した部分列により媒体上の位置を特定できる可能性を高めることにある。

【課題を解決するための手段】

【0009】

請求項1に記載の発明は、媒体上の位置を表す特定ビット列を構成する複数の部分列であって、当該媒体上の所定位置を表す m ($m \geq 1$) 個の部分列と当該 m 個の部分列に隣接する n ($n \geq 1$) 個の部分列とからなる複数の部分列にそれぞれ対応する複数のパターン画像を取得する画像取得手段と、前記複数のパターン画像のうちの前記 m 個の部分列にそれぞれ対応する m 個のパターン画像から検出できた p ($1 \leq p \leq m$) 個の部分列と、前記複数のパターン画像のうちの前記 n 個の部分列にそれぞれ対応する n 個のパターン画像から検出できた q ($1 \leq q \leq n$) 個の部分列とを取得する部分列取得手段と、前記 p 個の部分列により前記所定位置が特定されない場合に、前記特定ビット列を構成する複数の部分列と当該複数の部分列が表す前記媒体上の位置とを対応付けた対応情報を、前記画像取得手段により前回に取得された複数のパターン画像に基づいて特定された位置を検索開始位置として、当該 p 個の部分列と前記 q 個の部分列とを含む検索用部分列に基づいて検索することにより、当該所定位置を特定する特定手段とを備えたことを特徴とする画像処理装置である。

請求項2に記載の発明は、前記特定ビット列は、 k 次の M 系列であり、前記特定手段は、前記 p 個の部分列の長さ $と前記 q 個の部分列の長さとの和が k 以上であれば、任意の位置を検索開始位置として、前記対応情報を前記検索用部分列に基づいて検索し、前記 p 個の部分列の長さ $と前記 q 個の部分列の長さとの和が k 未満であれば、前記対応情報の前記検索用部分列に基づく検索を実行しないことを特徴とする請求項1記載の画像処理装置である。$$

請求項3に記載の発明は、前記複数のパターン画像のうちの対応する部分列を取得できないパターン画像である誤りパターン画像を検出する検出手段を更に備え、前記部分列取得手段は、前記複数のパターン画像のうち、前記誤りパターン画像以外の p 個のパターン画像から前記 p 個の部分列を取得し、前記誤りパターン画像以外の q 個のパターン画像から前記 q 個の部分列を取得することを特徴とする請求項1記載の画像処理装置である。

請求項4に記載の発明は、前記特定手段は、前記対応情報において前記所定位置に対応付けられた複数の部分列に基づいて、前記複数のパターン画像から取得できなかった部分列を補完することにより、前記 m 個の部分列と前記 n 個の部分列とを取得することを特徴とする請求項3記載の画像処理装置である。

請求項5に記載の発明は、前記複数のパターン画像は、 p 個のパターン画像と q 個のパターン画像とからなり、前記部分列取得手段は、前記複数のパターン画像のうち、前記 p 個のパターン画像から前記 p 個の部分列を取得し、前記 q 個のパターン画像から前記 q 個の部分列を取得することを特徴とする請求項1記載の画像処理装置である。

請求項6に記載の発明は、前記特定手段は、前記対応情報において前記所定位置に対応付けられた複数の部分列に基づいて、前記複数のパターン画像から誤って取得した部分列を訂正することにより、前記 m 個の部分列と前記 n 個の部分列とを取得することを特徴とする請求項5記載の画像処理装置である。

請求項7に記載の発明は、コンピュータに、媒体上の位置を表す特定ビット列を構成する複数の部分列であって、当該媒体上の所定位置を表す m ($m \geq 1$) 個の部分列と当該 m 個の部分列に隣接する n ($n \geq 1$) 個の部分列とからなる複数の部分列にそれぞれ対応する複数のパターン画像を取得する機能と、前記複数のパターン画像のうちの前記 m 個の部分列にそれぞれ対応する m 個のパターン画像から検出できた p ($1 \leq p \leq m$) 個の部分列と、前記複数のパターン画像のうちの前記 n 個の部分列にそれぞれ対応する n 個のパター

ン画像から検出できた q ($1 \leq q \leq n$) 個の部分列とを取得する機能と、前記 p 個の部分列により前記所定位置が特定されない場合に、前記特定ビット列を構成する複数の部分列と当該複数の部分列が表す前記媒体上の位置とを対応付けた対応情報を、前記複数のパターン画像を取得する機能により前回に取得された複数のパターン画像に基づいて特定された位置を検索開始位置として、当該 p 個の部分列と前記 q 個の部分列とを含む検索用部分列に基づいて検索することにより、当該所定位置を特定する機能とを実現させるためのプログラムである。

請求項 8 に記載の発明は、前記特定ビット列は、 k 次の M 系列であり、前記特定する機能では、前記 p 個の部分列の長さとの和が k 以上であれば、任意の位置を検索開始位置として、前記対応情報を前記検索用部分列に基づいて検索し、前記 p 個の部分列の長さとの和が k 未満であれば、前記対応情報の前記検索用部分列に基づく検索を実行しないことを特徴とする請求項 7 記載のプログラムである。

【発明の効果】

【0010】

請求項 1 の発明は、本構成を有していない場合に比較して、パターン画像から取得した部分列により媒体上の位置を特定する処理の効率化が図れるという効果を有する。

請求項 2 の発明は、本構成を有していない場合に比較して、パターン画像から取得した部分列により特定される媒体上の位置の精度が高まるという効果を有する。

請求項 3 の発明は、一部のパターン画像から部分列を取得できない場合であっても、パターン画像から取得した部分列により媒体上の位置を特定できるという効果を有する。

請求項 4 の発明は、一部のパターン画像から部分列を取得できない場合であっても、正しい部分列を取得することができるという効果を有する。

請求項 5 の発明は、一部のパターン画像から誤った部分列を取得した場合であっても、パターン画像から取得した部分列により媒体上の位置を特定できるという効果を有する。

請求項 6 の発明は、一部のパターン画像から部分列を取得できない場合であっても、正しい部分列を取得することができるという効果を有する。

請求項 7 の発明は、本構成を有していない場合に比較して、パターン画像から取得した部分列により媒体上の位置を特定する処理の効率化が図れるという効果を有する。

請求項 8 の発明は、本構成を有していない場合に比較して、パターン画像から取得した部分列により特定される媒体上の位置の精度が高まるという効果を有する。

【発明を実施するための最良の形態】

【0011】

以下、添付図面を参照して、本発明を実施するための最良の形態（以下、「実施の形態」という）について詳細に説明する。

まず、本実施の形態で利用する符号化方式について説明する。

本実施の形態における符号化方式では、 m ($m \geq 3$) 箇所から選択した n ($1 \leq n < m$) 箇所に単位画像を配置してなるパターン画像（以下、「符号パターン」という）によって mCn ($= m! / \{(m-n)! \times n!\}$) 通りの情報を表現する。つまり、1つの単位画像を情報に対応させるのではなく、複数の単位画像を情報に対応させている。仮に1つの単位画像を情報に対応させた다고すると、単位画像が欠損したり、ノイズが加わったりした場合に、誤った情報を表現してしまうという欠点がある。これに対して、例えば2つの単位画像を情報に対応させた다고すると、単位画像が1つであったり3つであったりした場合に、容易に誤りであることが分かる。更に、1つの単位画像で1ビット、又は、高々2ビットを表現する方法では、情報を表現するための情報パターンと視覚的に似たパターンで、情報パターンの読出しを制御する同期パターンを表現することができない。このようなことから、本実施の形態では、上記のような符号化方式を採用している。以下では、このような符号化方式を mCn 方式と称する。

ここで、単位画像としては、如何なる形状のものを用いてもよい。本実施の形態では、単位画像の一例としてドット画像（以下、単に「ドット」という）を用いるが、例えば、

斜線パターン等、他の形状の画像であってもよい。

【 0 0 1 2 】

図 1 に、 mCn 方式における符号パターンの例を示す。

図では、黒色の領域と斜線の領域をドット配置可能な領域とし、その間にある白色の領域をドット配置不可能な領域としている。そして、ドット配置可能な領域のうち、黒色の領域にドットが配置され、斜線の領域にはドットが配置されていないことを示している。即ち、図 1 は、縦 3 × 横 3 の合計 9 個のドット配置可能な領域を設けた例を示しており、このうち、(a) は、9 個のドット配置可能な領域に 2 ドットを配置する $9C2$ 方式における符号パターンの一例を、(b) は、9 個のドット配置可能な領域に 3 ドットを配置する $9C3$ 方式における符号パターンの一例を示している。

10

【 0 0 1 3 】

但し、図 1 で配置されるドット（黒色の領域）は、あくまで情報表現のためのドットであり、画像を構成する最小単位であるドット（図 1 の最小の四角）とは一致していない。本実施の形態において、「ドット」というときは前者のドットを指し、後者のドットは「ピクセル」と呼ぶことにすれば、ドットは、600 dpi における 2 ピクセル × 2 ピクセルの大きさを有することになる。600 dpi における 1 ピクセルの一辺の長さは 0.0423 mm なので、1 ドットの一辺の長さは、 $84.6 \mu m (= 0.0423 mm \times 2)$ となる。ドットは、大きくなればなるほど目に付きやすくなるため、できるだけ小さいほうが好ましい。ところが、あまり小さくすると、プリンタで印刷できなくなってしまう。そこで、ドットの大きさとして、 $50 \mu m$ より大きく $100 \mu m$ より小さい上記の値を採用している。但し、上記の値 $84.6 \mu m$ は、あくまで計算上の数値であり、実際に印刷されたトナー像では $100 \mu m$ 程度になる。

20

【 0 0 1 4 】

次いで、図 2 に、 $9C2$ 方式における全符号パターンの例を示す。尚、ここでは、ドット間の空白は省略している。図示するように、 $9C2$ 方式では、36 ($= {}_9C_2$) 通りの符号パターンが利用される。また、全符号パターンに対して、各符号パターンを一意に識別するための番号であるパターン値が付される。図では、各符号パターンに対するこのパターン値の割り当ての例も示している。但し、図に示した対応はあくまで一例であり、どの符号パターンにどのパターン値を割り当ててもよい。

30

【 0 0 1 5 】

また、図 3 に、 $9C3$ 方式における全符号パターンの例を示す。尚、ここでも、ドット間の空白は省略している。図示するように、 $9C3$ 方式では、84 ($= {}_9C_3$) 通りの符号パターンが利用される。また、この場合も、全符号パターンに対して、各符号パターンを一意に識別するための番号であるパターン値が付される。図では、各符号パターンに対するこのパターン値の割り当ての例も示している。但し、この場合も、図に示した対応はあくまで一例であり、どの符号パターンにどのパターン値を割り当ててもよい。

【 0 0 1 6 】

尚、ここでは、符号パターンが配置される領域（以下、「パターンブロック」という）の大きさを、3 ドット × 3 ドットを配置可能な大きさとした。しかしながら、パターンブロックの大きさはこれに限るものではない。つまり、2 ドット × 2 ドット、4 ドット × 4 ドット等を配置可能な大きさであってもよい。

40

また、パターンブロックの形状として、正方形でなく、例えば 3 ドット × 4 ドットを配置する場合のように長方形を採用してもよい。尚、本明細書において、長方形とは、隣り合う 2 辺の長さが等しくない矩形、つまり、正方形以外の矩形のことをいうものとする。

更に、任意に決めた数のドット配置可能な領域のうち、幾つの領域にドットを配置するかも、表現したい情報の量と許容できる画像濃度とを考慮して、適宜決めるとよい。

【 0 0 1 7 】

このように、本実施の形態では、 m 箇所から n 箇所を選択することで mCn 種類の符号パターンを用意している。そして、これらの符号パターンのうち、特定のパターンを情報パターンとして利用し、残りを同期パターンとして利用する。ここで、情報パターンとは

50

、媒体に埋め込む情報を表現するパターンである。また、同期パターンとは、媒体に埋め込まれた情報パターンを取り出すために用いられるパターンである。例えば、情報パターンの位置を特定したり、画像の回転を検出したりするために用いられる。尚、媒体としては、画像を印刷することが可能であれば、如何なるものを用いてもよい。紙が代表例なので、以下では媒体を紙として説明するが、金属、プラスチック、繊維等であってもよい。

【0018】

ここで、図2又は図3に示した符号パターンのうちの同期パターンについて説明する。尚、これらの符号パターンを利用する場合、パターンブロックの形状は正方形となるため、画像の回転を90度単位で認識する必要がある。従って、4種類の符号パターンで1組の同期パターンが構成される。

図4(a)は、9C2方式における同期パターンの例である。ここでは、36種類の符号パターンのうち、32種類の符号パターンを5ビットの情報を表現する情報パターンとし、残りの4種類の符号パターンで1組の同期パターンを構成するものとする。例えば、パターン値「32」の符号パターンを正立した同期パターン、パターン値「33」の符号パターンを右に90度回転した同期パターン、パターン値「34」の符号パターンを右に180度回転した同期パターン、パターン値「35」の符号パターンを右に270度回転した同期パターンとしている。但し、36種類の符号パターンの情報パターンと同期パターンへの振り分け方は、これには限らない。例えば、16種類の符号パターンを4ビットの情報を表現する情報パターンとし、残りの20種類の符号パターンで5組の同期パターンを構成してもよい。

【0019】

また、図4(b)は、9C3方式における同期パターンの例である。ここでは、84種類の符号パターンのうち、64種類の符号パターンを6ビットの情報を表現する情報パターンとし、残りの20種類の符号パターンで5組の同期パターンを構成するものとする。図は、この5組の同期パターンのうちの2組を示す。例えば、第1組においては、パターン値「64」の符号パターンを正立した同期パターン、パターン値「65」の符号パターンを右に90度回転した同期パターン、パターン値「66」の符号パターンを右に180度回転した同期パターン、パターン値「67」の符号パターンを右に270度回転した同期パターンとしている。また、第2組においては、パターン値「68」の符号パターンを正立した同期パターン、パターン値「69」の符号パターンを右に90度回転した同期パターン、パターン値「70」の符号パターンを右に180度回転した同期パターン、パターン値「71」の符号パターンを右に270度回転した同期パターンとしている。

【0020】

尚、図示しないが、パターンブロックの形状が長方形である場合は、画像の回転の検出のために2種類の符号パターンを同期パターンとして用意すればよい。例えば、縦3ドット×横4ドットを配置可能な領域が検出されるべきなのに、縦4ドット×横3ドットを配置可能な領域が検出された場合は、その時点で画像が90度又は270度回転していることが分かるからである。

【0021】

次に、同期パターンと情報パターンを配置してなる情報表現の最小単位(以下、「符号ブロック」という)について説明する。

図5は、符号ブロックのレイアウトの一例を示したものである。

図中、(a)及び(b)のそれぞれの右側に、符号ブロックのレイアウトを示している。ここでは、レイアウトとして、パターンブロックを5個×5個の25個並べたものを採用している。この25個のパターンブロックのうち、左上の1ブロックに同期パターンを配置している。また、同期パターンの右の4ブロックに紙面上の横方向の座標を特定するX座標情報を表す情報パターンを配置し、同期パターンの下の4ブロックに紙面上の縦方向の座標を特定するY座標情報を表す情報パターンを配置している。更に、これらの座標情報を表す情報パターンで囲まれた16ブロックに、紙面又は紙面に印刷される文書の識別情報を表す情報パターンを配置している。

【 0 0 2 2 】

また、(a) の左側には、9 C 2 方式における符号パターンが各パターンブロックに配置されていることを示している。即ち、3 6 種類の符号パターンを、例えば、4 種類の同期パターンと3 2 種類の情報パターンとに分け、各パターンをレイアウトに従って配置している。一方、(b) の左側には、9 C 3 方式における符号パターンが各パターンブロックに配置されていることを示している。即ち、8 4 種類の符号パターンを、例えば、2 0 種類の同期パターンと6 4 種類の情報パターンとに分け、各パターンをレイアウトに従って配置している。

【 0 0 2 3 】

尚、本実施の形態において、座標情報は、紙面の縦方向及び横方向にM系列で表現される。ここで、M系列とは、その部分列が、他の部分列と一致することがない系列である。例えば、1 1 次のM系列は、2 0 4 7 ビットのビット列である。そして、この2 0 4 7 ビットのビット列の中から取り出された1 1 ビット以上の部分列と同じ並びは、この2 0 4 7 ビットのビット列の中に自身以外には存在しない。本実施の形態では、1 個の符号パターンを4 ビットに対応付ける。即ち、2 0 4 7 ビットのビット列を4 ビットごとに1 0 進数で表し、図 2 又は図 3 の割り当てに従って符号パターンを決定し、紙面の横と縦に渡って印刷する。従って、復号の際は、3 つの連続する符号パターンを特定し、符号パターンと座標との対応関係を格納したテーブルを参照することにより、ビット列上の位置が特定されることになる。

【 0 0 2 4 】

図 6 に、M系列を用いた座標情報の符号化の例を示す。

(a) は、1 1 次のM系列の例として、「0 1 1 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 0 ...」というビット列を示している。本実施の形態では、これを4 ビットずつに区切り、1 つ目の部分列「0 1 1 1」をパターン値「7」の符号パターンとして、2 つ目の部分列「0 0 0 1」をパターン値「1」の符号パターンとして、3 つ目の部分列「0 1 0 1」をパターン値「5」の符号パターンとして、4 つ目の部分列「1 0 1 0」をパターン値「1 0」の符号パターンとして、それぞれ紙面上に配置する。

【 0 0 2 5 】

また、このように4 ビットずつ区切って符号パターンに割り当てると、(b) に示すように、4 周期で全てのパターン列が表現される。即ち、1 1 次のM系列は、2 0 4 7 ビットなので、この系列を4 ビットごとに切り出し、符号パターンで表現していくと、最後に3 ビットあまることになる。この3 ビットに、M系列の最初の1 ビットを加えて4 ビットとし符号パターンで表す。更に、M系列の2 ビット目から4 ビットごとに切り出し符号パターンで表す。すると、次の周期は、M系列の3 ビット目から始まり、その次の周期は、M系列の4 ビット目から始まる。更に、5 周期目は、5 ビット目からとなるが、これは最初の周期と一致する。従って、M系列の4 周期を4 ビットごとに切り出していくと、2 0 4 7 個の符号パターンで全てを尽くすことができる。M系列は1 1 次であるので、3 つの連続する符号パターンは、他のどの位置の連続する符号パターンとも一致することはない。そこで、読出し時には3 つの符号パターンを読み出せば、復号が可能である。但し、本実施の形態では、誤りの発生を考慮し、4 つの符号パターンで座標情報を表現している。

【 0 0 2 6 】

また、識別情報の符号化には、幾つかの方法の利用が考えられるが、本実施の形態では、R S 符号化が適している。R S 符号は多値の符号法であり、各パターンブロックに配置された符号パターンのパターン値をR S 符号の多値に対応させるとよいからである。

尚、本実施の形態における符号パターンの利用形態としては、例えば、文書画像に重ねてその識別情報を紙面に印刷し、ペン状のスキヤナで紙面上の部分画像を読み込み、そこから文書画像の識別情報を取得する、といったものが想定される。この場合、紙面上の汚れやスキヤナの性能によって誤りが発生するが、この誤りはR S 符号により訂正される。

【 0 0 2 7 】

ここで、R S 符号による訂正及びそのような訂正を行う場合に表現可能な情報量につい

10

20

30

40

50

て具体的に説明する。

本実施の形態では、上述したように、1パターンブロック内のドット数が一定である符号パターンを採用している。これにより、仮に1ドットが消失した場合や、1ドットが付加された場合は、パターンブロック内のドット数が変わる。従って、これらは誤りだと分かる誤りとなる。一方で、ドットの消失と付加が同時に起こった場合は、他の符号パターンであると誤認識してしまうので、誤りだと分からない誤りとなる。

例えば、識別情報を表す情報パターンを配置する16ブロックのうち、10ブロックを識別情報そのものを表す情報パターンを配置するブロックとし、6ブロックを訂正のためのブロックとする。この場合、誤りだと分かるブロックは6ブロックまで、誤りだと分からないブロックも3ブロックまで、訂正される。これを例えば9C2方式における32種類の情報パターンで実現したとすると、1ブロックで5ビットの情報が表現されるので、識別情報そのものは10ブロックで50ビット分表現される。また例えば9C3方式における64種類の情報パターンで実現したとすると、1ブロックで6ビットの情報が表現されるので、識別情報そのものは10ブロックで60ビット分表現される。

【0028】

次いで、上記符号ブロックを含む広範囲のレイアウトについて説明する。

図7は、そのようなレイアウトの一例を示した図である。このレイアウトでは、図5の符号ブロックが基本単位として紙面全体の縦方向及び横方向に周期的に配置されている。

ここで、同期パターンとしては、同じ符号パターンが各符号ブロックにおける左上のパターンブロックに配置されている。図では、同期パターンを「S」で表している。

また、X座標情報としては、符号パターンの同じ並びが、同期パターンが配置されたのと同じ行の各パターンブロックに配置されている。Y座標情報としては、符号パターンの同じ並びが、同期パターンが配置されたのと同じ列の各パターンブロックに配置されている。図では、X座標情報を表すパターンを「X01」、「X02」、...で表し、Y座標情報を表すパターンを「Y01」、「Y02」、...で表している。

更に、識別情報としては、符号パターンの同じ配列が、縦方向及び横方向に周期的に配置されている。図では、識別情報を表すパターンを「I01」、「I02」、...、「I16」で表している。

そして、このようなレイアウトを採用することにより、例えば、図中、丸印で示した範囲が読み取られた場合のように、図5の符号ブロックの全体を含む範囲が読み取られていない場合であっても、後述する処理により、識別情報及び座標情報が得られる。

【0029】

尚、このようなレイアウトで紙面に印刷される符号画像は、例えば電子写真方式を用いて、Kトナー（カーボンを含む赤外光吸収トナー）、又は、特殊トナーにより形成される。

ここで、特殊トナーとしては、可視光領域（400nm～700nm）における最大吸収率が7%以下であり、近赤外領域（800nm～1000nm）における吸収率が30%以上の不可視トナーが例示される。ここで、「可視」及び「不可視」は、目視により認識できるかどうかとは関係しない。印刷された媒体に形成された画像が可視光領域における特定の波長の吸収に起因する発色性の有無により認識できるかどうかで「可視」と「不可視」とを区別している。また、可視光領域における特定の波長の吸収に起因する発色性が若干あるが、人間の目で認識し難いものも「不可視」に含める。

【0030】

ところで、上述したように、M系列を利用して座標を表現する場合には、M系列の部分列と座標値との対応を座標テーブルとして保持する必要がある。そして、連続する3つの符号パターンから得られた11ビット値に基づいて座標テーブルを検索することにより、座標値が求められる。しかしながら、本実施の形態における符号パターンには、ドットの消失又は付加によりブロック内のドット数が変化してパターン値を特定できなくなる誤り（以下、「特定不能誤り」という）と、ドットの消失及び付加が起こったもののブロック内のドット数が変化しなかったために誤りだと分からず間違ったパターン値を特定してし

もう誤り（以下、「誤特定誤り」という）とが生じる可能性がある。こうした誤りの発生に備え、誤ったパターンも含む部分列を座標テーブルに保持しておく方法も考えられるが、この方法は記憶容量の点で現実的ではない。

【 0 0 3 1 】

そこで、本実施の形態では、最初は誤りが無いと仮定し、連続する3つの符号パターンから得られた11ビット値に基づいて座標テーブルを検索し、検索に失敗した場合に、連続する4つの符号パターンを用いてM系列テーブルを検索して座標値を求める。ここで、M系列テーブルとは、座標とその座標に配置されるブロックのパターン値とを対応付けたものである。即ち、連続する4つの符号パターンの一部に特定不能誤り又は誤特定誤りがあり、座標テーブルから座標値が得られなかった場合であっても、残りの符号パターンに合致する場所を特定することにより、座標値を求めるものである。

10

【 0 0 3 2 】

また、このときのM系列テーブルの検索方法としては、先頭から順に走査して適合する部分列を検索するという方法もあるが、この方法は効率的ではない。

そこで、本実施の形態では、ストローク内の各フレームで取得される座標に大きな飛躍が無いことを利用する。即ち、例えば1フレーム前の履歴情報があれば、その履歴情報で示される座標を検索開始位置として、M系列の探索を行う。

【 0 0 3 3 】

図8は、本実施の形態におけるM系列の探索方法の第1の例を示した図である。

このうち、(a)は、読み取った画像から得られた獲得パターン列に基づいて、登録パターン列を探索する際の様子を示した図である。ここでは、獲得パターン列の4ブロックのうちの3番目のブロックに特定不能誤りが発生したものとしている。そして、登録パターン列を先頭から探索していき、座標値「7」、「8」、「9」、「10」に対応する登録パターン列「6」、「2」、「0」、「10」が、3番目のブロックを除き、獲得パターン列と一致しているので、座標値「7」を取得している。

20

【 0 0 3 4 】

ところで、k次のM系列を用いて座標を表現した場合、探索に用いるビット列の長さがkビット以上であれば、これらのパターン値に合致する位置が2つ以上見つかることはない。

(b)に、このことを示す。ここでは、Pが、2047の座標位置のうちの求めるべき座標位置である。また、Qは、前フレームで求められた座標位置である。11次のM系列を用いた場合において、連続する4つの符号パターンのうちの3つの符号パターンのパターン値が一致する並びは、P以外には存在しない。従って、(a)のようにM系列の先頭から探索しても、間違った座標値を得ることはない。但し、この方法は効率的ではないので、Qを中心としてその近くから左右に探索していくのがよい。

30

【 0 0 3 5 】

図9は、本実施の形態におけるM系列の探索方法の第2の例を示した図である。

このうち、(a)は、読み取った画像から得られた獲得パターン列に基づいて、登録パターン列を探索する際の様子を示した図である。ここでは、獲得パターン列の4ブロックのうちの1番目のブロックと3番目のブロックに特定不能誤りが発生したものとしている。そして、登録パターン列を座標値「9」の位置から探索していき、座標値「17」、「18」、「19」、「20」に対応する登録パターン列「8」、「8」、「1」、「2」が、1番目のブロックと3番目のブロックを除き、獲得パターン列と一致しているので、座標値「17」を取得している。

40

【 0 0 3 6 】

ところで、k次のM系列を用いて座標を表現した場合、探索に用いるビット列の長さがkビット未満であれば、これらのパターン値に合致する位置が2つ以上見つかってしまう。

(b)に、このことを示す。ここでは、Pが、2047の座標位置のうちの求めるべき座標位置である。また、Qは、前フレームで求められた座標位置である。11次のM系列

50

を用いた場合において、連続する4つの符号パターンのうちの2つの符号パターンのパターン値が一致する並びは、7箇所存在する。図では、P以外の6か所を、R1、R2、...、R6としている。P及びR1、R2、...、R6の位置はランダムであるが、平均間隔186mmで必ず7つ存在する。従って、M系列の先頭から探索すると間違った座標値を得る可能性があるので、Qを中心としてその近くから左右に探索していくのがよい。(a)では、前フレームで得た座標(座標値「9」)から検索を行っている。

【0037】

尚、図8及び図9に示した座標値と登録パターン列との対応が、上述したM系列テーブルに対応する。

また、座標テーブルについては、図示しないが、次のようなものが想定される。即ち、M系列テーブルにおける座標値jに対応する登録パターン列をPATTERN_SEQUENCE[j]と表記すると、座標値jと、 $PATTERN_SEQUENCE[j] + 16 \times PATTERN_SEQUENCE[j+1] + (16 \times 16 \times PATTERN_SEQUENCE[j+2] >> 1)$ により得られた10進数とを対応付けたものである。尚、「>>」は、ビット列を右にシフトすることを示す。また、獲得パターン列に基づく検索が効率的に行えるよう、座標テーブル内のデータは、10進数の昇順にソートされていることが望ましい。

そして、本実施の形態において、M系列テーブル及び座標テーブルは、X座標符号復号部42及びY座標符号復号部47が参照可能な図示しないメモリに記憶されているものとする。

【0038】

以下、このような座標の復号について、より詳細に説明する。尚、本実施の形態では、如何なるmcn方式を利用してもよいが、以降の説明においては、簡単のため、9c3方式の利用を前提とする。また、以下では、パターンブロックのことを単に「ブロック」ともいう。

【0039】

まず、紙面に形成された符号画像を読み取って処理する画像処理装置20について説明する。

図10は、画像処理装置20の構成例を示したブロック図である。

図示するように、画像処理装置20は、画像読取部21と、ドット配列生成部22と、ブロック検出部23と、同期符号検出部24とを備える。また、識別符号検出部30と、識別符号復号部32と、X座標符号検出部40と、X座標符号復号部42と、Y座標符号検出部45と、Y座標符号復号部47と、情報出力部50とを備える。

【0040】

画像読取部21は、CCD(Charge Coupled Devices)やCMOS(Complementary Metal Oxide Semiconductor)等の撮像素子を用いて、紙面に印刷された符号画像を読み取る。

ドット配列生成部22は、読み取った符号画像からドットを検出し、ドットの位置を参照して、ドット配列を生成する。尚、符号画像からのドット検出の前処理として、読み取った画像に含まれるノイズを除去するための処理も行う。ここで、ノイズには、例えば、撮像素子感度のばらつきや電子回路により発生するノイズがある。ノイズ除去の処理の種類は、撮像素子の特性に合わせるべきだが、ぼかし処理やアンシャープマスキング等の先鋭化処理を適用するとよい。また、ドットの検出は、次のように行う。即ち、まず、2値化処理によりドット画像の部分と、その他の背景画像の部分とを切り分け、2値化された個々の画像位置からドットの位置を検出する。その際、2値化された画像にノイズ成分が多数含まれる場合があるため、2値化された画像の面積や形状によりドットの判定を行うフィルタ処理を組み合わせる必要がある。その後、ドット配列の生成は、2次元の配列上で、例えば、ドットがある位置を「1」、ドットがない位置を「0」というように、画像として検出したドットをデジタルデータに置き換えることにより行う。本実施の形態では、複数のパターン画像を取得する画像取得手段の一例として、ドット配列生成部22を設け

ている。

【 0 0 4 1 】

ブロック検出部 2 3 は、ドット配列上で、符号ブロック内のパターンブロックを検出する。即ち、符号ブロックと同じ大きさ及び形状のフレームで、パターンブロックと同じ大きさ及び形状のブロックを有するブロックフレームをドット配列上で適宜動かし、ブロック内のドット数が均等になる位置を正しいフレーム位置とし、各ブロック内のパターン値を格納した符号配列を生成する。本実施の形態では、誤りパターン画像を検出する検出手段の一例として、ブロック検出部 2 3 を設けている。

【 0 0 4 2 】

同期符号検出部 2 4 は、ドット配列から検出された各符号パターンの種類を参照して、同期符号を検出する。また、同期符号検出部 2 4 は、検出した同期符号に基づいて、画像の回転の判定も行う。例えば、正方形の符号パターンを用いた場合、90度単位で回転している可能性がある。そこで、検出した同期符号が4種類の同期パターンのいずれに対応しているかによって、その向きを検出する。また、長方形の符号パターンを用いた場合、180度単位で回転している可能性がある。そこで、検出された同期符号が2種類の同期パターンのいずれに対応しているかによって、その向きを検出する。更に、同期符号検出部 2 4 は、このように検出された回転角度だけ符号配列を回転させて、符号配列を正しい向きに設定する。

【 0 0 4 3 】

識別符号検出部 3 0 は、角度が補正された符号配列から、同期符号の位置を基準にして、識別符号を検出する。

識別符号復号部 3 2 は、上述した R S 符号の符号化処理で用いたパラメータ（ブロック数等）と同じパラメータを用いて識別符号を復号し、識別情報を出力する。

【 0 0 4 4 】

X座標符号検出部 4 0 は、角度が補正された符号配列から、同期符号の位置を基準にしてX座標符号を検出する。本実施の形態では、部分列を取得する部分列取得手段の一例として、X座標符号検出部 4 0 を設けている。

X座標符号復号部 4 2 は、検出されたX座標符号からM系列の部分系列を取り出し、画像生成に使用したM系列におけるこの部分系列の位置を参照し、この位置を符号ブロックのシフト量で補正した値をX座標情報として出力する。本実施の形態では、所定位置を特定する特定手段の一例として、X座標符号復号部 4 2 を設けている。

【 0 0 4 5 】

Y座標符号検出部 4 5 は、角度が補正された符号配列から、同期符号の位置を基準にしてY座標符号を検出する。本実施の形態では、部分列を取得する部分列取得手段の一例として、Y座標符号検出部 4 5 を設けている。

Y座標符号復号部 4 7 は、検出されたY座標符号からM系列の部分系列を取り出し、画像生成に使用したM系列におけるこの部分系列の位置を参照し、この位置を符号ブロックのシフト量で補正した値をY座標情報として出力する。本実施の形態では、所定位置を特定する特定手段の一例として、Y座標符号復号部 4 7 を設けている。

【 0 0 4 6 】

情報出力部 5 0 は、識別符号復号部 3 2、X座標符号復号部 4 2、Y座標符号復号部 4 7 からそれぞれ取得した識別情報、X座標情報、Y座標情報を出力する。

【 0 0 4 7 】

次いで、この画像処理装置 2 0 の動作を説明する。尚、この動作の説明では、9 C 3 方式の符号パターンが図 5 のレイアウトで配置されていることを前提とする。

まず、画像読取部 2 1 が、符号画像が印刷された媒体から、所定の大きさの領域の符号画像を読み取る。

次に、ドット配列生成部 2 2 が、ドットを検出した位置に「1」を、ドットを検出なかった位置に「0」を設定したドット配列を生成する。

その後、ブロック検出部 2 3 が、このドット配列にブロックフレームを重ねて、全ての

10

20

30

40

50

パターンブロック内のドットの数「3」となるブロックフレームの位置を探索することで、パターンブロックの境界を検出する。本実施の形態において、符号ブロックは、パターンブロックが5個×5個で配置されたものとしている。従って、ブロックフレームとしても、5ブロック×5ブロックの大きさのものを用いる。

【0048】

ここで、ブロック検出部23の動作について説明する。

図11は、ブロック検出部23の動作例を示したフローチャートである。

まず、ブロック検出部23は、ドット配列生成部22からドット配列を取得する（ステップ201）。このドット配列の大きさは、（復号に必要なブロック数×ブロックの一边のドット数+ブロックの一边のドット数-1）²である。本実施の形態では、復号に必要なブロック数は5×5であり、ブロックの一边のドット数が3であるので、17×17のドット配列を取得する。

10

【0049】

次に、取得したドット配列に、ブロックフレームを重ね合わせる（ステップ202）。そして、カウンタI、Jに「0」を代入し、MaxBNにも「0」を代入する（ステップ203）。ここで、I、Jは、ブロックフレームを初期位置から移動させたステップ数をカウントするものである。画像の1ラインごとにブロックフレームを移動させ、そのとき移動させたライン数をカウンタI、Jでカウントする。また、MaxBNは、ブロック内で検出されるドット数が「3」となるブロック数をブロックフレームを移動させながらカウントしていったときに、最大となるカウント値を記録するものである。

20

【0050】

次に、ブロック検出部23は、ブロックフレームをX方向にI、Y方向にJ移動させる（ステップ204）。初期状態においてI、Jは「0」であるので、ブロックフレームは移動しない。そして、ブロックフレームの各ブロックに含まれるドット数をカウントして、ドット数が「3」となるブロックの数をカウントする。カウントしたブロック数はIB[I, J]に格納する（ステップ205）。IB[I, J]のI、Jには、ブロックフレームの移動量を示すIとJの値がそれぞれ記録される。

【0051】

次に、ブロック検出部23は、IB[I, J]とMaxBNとを比較する（ステップ206）。MaxBNは初期値が「0」であるので、最初の比較では、IB[I, J]がMaxBNよりも大きくなる。この場合、MaxBNにIB[I, J]の値を代入すると共に、MXにIの値を、MYにJの値を代入する（ステップ207）。尚、IB[I, J]がMaxBN以下である場合は、MaxBN、MX、MYの値はそのままにしておく。

30

【0052】

その後、ブロック検出部23は、I=2であるかどうかを判定する（ステップ208）。

ここで、I=2でない場合には、Iに「1」を加算する（ステップ209）。そして、ステップ204、205の処理を繰り返し行い、IB[I, J]とMaxBNとを比較していく（ステップ206）。

IB[I, J]が前回までのIB[I, J]の最大値であるMaxBNよりも大きいと、MaxBNにIB[I, J]の値を代入し、そのときのIの値をMXに、Jの値をMYに代入する（ステップ207）。また、MaxBNのほうがIB[I, J]よりも大きい場合には、I=2であるかどうかを判定する（ステップ208）。I=2となると、次にJ=2であるかどうかを判定する（ステップ210）。J=2でない場合には、Iに「0」を代入し、Jに「1」を加算する（ステップ211）。このような手順を繰り返し行い、（I, J）が（0, 0）から（2, 2）までで、IB[I, J]が最大のものを検出する。

40

【0053】

I=2、J=2までの処理が終了すると、ブロック検出部23は、保存しているMaxBNと閾値TBとを比較する（ステップ212）。閾値TBは、ドット数が「3」のプロ

50

ックの数が復号可能な程度のものを判定するための閾値である。

ここで、 $MaxBN$ が閾値 TB よりも大きい場合には、ブロックフレームを MX 、 MY の位置に固定し、その位置で各ブロックのパターン値を検出する。そして、検出したパターン値を、各ブロックを識別する変数 X 、 Y と共に符号配列 $PA[X, Y]$ としてメモリに記録する(ステップ213)。尚、このとき、対応するパターン値に変換することができなければ、パターン値として用いられていない「-1」を記録する。そして、ブロック検出部23は、 MX 、 MY と、符号配列 $PA[X, Y]$ とを同期符号検出部24に出力する(ステップ214)。

一方、 $MaxBN$ が閾値 TB 以下である場合には、画像のノイズが大きく復号は不可能と判定し、復号不能を出力する(ステップ215)。

10

【0054】

次に、同期符号検出部24の動作について説明する。

図12は、同期符号検出部24の動作例を示したフローチャートである。

まず、同期符号検出部24は、ブロック検出部23から MX 、 MY と、符号配列 $PA[X, Y]$ とを取得する(ステップ251)。

次に、同期符号検出部24は、 K 、 L に「1」を代入する(ステップ252)。尚、 K は X 方向のブロック数を示すカウンタであり、 L は Y 方向のブロック数を示すカウンタである。

【0055】

次に、同期符号検出部24は、 $PA[K, L]$ のパターン値が「64」かどうかを判定する(ステップ253)。

20

$PA[K, L]$ のパターン値が「64」であれば、符号配列 $PA[X, Y]$ の回転は必要ないと判定し、同期符号のあるブロックの X 座標 SX に K を代入し、 Y 座標 SY に L を代入する。また、ブロックフレームの X 方向への移動量 $ShiftX$ に MX を代入し、 Y 方向への移動量 $ShiftY$ に MY を代入する(ステップ254)。

【0056】

次に、同期符号検出部24は、 $PA[K, L]$ のパターン値が「65」かどうかを判定する(ステップ255)。

$PA[K, L]$ のパターン値が「65」であれば、符号配列 $PA[X, Y]$ を左方向に90度回転させる(ステップ256)。図4(b)に示すようにパターン値「65」の符号パターンは、パターン値「64」の符号パターンを右方向に90度回転させた画像であるので、逆方向に90度回転させて画像を正立させている。尚、このとき、符号配列 $PA[X, Y]$ 内の全てのパターン値が、左方向に90度回転させた場合のパターン値に変換される。

30

また、この回転に伴って、同期符号のあるブロックの X 座標 SX に L を代入し、 Y 座標 SY に $6 - K$ を代入する。また、ブロックフレームの X 方向への移動量 $ShiftX$ に MY を代入し、 Y 方向への移動量 $ShiftY$ に $2 - MX$ を代入する(ステップ257)。

【0057】

次に、同期符号検出部24は、 $PA[K, L]$ のパターン値が「66」かどうかを判定する(ステップ258)。

40

$PA[K, L]$ のパターン値が「66」であれば、符号配列 $PA[X, Y]$ を左方向に180度回転させる(ステップ259)。図4(b)に示すようにパターン値「66」の符号パターンは、パターン値「64」の符号パターンを180度回転させた画像であるので、パターン値「66」の符号パターンを180度回転させて画像を正立させている。尚、このとき、符号配列 $PA[X, Y]$ 内の全てのパターン値が、180度回転させた場合のパターン値に変換される。

また、この回転に伴って、同期符号のあるブロックの X 座標 SX に $6 - K$ を代入し、 Y 座標 SY に $6 - L$ を代入する。また、ブロックフレームの X 方向への移動量 $ShiftX$ に $2 - MX$ を代入し、 Y 方向への移動量 $ShiftY$ に $2 - MY$ を代入する(ステップ260)。

50

【 0 0 5 8 】

次に、同期符号検出部 2 4 は、 $PA[K, L]$ のパターン値が「 6 7 」かどうかを判定する（ステップ 2 6 1）。

$PA[K, L]$ のパターン値が「 6 7 」であれば、符号配列 $PA[X, Y]$ を左方向に 2 7 0 度回転させる（ステップ 2 6 2）。図 4（b）に示すようにパターン値「 6 7 」の符号パターンは、パターン値「 6 4 」の符号パターンを右に 2 7 0 度回転させた画像であるので、逆方向に 2 7 0 度回転させて画像を正立させている。尚、このとき、符号配列 $PA[X, Y]$ 内の全てのパターン値が、左方向に 2 7 0 度回転させた場合のパターン値に変換される。

また、この回転に伴って、同期符号のあるブロックの X 座標 SX に $6 - L$ を代入し、Y 座標 SY に K を代入する。また、ブロックフレームの X 方向への移動量 $ShiftX$ に $2 - MY$ を代入し、Y 方向への移動量 $ShiftY$ に MX を代入する（ステップ 2 6 3）。

【 0 0 5 9 】

そして、ステップ 2 5 4、2 5 7、2 6 0、2 6 3 で、 SX 、 SY 、 $ShiftX$ 、 $ShiftY$ に値が代入された場合、同期符号検出部 2 4 は、 $PA[X, Y]$ とこれらの値を識別符号検出部 3 0、X 座標符号検出部 4 0、Y 座標符号検出部 4 5 に出力する（ステップ 2 6 4）。

また、 $PA[K, L]$ がパターン値「 6 4 」～「 6 7 」のいずれでもなければ、同期符号検出部 2 4 は、 $K = 5$ であるかどうかを判定する（ステップ 2 6 5）。 $K = 5$ でない場合には、 K に「 1 」を加算し（ステップ 2 6 6）、ステップ 2 5 3 に戻る。 $K = 5$ であれば、 $L = 5$ であるかどうかを判定する（ステップ 2 6 7）。 $L = 5$ でない場合には、 K に「 1 」を代入し、 L に「 1 」を加算し（ステップ 2 6 8）、ステップ 2 5 3 に戻る。即ち、ステップ 2 5 3 ～ 2 6 4 の処理を、パターン値「 6 4 」～「 6 7 」のブロックを検出するまで K 、 L の値を変更しながら繰り返し行う。また、 $K = 5$ 、 $L = 5$ になっても、パターン値「 6 4 」～「 6 7 」のブロックを検出することができなかった場合には、復号不能の判定信号を出力する（ステップ 2 6 9）。

【 0 0 6 0 】

次に、識別符号検出部 3 0 及び識別符号復号部 3 2 の動作について説明する。

図 1 3 は、識別符号検出部 3 0 及び識別符号復号部 3 2 の動作例を示したフローチャートである。

まず、識別符号検出部 3 0 は、同期符号検出部 2 4 から符号配列 $PA[X, Y]$ 、 SX 、 SY を取得する（ステップ 3 0 1）。

次に、識別符号検出部 3 0 は、識別符号配列 $IA[X, Y]$ の全ての要素を「 - 1 」で初期化する（ステップ 3 0 2）。尚、この「 - 1 」は、パターン値として用いられていない番号とする。そして、符号ブロックにおける各ブロックを識別するためのカウンタ IX 、 IY に「 1 」を代入する（ステップ 3 0 3）。ここで、 IX は、X 方向のブロック数を示すカウンタであり、 IY は、Y 方向のブロック数を示すカウンタである。

【 0 0 6 1 】

また、識別符号検出部 3 0 は、 $IY - SY$ が「 5 」で割り切れるかどうかを判定する（ステップ 3 0 4）。即ち、 IY で特定される行に同期符号が配置されているかどうかを判定する。

ここで、 $IY - SY$ が「 5 」で割り切れた場合、つまり、この行に同期符号が配置されている場合は、識別符号を取り出す対象ではないため、 IY に「 1 」を加算し（ステップ 3 0 5）、ステップ 3 0 4 へ進む。

一方、 $IY - SY$ が「 5 」で割り切れなかった場合、つまり、この行に同期符号が配置されていない場合は、 $IX - SX$ が「 5 」で割り切れるかどうかを判定する（ステップ 3 0 6）。即ち、 IX で特定される列に同期符号が配置されているかどうかを判定する。

【 0 0 6 2 】

ここで、 $IX - SX$ が「 5 」で割り切れた場合、つまり、この列に同期符号が配置されている場合は、識別符号を取り出す対象ではないため、 IX に「 1 」を加算し（ステップ

10

20

30

40

50

307)、ステップ306へ進む。

一方、 $IX - SX$ が「5」で割り切れなかった場合、つまり、この列に同期符号が配置されていない場合、識別符号検出部30は、 $IA[(IX - SX) \bmod 5, (IY - SY) \bmod 5]$ に $PA[IX, IY]$ を代入する(ステップ308)。

【0063】

そして、 $IX = 5$ であるかどうかを判定する(ステップ309)。

ここで、 $IX = 5$ でない場合には、 IX に「1」を加算して(ステップ307)、ステップ306～308の処理を $IX = 5$ となるまで繰り返し行う。また、 $IX = 5$ となると、次に、 $IY = 5$ であるかどうかを判定する(ステップ310)。 $IY = 5$ でない場合には、 IX に「1」を代入し(ステップ311)、 IY に「1」を加算して(ステップ305)、ステップ304～309の処理を $IY = 5$ となるまで繰り返し行う。また、 $IY = 5$ となると、識別符号復号部32の処理に移る。

【0064】

即ち、識別符号復号部32は、 $IA[X, Y]$ を復号できるかどうかを判定する(ステップ312)。

ここで、 $IA[X, Y]$ を復号できると判定すると、識別符号復号部32は、 $IA[X, Y]$ から識別情報を得る(ステップ313)。また、 $IA[X, Y]$ を復号できないと判定すると、識別情報に N/A を代入する(ステップ314)。

【0065】

次に、X座標符号検出部40の動作について説明する。

図14は、X座標符号検出部40の動作例を示したフローチャートである。

まず、X座標符号検出部40は、同期符号検出部24から符号配列 $PA[X, Y]$ 、 SX 、 SY 、 $ShiftX$ 、 $ShiftY$ を取得する(ステップ401)。

次に、X座標符号検出部40は、X座標符号配列 $XA[X]$ の全ての要素を「-1」で初期化する(ステップ402)。尚、この「-1」は、パターン値として用いられていない番号とする。そして、符号ブロックにおける各ブロックを識別するためのカウンタ IX 、 IY に「1」を代入する。ここで、 IX は、X方向のブロック数を示すカウンタであり、 IY は、Y方向のブロック数を示すカウンタである。更に、X座標符号検出部40は、X座標符号配列における各要素を識別するためのカウンタ KX にも「1」を代入する(ステップ403)。

【0066】

また、X座標符号検出部40は、 $IY - SY$ が「5」で割り切れるかどうかを判定する(ステップ404)。即ち、 IY で特定される行に同期符号が配置されているかどうかを判定する。

ここで、 $IY - SY$ が「5」で割り切れなかった場合、つまり、この行に同期符号が配置されていない場合は、X座標符号を取り出す対象ではないため、 IY に「1」を加算し(ステップ405)、ステップ404へ進む。

一方、 $IY - SY$ が「5」で割り切れた場合、つまり、この行に同期符号が配置されている場合、X座標符号検出部40は、 $IX - SX$ が「5」で割り切れるかどうかを判定する(ステップ406)。即ち、 IX で特定される列に同期符号が配置されているかどうかを判定する。

【0067】

ここで、 $IX - SX$ が「5」で割り切れた場合、つまり、この列に同期符号が配置されている場合は、X座標符号を取り出す対象ではないため、 IX に「1」を加算し(ステップ407)、ステップ406へ進む。

一方、 $IX - SX$ が「5」で割り切れなかった場合、つまり、この列に同期符号が配置されていない場合、X座標符号検出部40は、 $XA[KX]$ に $PA[IX, IY]$ を代入する(ステップ408)。

【0068】

そして、 $IX = 5$ であるかどうかを判定する(ステップ409)。

ここで、 $IX = 5$ でない場合には、 KX に「1」を加算し(ステップ410)、 IX に「1」を加算して(ステップ407)、ステップ406～408の処理を $IX = 5$ となるまで繰り返し行う。また、 $IX = 5$ となると、 X 座標符号復号部42の処理に移る。

【0069】

次に、 X 座標符号復号部42の動作について説明する。

図15は、 X 座標符号復号部42の動作例を示したフローチャートである。

まず、 X 座標符号復号部42は、特定不能誤りの数を $iErasure$ に設定する(ステップ421)。ここで、特定不能誤りの数は、 $XA[X, Y] = -1$ となっているブロックを数えることで求める。

次に、 X 座標符号復号部42は、 $iErasure$ が「2」を超えているかどうかを判定する(ステップ422)。ここで、「2」は、誤り訂正復号を行うことの可能な $iErasure$ の数である。そして、 $iErasure$ が「2」を超えていれば、つまり、「3」又は「4」であれば、 X 座標情報に N/A を代入して(ステップ439)、処理を終了する。また、 $iErasure$ が「2」を超えていなければ、復号処理を継続する。

【0070】

その場合、まず、 X 座標符号復号部42は、 $iErasure$ が「0」であるかどうかを判定する(ステップ423)。即ち、 X 座標符号検出部40が検出した4ブロックの中に、特定不能誤りがないかどうかを判定する。

ここで、 $iErasure$ が「0」であれば、即ち、特定不能誤りがなければ、最初の3ブロックのパターン値から11ビットのビット値を算出する(ステップ424)。このビット値 $read_num$ は、 X 座標符号検出部40が検出した4ブロックのうち j 番目のブロックにおけるパターン値を $read_sequence[j]$ ($j = 0 \sim 3$)とすると、

$read_num = read_sequence[0] + 16 \times read_sequence[1] + (16 \times 16 \times read_sequence[2] >> 1)$ により求められる。尚、「>>」は、ビット列を右にシフトすることを示す。求めるのは11ビットのビット値であるので、3番目のブロックについては、前から3ビットが計算の対象になるからである。

【0071】

これにより、 X 座標符号復号部42は、座標テーブルから座標値を検索する(ステップ425)。座標テーブルにおいて、11ビット値 i に対する座標値を $POSITION_TABLE[i]$ と表記することになると、 X 座標の座標値 im は、

$im = POSITION_TABLE[read_num]$

により求められる。ここで、 $POSITION_TABLE$ は、パターン列の連続する3つの $PATTERN_SEQUENCE[j]$ 、 $PATTERN_SEQUENCE[j+1]$ 、 $PATTERN_SEQUENCE[j+2]$ からステップ424と同じ方法で得られた値と j との対応を表している。

【0072】

尚、ここでは、11ビットからビット値を計算している。本実施の形態で使用している M 系列は11ビットであり、その計算方法を示しているが、11ビット以上で計算してもよい。例えば、3つの符号パターンから得られる12ビットで計算することも可能で、余分な1ビットで誤り検出を行うことができる。この場合、上記の $POSITION_TABLE$ は、座標値 j と、 $PATTERN_SEQUENCE[j] + 16 \times PATTERN_SEQUENCE[j+1] + 16 \times 16 \times PATTERN_SEQUENCE[j+2]$ により得られた10進数とを対応付けたものになる。上述したように11ビットから $POSITION_TABLE$ を作成した場合は、ビット値に対応して座標値が出力されるが、12ビットから $POSITION_TABLE$ を作成した場合は、ビット値に対して座標値が対応しないことがある。このときは「-1」を出力するようにする。

従って、ステップ424で12ビットのビット値を算出し、ステップ425でこのビット値により座標テーブルを検索した場合は、その直後に、ステップ426として、座標テ

10

20

30

40

50

ーブルから座標値が得られたか、つまり、検索が成功したかどうかを判定するステップを設けてもよい。このステップでは、 $i m > - 1$ となった場合に、検索が成功したと判断するとよい。

【 0 0 7 3 】

その後、X座標符号復号部42は、4番目のブロックが一致しているかどうかを判定する(ステップ427)。この判定は、座標値jに対するパターン値をPATTERN__SEQUENCE[j]で表すことにすると、read__sequence[3]とPATTERN__SEQUENCE[i m + 3]を比較することにより行われる。ここで、4番目のブロックが一致していなければ、X座標符号検出部40が検出したX座標符号の4番目のブロックを訂正する(ステップ428)。つまり、read__sequence[3]にPATTERN__SEQUENCE[i m + 3]を代入する。また、4番目のブロックが一致していれば、4番目のブロックを訂正する処理は行わない。そして、何れの場合も、X座標情報にimを格納する(ステップ429)。尚、ステップ428は、座標復号後に、検出したパターン列read__sequence[j]を利用しない場合は、実施する必要は無い。

【 0 0 7 4 】

一方、ステップ423で特定不能誤りがあったと判定された場合や、12ビット値で検索してステップ426で検索が成功しなかったと判定された場合は、特定不能誤りや誤特定誤りが発生していないブロックを用いてX座標情報を求める処理に移る。

即ち、X座標符号復号部42は、画像読取部21が前回のフレームで復号が成功しているかどうかを判定する(ステップ430)。この判定は、変数pastPositionを用いて行う。変数pastPositionには、前回のフレームにおける復号が成功すれば、復号結果である座標値が格納され、失敗すれば「- 1」が格納されている。従って、pastPosition 0である場合に、前回のフレームで復号が成功したと判断するとよい。そして、前回のフレームで復号が成功していなければ、後述する照合処理における検索開始位置を示す変数startPositionに「0」を代入する(ステップ431)。また、前回のフレームで復号が成功していれば、後述する照合処理における検索開始位置を示す変数startPositionにpastPositionの値を代入する(ステップ432)。

【 0 0 7 5 】

その後、X座標符号復号部42は、iErasureが「2」未満であるかどうかを判定する(ステップ433)。即ち、X座標符号検出部40が検出した4ブロックのうち、特定不能誤りが発生したブロックが1つであるかどうかを判定する。そして、1つであれば、後述する3ブロック照合処理を行い(ステップ434)、照合が成功したかどうかを判定する(ステップ435)。

ここで、この3ブロック照合処理によりX座標情報の復号が成功した場合は、処理を終了する。

【 0 0 7 6 】

一方、この3ブロック照合処理によりX座標符号の復号が成功しなかった場合は、前回のフレームで復号が成功しているかどうかを判定する(ステップ436)。この判定も、上記と同様、pastPosition 0であるかどうかを調べることにより行えばよい。このとき、前回のフレームで復号が成功していなければ、X座標情報にN/Aを代入して(ステップ439)、処理を終了する。既に述べた通り、2ブロック照合処理では、2ブロックが合致する箇所が7箇所あるため、前回のフレームで復号が成功しておらず検索開始位置を設定できないと、妥当な座標値を得られないためである。

その後、X座標符号復号部42は、後述する2ブロック照合処理を行い(ステップ437)、照合が成功したかどうかを判定する(ステップ438)。

ここで、この2ブロック照合処理によりX座標情報の復号が成功した場合は、処理を終了する。

一方、この2ブロック照合処理によりX座標符号の復号が成功しなかった場合は、X座

10

20

30

40

50

標情報にN/Aを代入して(ステップ439)、処理を終了する。

【0077】

次に、図15の3ブロック照合処理について詳細に説明する。

図16は、X座標符号復号部42による3ブロック照合処理の動作例を示したフローチャートである。

まず、X座標符号復号部42は、M系列テーブル走査ループを開始する(ステップ441)。具体的には、M系列テーブルの走査位置を表す変数*i*の初期値を「0」とし、*i*がM系列の長さSEQUENCE_LENGTH未満である限り、*i*に「1」を加えながらループを実行する。

次に、M系列テーブル走査ループ内で、X座標符号復号部42は、左右照合ループを開始する(ステップ442)。具体的には、検索開始位置の左右何れを照合位置とするかを表す変数*i*_pの初期値を「-1」とし、*i*_pが「2」未満である限り、*i*_pに「2」を加えながらループを実行する。つまり、*i*_p = -1として1回ループを実行し、*i*_p = 1として更に1回ループを実行する。

【0078】

この左右照合ループ内で、X座標符号復号部42は、まず、照合位置がM系列の範囲内にあるかどうかを判定する(ステップ443)。照合位置は、startPosition + (*i* × *i*_p)と表わされるので、0 ≤ startPosition + (*i* × *i*_p) ≤ SEQUENCE_LENGTH - 3が満たされるかを調べることにより、ここでの判定は行われる。

【0079】

その結果、照合位置がM系列の範囲内にあると判定されれば、照合用カウンタmatchBlockに「0」を代入し(ステップ444)、4ブロック走査ループを開始する(ステップ445)。具体的には、ブロック数をカウントする変数*j*の初期値を「0」とし、*j*が「4」未満である限り、*j*に「1」を加えながらループを実行する。

この4ブロック走査ループ内で、X座標符号復号部42は、X座標符号検出部40が検出したX座標符号におけるパターン値と、M系列テーブルにおけるパターン値とが一致しているかどうかを判定する(ステップ446)。つまり、readSequence[*j*] = PATTERN_SEQUENCE[startPosition + (*i* × *i*_p) + *j*]であるかどうかを判定する。その結果、両者が一致すれば、照合用カウンタmatchBlockに「1」を加算する(ステップ447)が、両者が一致しなければ、そのような処理は行わない。そして、次の*j*についてループを実行し、*j*が「4」になる前にループを終了する(ステップ448)。

【0080】

これにより、X座標符号復号部42は、4ブロックのうちの3ブロックが一致する部分がM系列テーブルの中にあるかどうかを判定する(ステップ449)。つまり、上記4ブロック走査ループでmatchBlockの値が「3」となっているかどうかを判定する。

その結果、3ブロックが一致する部分があれば、つまり、matchBlockの値が「3」であれば、X座標符号復号部42は、1ブロック訂正ループを開始する(ステップ450)。具体的には、ブロックの数をカウントする変数*j*の初期値を「0」とし、*j*が「4」未満である限り、*j*に「1」を加えながらループを実行する。

この1ブロック訂正ループ内で、X座標符号復号部42は、*j*番目のブロックを訂正する(ステップ451)。即ち、readSequence[*j*]にPATTERN_SEQUENCE[startPosition + (*i* × *i*_p) + *j*]を格納する。そして、次の*j*についてループを実行し、*j*が「4」になる前にループを終了する(ステップ452)。尚、ステップ451は、座標復号後に、検出したパターン列readSequence[*j*]を利用しない場合は、実施する必要は無い。

その後、X座標符号復号部42は、X座標情報に*i*を格納し(ステップ453)、3ブロック照合が成功した旨の情報を返す(ステップ454)。

【 0 0 8 1 】

一方、ステップ 4 4 9 で 3 ブロックが一致する部分がなければ、つまり、`matchBlock` の値が「 3 」でなければ、X 座標符号復号部 4 2 は、次の `ip` についてループを実行し、`ip` が「 2 」になる前にループを終了する（ステップ 4 5 5）。更に、X 座標符号復号部 4 2 は、次の `i` についてループを実行し、`i` が `SEQUENCE_LENGTH` になる前にループを終了する（ステップ 4 5 6）。

このように左右照合ループの途中でステップ 4 5 0 ~ 4 5 4 の処理へ抜けることなく M 系列テーブル走査ループが終了した場合、この 3 ブロック照合処理で X 座標情報は得られていないことになる。従って、この場合は、3 ブロック照合が失敗した旨の情報を返す（ステップ 4 5 7）。

10

【 0 0 8 2 】

次に、図 1 5 の 2 ブロック照合処理について詳細に説明する。

図 1 7 は、X 座標符号復号部 4 2 による 2 ブロック照合処理の動作例を示したフローチャートである。

まず、X 座標符号復号部 4 2 は、M 系列テーブル走査ループを開始する（ステップ 4 6 1）。具体的には、M 系列テーブルの走査位置を表す変数 `i` の初期値を「 0 」とし、`i` が所定の長さ（例えば `SEQUENCE_LENGTH / 100`）未満である限り、`i` に「 1 」を加えながらループを実行する。

次に、M 系列テーブル走査ループ内で、X 座標符号復号部 4 2 は、左右照合ループを開始する（ステップ 4 6 2）。具体的には、検索開始位置の左右何れを照合位置とするかを表す変数 `ip` の初期値を「 - 1 」とし、`ip` が「 2 」未満である限り、`ip` に「 2 」を加えながらループを実行する。つまり、`ip = - 1` として 1 回ループを実行し、`ip = 1` として更に 1 回ループを実行する。

20

【 0 0 8 3 】

この左右照合ループ内で、X 座標符号復号部 4 2 は、まず、照合位置が M 系列の範囲内にあるかどうかを判定する（ステップ 4 6 3）。照合位置は、`startPosition + (i × ip)` と表わされるので、`0 ≤ startPosition + (i × ip) ≤ SEQUENCE_LENGTH - 3` が満たされるかを調べることにより、ここでの判定は行われる。

【 0 0 8 4 】

30

その結果、照合位置が M 系列の範囲内にあると判定されれば、照合用カウンタ `matchBlock` に「 0 」を代入し（ステップ 4 6 4）、4 ブロック走査ループを開始する（ステップ 4 6 5）。具体的には、ブロック数をカウントする変数 `j` の初期値を「 0 」とし、`j` が「 4 」未満である限り、`j` に「 1 」を加えながらループを実行する。

この 4 ブロック走査ループ内で、X 座標符号復号部 4 2 は、X 座標符号検出部 4 0 が検出した X 座標符号におけるパターン値と、M 系列テーブルにおけるパターン値とが一致しているかどうかを判定する（ステップ 4 6 6）。つまり、`read_sequence[j] = PATTERN_SEQUENCE[startPosition + (i × ip) + j]` であるかどうかを判定する。その結果、両者が一致すれば、照合用カウンタ `matchBlock` に「 1 」を加算する（ステップ 4 6 7）が、両者が一致しなければ、そのような処理は行わない。そして、次の `j` についてループを実行し、`j` が「 4 」になる前にループを終了する（ステップ 4 6 8）。

40

【 0 0 8 5 】

これにより、X 座標符号復号部 4 2 は、4 ブロックのうちの 2 ブロックが一致する部分が M 系列テーブルの中にあるかどうかを判定する（ステップ 4 6 9）。つまり、上記 4 ブロック走査ループで `matchBlock` の値が「 2 」となっているかどうかを判定する。

その結果、2 ブロックが一致する部分があれば、つまり、`matchBlock` の値が「 2 」であれば、X 座標符号復号部 4 2 は、2 ブロック訂正ループを開始する（ステップ 4 7 0）。具体的には、ブロックの数をカウントする変数 `j` の初期値を「 0 」とし、`j` が

50

「4」未満である限り、jに「1」を加えながらループを実行する。

この2ブロック訂正ループ内で、X座標符号復号部42は、j番目のブロックを訂正する(ステップ471)。即ち、`read_sequence[j]`に`PATTERN_SEQUENCE[startPosition + (i * ip) + j]`を格納する。そして、次のjについてループを実行し、jが「4」になる前にループを終了する(ステップ472)。尚、ステップ471は、座標復号後に、検出したパターン列`read_sequence[j]`を利用しない場合は、実施する必要は無い。

その後、X座標符号復号部42は、X座標情報にiを格納し(ステップ473)、2ブロック照合が成功した旨の情報を返す(ステップ474)。

【0086】

10

一方、ステップ469で2ブロックが一致する部分が無ければ、つまり、`matchBlock`の値が「2」でなければ、X座標符号復号部42は、次のipについてループを実行し、ipが「2」になる前にループを終了する(ステップ475)。更に、X座標符号復号部42は、次のiについてループを実行し、iが`SEQUENCE_LENGTH`になる前にループを終了する(ステップ476)。

このように左右照合ループの途中でステップ470～474の処理へ抜けることなくM系列テーブル走査ループが終了した場合、この2ブロック照合処理でX座標情報は得られていないことになる。従って、この場合は、2ブロック照合が失敗した旨の情報を返す(ステップ477)。

【0087】

20

尚、ここでは、X座標符号検出部40及びX座標符号復号部42の動作のみ説明したが、Y座標符号検出部45及びY座標符号復号部47も同様の動作を行う。

以上により、本実施の形態における画像処理装置20の動作説明を終了する。

【0088】

ところで、上記説明では、前回のフレームで座標を復号できていれば、その座標をM系列テーブルにおける検索開始位置としたが、これには限らない。即ち、前回のフレームだけでなく、所定回数前から前回までのフレームで復号できた座標を用いて検索開始位置を決定してもよい。また、復号できた座標をそのまま検索開始位置とするのではなく、復号できた座標に基づく何らかの処理によって決定された座標を検索開始位置としてもよい。

【0089】

30

また、上記では、前回のフレームで座標を復号できていない場合、3ブロック照合処理では、M系列の先頭を検索開始位置としてM系列テーブルを検索し、2ブロック照合処理では、M系列テーブルの検索を行わないようにした。しかしながら、一般にk次のM系列で座標を表現し、任意のビット数を符号パターンに割り当ててを考えると、このような処理パターンだけとは限らない。例えば、座標を表す $m(m-1)$ 個の符号パターンのうち $p(1 \leq p \leq m)$ 個の符号パターンと、これとは別の $n(n-1)$ 個の符号パターンのうち $q(1 \leq q \leq n)$ 個の符号パターンとを連結したパターン列を用いて、 $(p+q)$ ブロック照合処理を行うとする。この場合、p個の符号パターンに対応するビット列の長さ p とq個の符号パターンに対応するビット列の長さ q の和がk以上であれば、任意の位置(例えばM系列の先頭)を検索開始位置とすればよい。一方、p個の符号パターンに対応するビット列の長さ p とq個の符号パターンに対応するビット列の長さ q の和がk未満であれば、M系列の検索は行わないほうがよい。尚、n個の符号パターンは、m個の符号パターンの後ろに隣接する符号パターンには限らず、m個の符号パターンの前に隣接する符号パターンであってもよい。

40

【0090】

ここで、本実施の形態における符号画像を利用したサービスの例を述べる。

図18(a)は、第1のサービスのイメージを具体的に示したものである。

このサービスでは、まず、紙面上のある領域に、その領域に対応するオブジェクトIDを符号画像として埋め込んでおく。そして、矢印の左側に示すように、紙面上のその領域にカメラ付きの携帯端末装置70を置く。すると、携帯端末装置70がオブジェクトID

50

を読み取り、矢印の右側に示すように、そのオブジェクトIDに対応した情報入力用UIを生成して表示する。ここで、情報入力用UIには、図示するように、文字種や文字数の情報を含めることが考えられる。また、情報入力用UIから入力された情報は、サーバに送信される。

【0091】

図18(b)は、第2のサービスのイメージを具体的に示したものである。

このサービスでも、まず、紙面上のある領域に、その領域に対応するオブジェクトIDを符号画像として埋め込んでおく。また、この例では、紙面上のその領域にはペンデバイス60で筆記がなされており、右上方向への矢印の右側に示すように、ペンデバイス60がオブジェクトIDと筆記データとを関連付けて保持しているものとする。そして、矢印の左側に示すように、紙面上のその領域にカメラ付きの携帯端末装置70を置く。すると、携帯端末装置70がオブジェクトIDを読み取り、ペンデバイス60からそのオブジェクトIDに関連付けられた筆記データを受信し、右下方向への矢印の右側に示すように、携帯端末70上で情報の追加編集ができるようにする。例えば、ペンデバイス60で紙面上に片仮名を筆記したとする。この場合、携帯端末装置70が片仮名を文字認識し、それをユーザの操作により漢字に変換する。つまり、漢字が分からない場合にペンデバイス60で片仮名を筆記しておき、携帯端末装置70で漢字変換してからサーバに送信する、といったことが可能になる。

【0092】

次に、本実施の形態における画像処理装置20の具体的なハードウェア構成について説明する。

まず、画像処理装置20を実現するペンデバイス60について説明する。

図19は、ペンデバイス60の機構を示した図である。

図示するように、ペンデバイス60は、ペン全体の動作を制御する制御回路61を備える。また、制御回路61は、入力画像から検出した符号画像を処理する画像処理部61aと、そこでの処理結果から識別情報及び座標情報を抽出するデータ処理部61bとを含む。

そして、制御回路61には、ペンデバイス60による筆記動作をペンチップ69に加わる圧力によって検出する圧力センサ62が接続されている。また、媒体上に赤外光を照射する赤外LED63と、画像を入力する赤外CMOS64も接続されている。更に、識別情報及び座標情報を記憶するための情報メモリ65と、外部装置と通信するための通信回路66と、ペンを駆動するためのバッテリー67と、ペンの識別情報(ペンID)を記憶するペンIDメモリ68も接続されている。

【0093】

尚、図10に示した画像読取部21は、例えば、図19の赤外CMOS64にて実現される。また、ドット配列生成部22は、例えば、図19の画像処理部61aにて実現される。更に、図10に示したブロック検出部23、同期符号検出部24、識別符号検出部30、識別符号復号部32、X座標符号検出部40、X座標符号復号部42、Y座標符号検出部45、Y座標符号復号部47、情報出力部50は、例えば、図19のデータ処理部61bにて実現される。

【0094】

また、図19の画像処理部61a又はデータ処理部61bにて実現される処理は、例えば、汎用のコンピュータで実現してもよい。そこで、かかる処理をコンピュータ90で実現するものとし、コンピュータ90のハードウェア構成について説明する。

図20は、コンピュータ90のハードウェア構成を示した図である。

図示するように、コンピュータ90は、演算手段であるCPU(Central Processing Unit)91と、記憶手段であるメインメモリ92及び磁気ディスク装置(HDD:Hard Disk Drive)93とを備える。ここで、CPU91は、OS(Operating System)やアプリケーション等の各種ソフトウェアを実行し、上述した各機能を実現する。また、メインメモリ92は、各種ソフトウェアやその実行に用いるデータ等を記憶する記憶領域であり、

10

20

30

40

50

磁気ディスク装置 93 は、各種ソフトウェアに対する入力データや各種ソフトウェアからの出力データ等を記憶する記憶領域である。

更に、コンピュータ 90 は、外部との通信を行うための通信 I/F 94 と、ビデオメモリやディスプレイ等からなる表示機構 95 と、キーボードやマウス等の入力デバイス 96 とを備える。

【0095】

尚、本実施の形態を実現するプログラムは、通信手段により提供することはもちろん、CD-ROM等の記録媒体に格納して提供することも可能である。

【図面の簡単な説明】

【0096】

10

【図1】9C2方式及び9C3方式における符号パターンの一例を示した図である。

【図2】9C2方式における全符号パターンの一例を示した図である。

【図3】9C3方式における全符号パターンの一例を示した図である。

【図4】9C2方式及び9C3方式における同期パターンの例を示した図である。

【図5】符号ブロックの基本的なレイアウトの例を示した図である。

【図6】M系列による座標の表現について説明するための図である。

【図7】符号ブロックの広範囲のレイアウトの例を示した図である。

【図8】3ブロック照合処理の概要を説明するための図である。

【図9】2ブロック照合処理の概要を説明するための図である。

【図10】本実施の形態における画像処理装置の機能構成例を示したブロック図である。

20

【図11】本実施の形態におけるブロック検出部の動作例を示したフローチャートである。

【図12】本実施の形態における同期符号検出部の動作例を示したフローチャートである。

【図13】本実施の形態における識別符号検出部等の動作例を示したフローチャートである。

【図14】本実施の形態におけるX座標符号検出部の動作例を示したフローチャートである。

【図15】本実施の形態におけるX座標符号復号部の動作例を示したフローチャートである。

30

【図16】本実施の形態におけるX座標符号復号部の動作例を示したフローチャートである。

【図17】本実施の形態におけるX座標符号復号部の動作例を示したフローチャートである。

【図18】本実施の形態における符号画像を用いたサービスの例を示した図である。

【図19】本実施の形態における画像処理装置を実現可能なペンデバイスの機構を示した図である。

【図20】本実施の形態を適用可能なコンピュータのハードウェア構成図である。

【符号の説明】

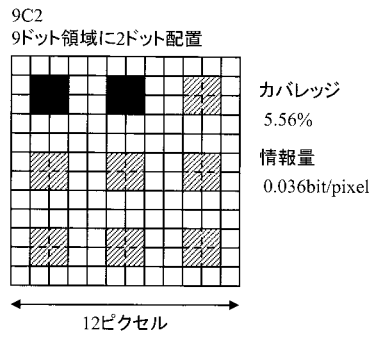
【0097】

40

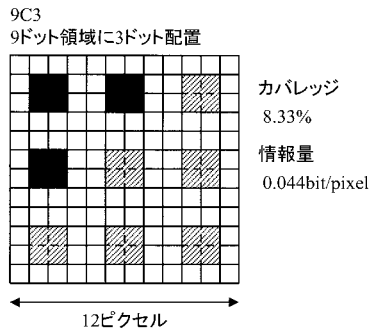
20...画像処理装置、21...画像読取部、22...ドット配列生成部、23...ブロック検出部、24...同期符号検出部、30...識別符号検出部、32...識別符号復号部、40...X座標符号検出部、42...X座標符号復号部、45...Y座標符号検出部、47...Y座標符号復号部、50...情報出力部

【図 1】

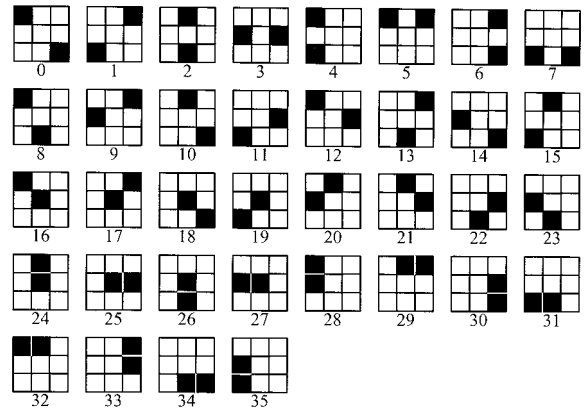
(a)



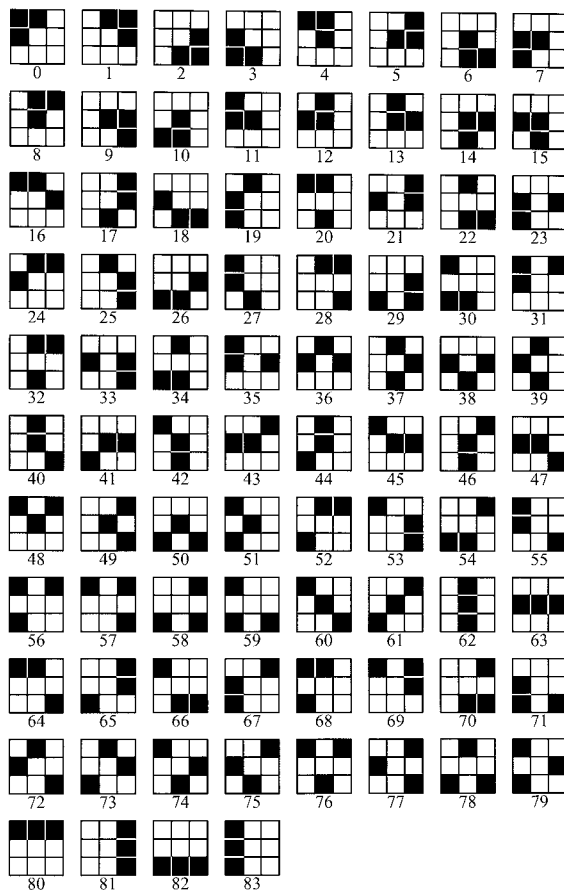
(b)



【図 2】

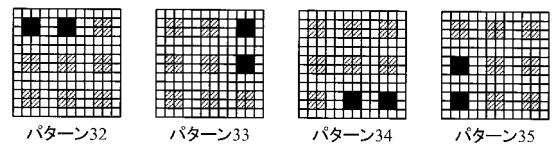


【図 3】

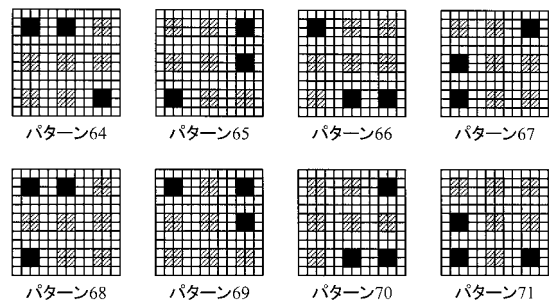


【図 4】

(a)

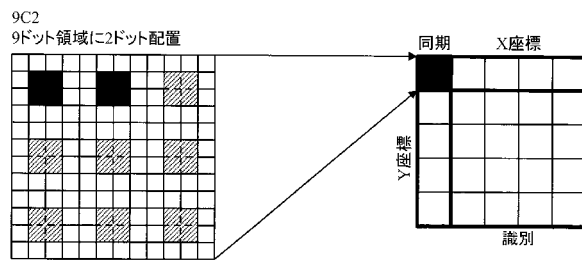


(b)

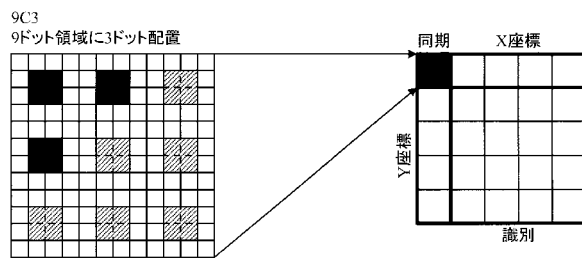


【図 5】

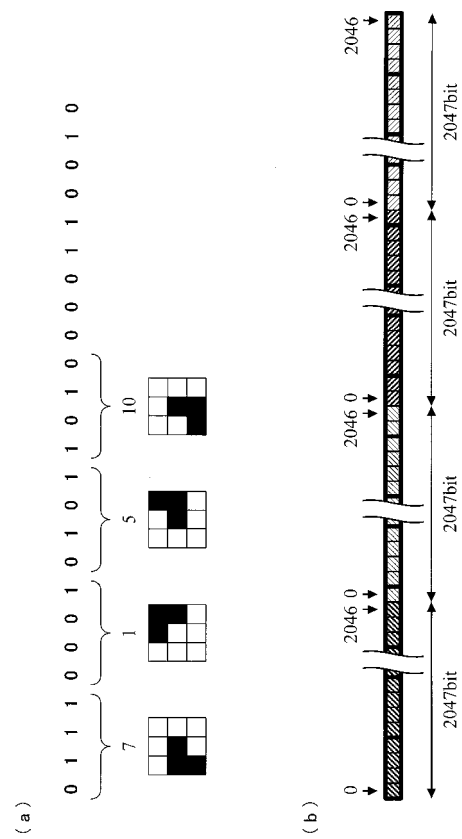
(a)



(b)



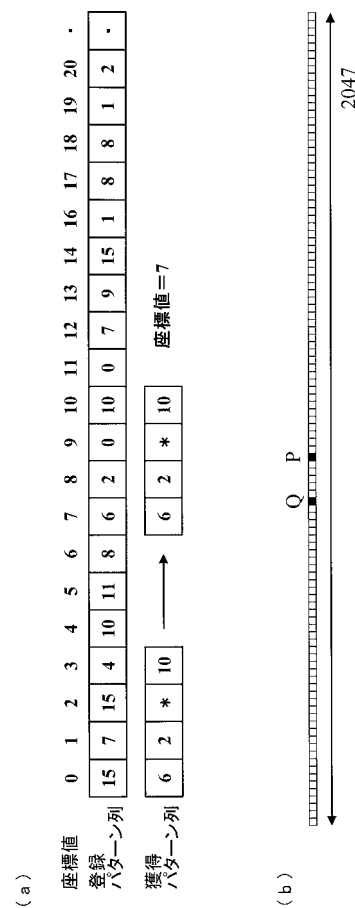
【図 6】



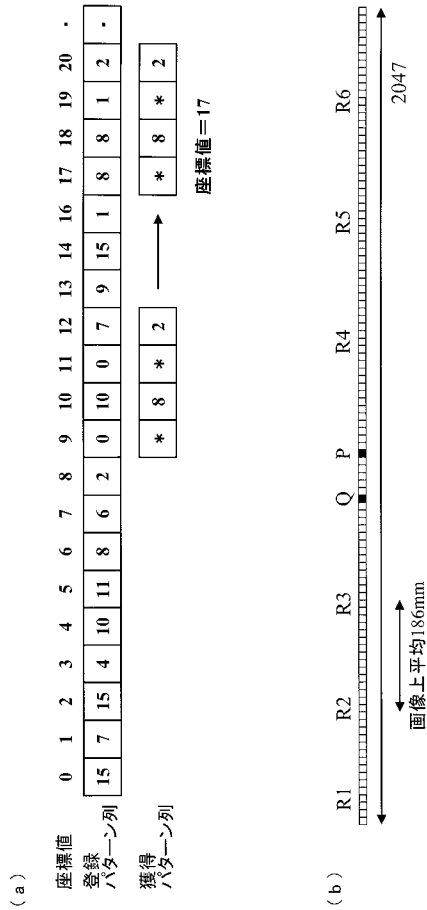
【図 7】

S	X01	X02	X03	X04	S	X05	X06	X07	X08	S	X09	X10	X11	X12
Y01	I01	I02	I03	I04	Y01	I01	I02	I03	I04	Y01	I01	I02	I03	I04
Y02	I05	I06	I07	I08	Y02	I05	I06	I07	I08	Y02	I05	I06	I07	I08
Y03	I09	I10	I11	I12	Y03	I09	I10	I11	I12	Y03	I09	I10	I11	I12
Y04	I13	I14	I15	I16	Y04	I13	I14	I15	I16	Y04	I13	I14	I15	I16
S	X01	X02	X03	X04	S	X05	X06	X07	X08	S	X09	X10	X11	X12
Y05	I01	I02	I03	I04	Y05	I01	I02	I03	I04	Y05	I01	I02	I03	I04
Y06	I05	I06	I07	I08	Y06	I05	I06	I07	I08	Y06	I05	I06	I07	I08
Y07	I09	I10	I11	I12	Y07	I09	I10	I11	I12	Y07	I09	I10	I11	I12
Y08	I13	I14	I15	I16	Y08	I13	I14	I15	I16	Y08	I13	I14	I15	I16
S	X01	X02	X03	X04	S	X05	X06	X07	X08	S	X09	X10	X11	X12
Y09	I01	I02	I03	I04	Y09	I01	I02	I03	I04	Y09	I01	I02	I03	I04
Y10	I05	I06	I07	I08	Y10	I05	I06	I07	I08	Y10	I05	I06	I07	I08
Y11	I09	I10	I11	I12	Y11	I09	I10	I11	I12	Y11	I09	I10	I11	I12
Y12	I13	I14	I15	I16	Y12	I13	I14	I15	I16	Y12	I13	I14	I15	I16

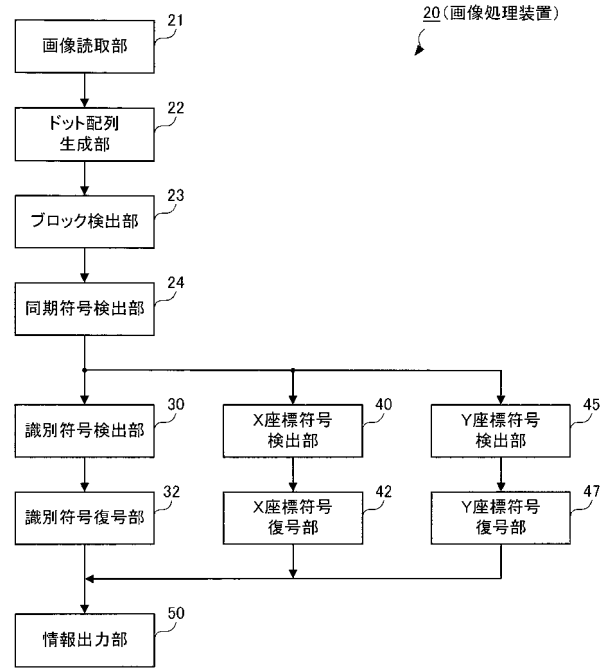
【図 8】



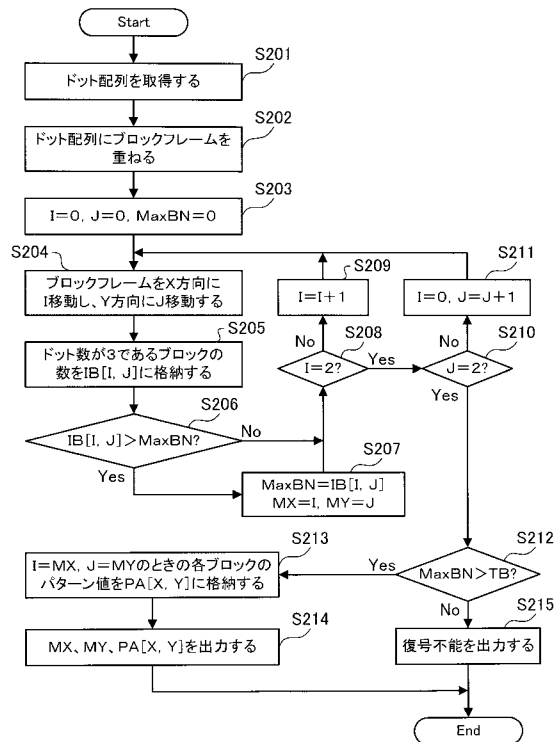
【図 9】



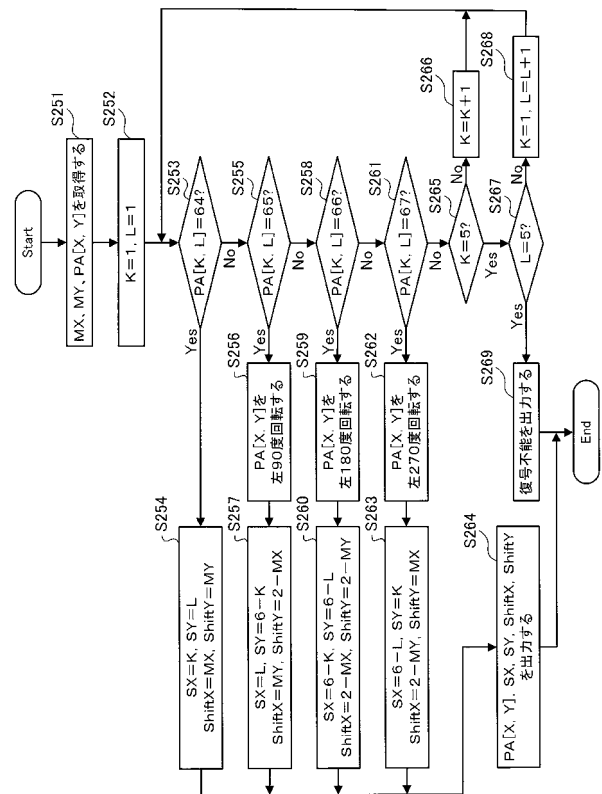
【図 10】



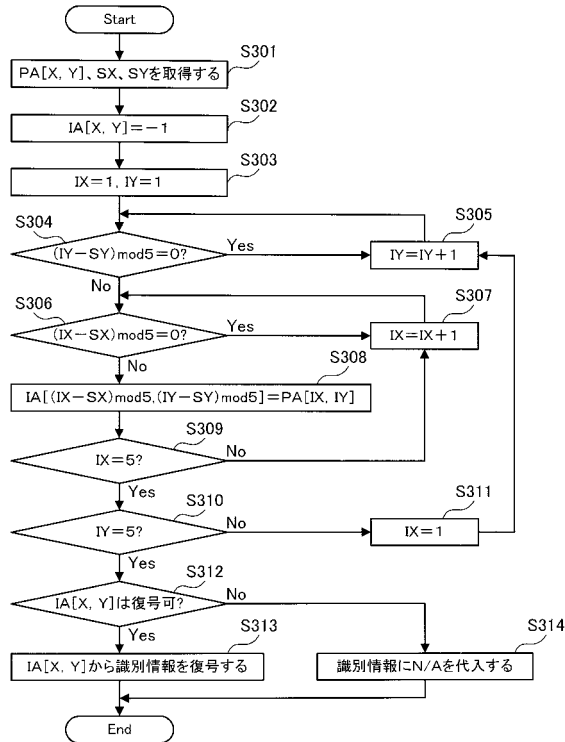
【図 11】



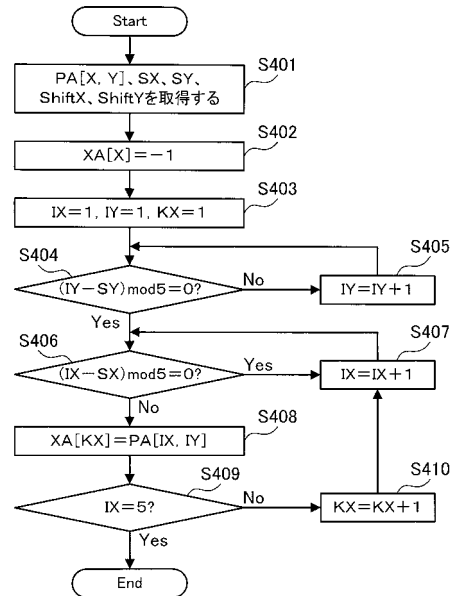
【図 12】



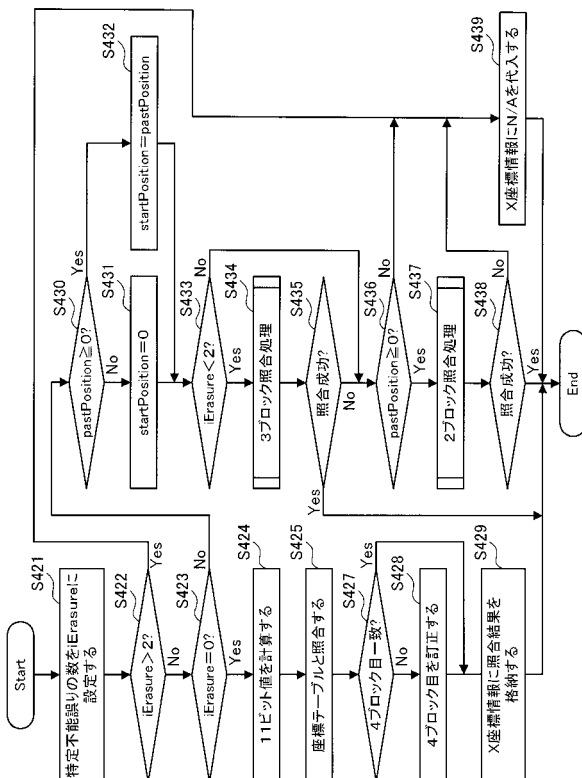
【 図 1 3 】



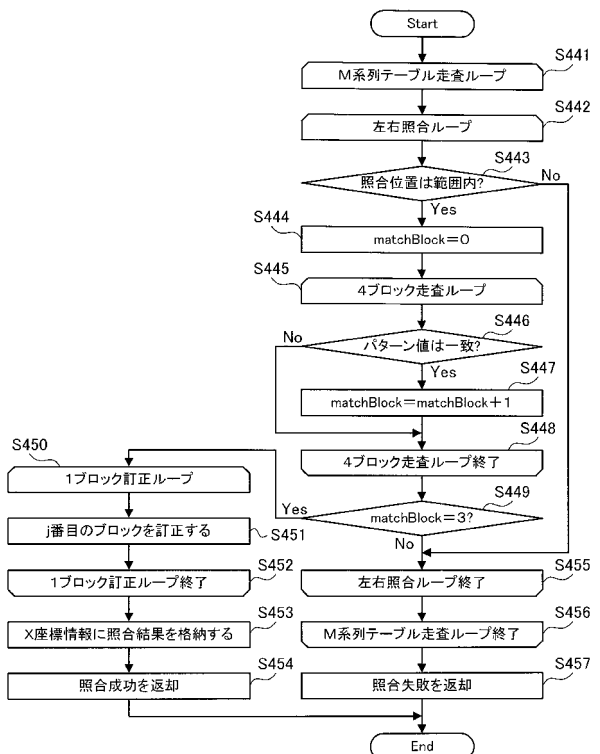
【 図 1 4 】



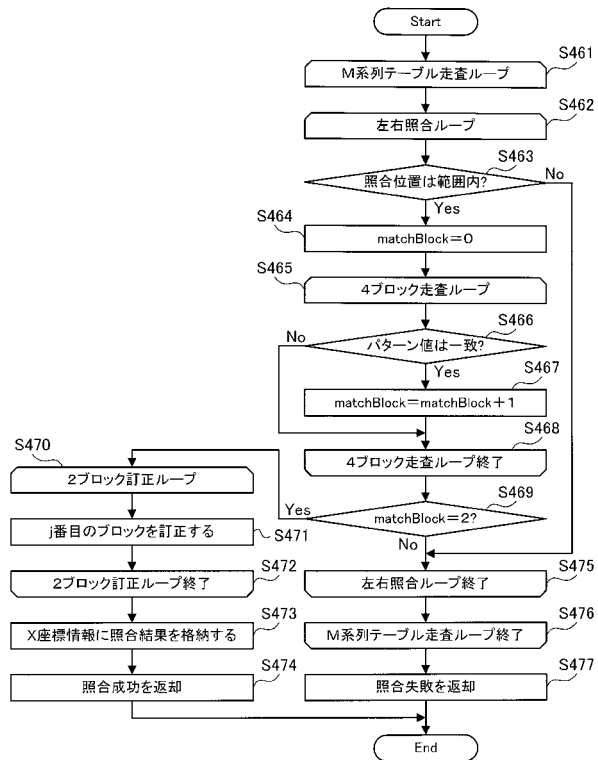
【 図 1 5 】



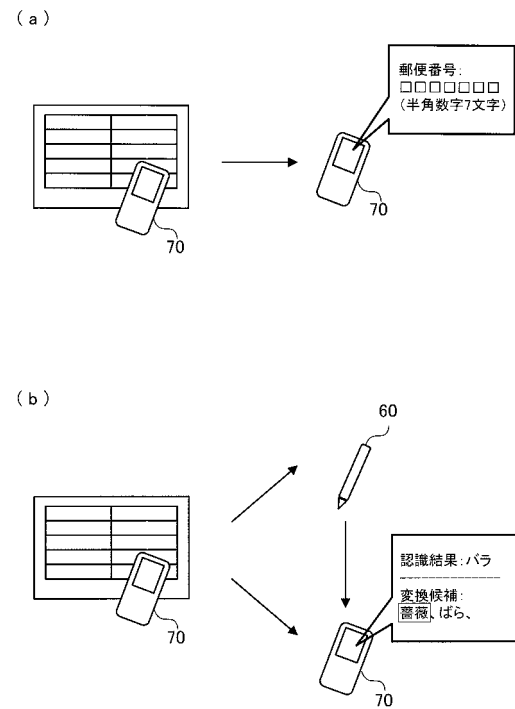
【 図 1 6 】



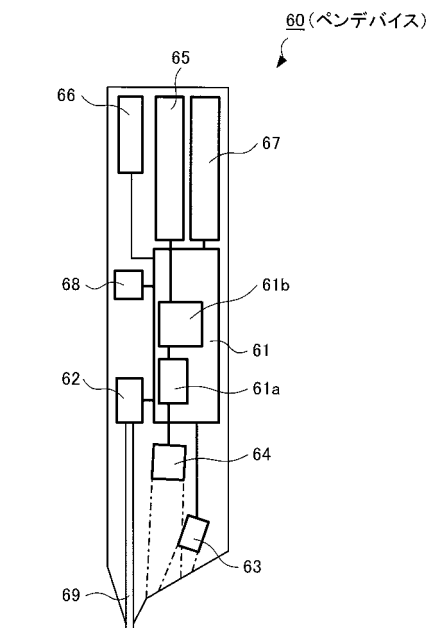
【図 17】



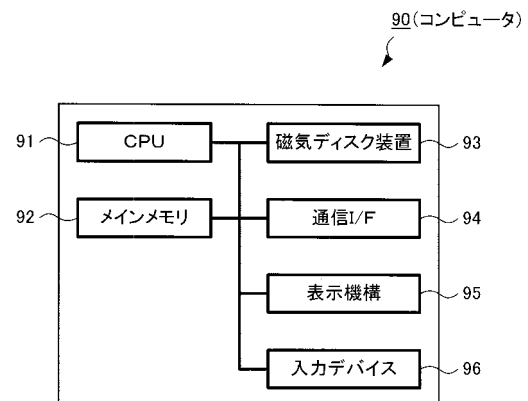
【図 18】



【図 19】



【図 20】



フロントページの続き

(56)参考文献 特開 2 0 0 7 - 1 7 9 1 1 1 (J P , A)
特開 2 0 0 8 - 0 0 9 8 3 3 (J P , A)
特開 2 0 0 7 - 2 8 6 6 9 8 (J P , A)
特開 2 0 0 5 - 0 9 2 4 3 6 (J P , A)
特開 2 0 0 6 - 0 7 9 3 9 1 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G 0 6 F	3 / 0 3		
G 0 6 F	3 / 0 4 1	-	3 / 0 4 7
G 0 6 K	7 / 0 0	-	7 / 1 4
G 0 6 K	1 9 / 0 0	-	1 9 / 0 6
G 0 6 K	1 9 / 0 7	-	1 9 / 1 0