

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号
特許第5800525号
(P5800525)

(45) 発行日 平成27年10月28日 (2015. 10. 28)

(24) 登録日 平成27年9月4日 (2015. 9. 4)

(51) Int.Cl.

G06K 7/14 (2006.01)

F I

G06K 7/14 O6O

請求項の数 14 (全 44 頁)

(21) 出願番号	特願2011-35162 (P2011-35162)	(73) 特許権者	391062872
(22) 出願日	平成23年2月21日 (2011. 2. 21)		株式会社オプトエレクトロニクス
(65) 公開番号	特開2012-173973 (P2012-173973A)		埼玉県蕨市塚越4丁目12番17号
(43) 公開日	平成24年9月10日 (2012. 9. 10)	(74) 代理人	110001209
審査請求日	平成25年12月12日 (2013. 12. 12)		特許業務法人山口国際特許事務所
		(72) 発明者	木村 一人
			埼玉県蕨市塚越4-12-17 株式会社
			オプトエレクトロニクス内
		(72) 発明者	早川 浩
			埼玉県蕨市塚越4-12-17 株式会社
			オプトエレクトロニクス内
		審査官	岡北 有平
最終頁に続く			

(54) 【発明の名称】 光学情報読取装置、光学情報読取方法、コンピュータが読み取り可能なプログラム及び記録媒体

(57) 【特許請求の範囲】

【請求項 1】

1 キャラクタ内で隣接するバー及びスペースを含む n ($n \geq 2$) 本のエレメントの幅を加算して $2 \sim n$ 本加算エレメント幅を各々求め、1 キャラクタ幅に対する前記 $2 \sim n$ 本加算エレメント幅の各々の比率を算出する演算手段と、

前記演算手段によって算出された前記比率から、前記 $2 \sim n$ 本加算エレメント幅の各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出する抽出手段と、

前記抽出手段によって抽出された各々の前記 $2 \sim n$ 本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較する比較手段と、

前記候補キャラクタと最も相関が強い期待値キャラクタを検索する検索手段とを備え、
前記演算手段は、
当該対象キャラクタの端の前記 $2 \sim n$ 本加算エレメント幅を各々求める際に、
前記対象キャラクタの前後のキャラクタのバー及びスペースを含むエレメントの幅を 1 本以上 $n - 1$ 本以下加算範囲を拡張することを特徴とする光学情報読取装置。

【請求項 2】

前記抽出手段は、
当該対象キャラクタの端の前記 $2 \sim n$ 本加算エレメント幅を各々求める際に、
前記 1 キャラクタ分の n 本の幅データと、当該対象キャラクタの前後の 1 本以上 $n - 1$

10

20

本以下の幅データとを抽出することを特徴とする請求項 1 に記載の光学情報読取装置。

【請求項 3】

前記検索手段は、

前記抽出手段から得られた複数の前記対象キャラクタの各々の前記 2 ～ n 本加算エレメント幅に対して 2 種類の前記対象キャラクタの何れかと一致する対象キャラクタを検索し、前記候補キャラクタの数を絞り込むことを特徴とする請求項 1 に記載の光学情報読取装置。

【請求項 4】

前記検索手段は、

前記候補キャラクタのモジュール数の実測値と、前記期待値キャラクタのモジュール数の理想値との間の誤差を算出し、前記誤差を合計して得られる誤差値が最小となる前記候補キャラクタと最も相関が強い期待値キャラクタを検索することを特徴とする請求項 1 に記載の光学情報読取装置。

10

【請求項 5】

前記検索手段は、

前記誤差値が最小となる前記候補キャラクタと一致する前記期待値キャラクタを検出することを特徴とする請求項 4 に記載の光学情報読取装置。

【請求項 6】

前記 1 キャラクタ内で隣接するバー及びスペースを含む $n(n-2)$ 本のエレメントの幅を加算した合計値と、前記 1 キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて当該 1 キャラクタ分の幅データが適切か否かを判別する比較手段を備えることを特徴とする請求項 1 に記載の光学情報読取装置。

20

【請求項 7】

光学情報読取装置が、

1 キャラクタ内で隣接するバー及びスペースを含む $n(n-2)$ 本のエレメントの幅を加算して 2 ～ n 本加算エレメント幅を各々求めるステップと、

求めた前記 2 ～ n 本加算エレメント幅の各々の、1 キャラクタ幅に対する比率を算出するステップと、

算出した前記比率から、前記 2 ～ n 本加算エレメント幅の各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出するステップと、

30

抽出された各々の前記 2 ～ n 本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索するステップとを実行し、

当該対象キャラクタの端の前記 2 ～ n 本加算エレメント幅を各々求める際に、

前記対象キャラクタの前後のキャラクタのバー及びスペースを含むエレメントの幅を 1 本以上 ～ n - 1 本以下加算範囲を拡張することを特徴とする光学情報読取方法。

【請求項 8】

当該対象キャラクタの前記 2 ～ n 本加算エレメント幅を各々求める際に、

前記 1 キャラクタ分の n 本の幅データと、当該対象キャラクタの前後の 1 本以上 n - 1 本以下の幅データとを抽出することを特徴とする請求項 7 に記載の光学情報読取方法。

40

【請求項 9】

抽出された複数の前記対象キャラクタの各々の前記 2 ～ n 本加算エレメント幅に対して 2 種類の前記対象キャラクタの何れかと一致する対象キャラクタを検索し、前記候補キャラクタの数を絞り込むことを特徴とする請求項 8 に記載の光学情報読取方法。

【請求項 10】

前記候補キャラクタのモジュール数の実測値と、前記期待値キャラクタのモジュール数の理想値との間の誤差を算出し、前記誤差を合計して得られる誤差値が最小となる前記候補キャラクタと最も相関が強い期待値キャラクタを抽出することを特徴とする請求項 7 に記載の光学情報読取方法。

【請求項 11】

50

前記誤差値が最小となる前記候補キャラクタと一致する前記期待値キャラクタを検出することを特徴とする請求項 10 に記載の光学情報読取方法。

【請求項 12】

前記 1 キャラクタ内で隣接するバー及びスペースを含む n ($n \geq 2$) 本のエレメントの幅を加算した合計値と、前記 1 キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて当該 1 キャラクタ分の幅データが適切か否かを判別することを特徴とする請求項 7 に記載の光学情報読取方法。

【請求項 13】

請求項 7 乃至 12 に記載の光学情報読取方法を実行するための指令を組み合わせたことを特徴とするコンピュータが読み取り可能なプログラム。

10

【請求項 14】

請求項 7 乃至 12 に記載の光学情報読取方法を実行するためのプログラムが記述されていることを特徴とするコンピュータが読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、バーコード記号を読み取ってバーコードデータを出力するバーコード記号読取装置に適用可能な光学情報読取装置、光学情報読取方法、コンピュータが読み取り可能なプログラム及び記録媒体に関するものである。

【背景技術】

20

【0002】

従来から、物品流通システムにおいて、流通物品にバーコード記号を貼付又は印刷し、この種のバーコード記号をバーコードリーダー（以下光学情報読取装置という）等により読み取って情報処理するシステムが利用されている。光学情報読取装置によれば、バーコード記号 1 を読み取って、そのスタートマージンを検索し、更に、スタートマージンに続くキャラクタをデコード処理してバーコードデータを出力するようになされる。

【0003】

ところで、バーコード記号の規格には、CODE 128 や、CODE 39、EAN/UPC (EAN/Universal Product Code)、インターリーブド 2 オブ 5、コードバー、RSS、RSS リミテッド、RSS エクスパンデド等の各々の方式が有る。一方、バーコード記号は流通物品に対して、バーコード記号標を貼付し、又は、パッケージに直接印刷して使用される場合が多い。しかし、これらの印刷技術や、貼付技術等のバラツキや、流通物品の取り扱い方法によって流通物品に貼付又は印刷されたバーコード記号を正しく読み取れない場合が想定される。バーコード記号を正しく読み取れない原因として、バーコード記号を構成するバーやスペースの欠落や、傷等が挙げられる。

30

【0004】

この種の光学情報読取装置に関連して、特許文献 1 にはバーコード記号読取装置、それによるスタートマージン候補検出方法並びにバーコード記号読取方法が開示されている。このバーコード記号読取装置を応用して、バーコード記号を構成するバーやスペースに不本意に生じた、欠落や傷等を含む難読キャラクタをデコード処理する場合について、図 14 ~ 図 17 を参照してその説明をする。

40

【0005】

図 14 は、従来例に係る光学情報読取装置 20 の構成例を示すブロック図である。図 14 に示す光学情報読取装置 20 は、光学読取部 2、クロック発生部 3、 m 段データバッファ 5、 x 倍器 7、セレクタ部 6、比較器 8、スタートマージン格納部 9、インターフェース部 11、デコードバッファ 25、カウントデータ値格納部 30、CPU 33 及び、バーコードデータ値用のカウンタ部 40 を有して構成される。

【0006】

光学読取部 2 は、読み取り対象であるバーコード記号 1 を読み取って、コード読取データ D2 を発生する。例えば、光学読取部 2 からバーコード記号 1 に対して、発光ダイオー

50

ド(ＬＥＤ)やレーザダイオード(ＬＤ)等の光源による光を照射し、その反射光をフォトダイオードやフォトトランジスタ等の受光素子で受光して電気信号に変換し、その電気信号を２値化してバーコード記号１の濃淡、すなわち、黒部分のバー又は白部分のスペースを示すデジタルのコード読取データＤ２に変換する。

【０００７】

光学読取部２はカウンタ部４０に接続される。カウンタ部４０にはクロック発生部３が接続される。クロック発生部３は、所定周波数のサンプリングクロック信号(以下単にＣＬＫ信号という)を発生して、ＣＬＫ信号をカウンタ部４０に出力する。クロック発生部３には発振器が使用される。

【０００８】

バーコードデータ値用のカウンタ部４０は、光学読取部２から出力されるコード読取データＤ２を入力して、クロック発生部３から出力されるＣＬＫ信号を入力し、当該ＣＬＫ信号に基づいてコード読取データＤ２をカウントし、コード読取データＤ２を白部分のスペースおよび黒部分のバーを含むエレメントの幅を示すカウントデータＤ４(バーコードデータ値：数値)に変換する。

【０００９】

カウンタ部４０にはｍ段データバッファ５、×倍器７及びカウントデータ値格納部３０が接続される。ｍ段データバッファ５はカウンタ部４０から出力されるカウントデータＤ４を、ｍ段分(ｍは任意の整数)だけ一時格納する。ｍ段は、バーコード記号１の印刷状態により任意に設定される。ｍ段データバッファ５はフリップフロップ回路やメモリ等の記憶回路から構成される。

【００１０】

×倍器７は、カウントデータＤ４からスタートマージンを検索するに当たり、カウントデータＤ４を何倍かを示す「×」が予め設定され、その×倍の演算を行って、×倍データＤ７を出力する。×倍器７はカウントデータＤ４を×倍する演算回路によって構成される。ＣＰＵ３３は、バーコード記号１の印刷状態に対応して、カウントデータＤ４を何倍にするかを×倍器７に設定する。

【００１１】

ｍ段データバッファ５にはセクタ部６が接続される。セクタ部６は、ｍ段データバッファ５に一時格納されたｍ段分のカウントデータＤ４のうち、比較対象とするデータを選択する。セクタ部６はＣＰＵ３３等の制御を受けて、ｍ段分のカウントデータＤ４のうち、どのカウントデータＤ４を比較するかを選択するようになされる。

【００１２】

ＣＰＵ３３は、ｍ段データバッファ５中の何段(ｋ段)前までのカウントデータＤ４を比較対象として選択するかを、バーコード記号１の印刷状態、印刷汚れ、外的要因による汚れ等の状態に準じてセクタ部６に設定される。この例では、ｍ段データバッファ５から出力されるカウントデータＤ４が、今、カウンタ部４０から出力されたカウントデータＤ４の値の何倍より大きいことをもってスタートマージン候補とするかが予め設定されている。

【００１３】

上述のセクタ部６及び×倍器７には比較器８が接続される。比較器８はセクタ部６が選択した、ｍ段データバッファ５中のｋ段前までの各カウントデータＤ４と、×倍器７から出力される×倍データＤ７とを比較する。比較結果によって×倍器７からの×倍データＤ７の方が小さいときは、スタートマージン候補があったと判定し、スタートマージン候補検索フラグ(以下でＳＭＦデータＤ９という)を発生する。ＳＭＦデータＤ９はスタートマージン格納部９に出力される。

【００１４】

比較器８にはスタートマージン格納部９が接続される。スタートマージン格納部９にはフリップフロップ回路のような記憶機能を有して構成される。スタートマージン格納部９には比較器８から出力されるＳＭＦデータＤ９、そのとき比較対象としたカウントデータ

10

20

30

40

50

値 = バーコードデータ値 D 3 0 とが対応させて格納される。どのカウントデータ D 4 によってスタートマージン候補となったかが判るようにするためである。スタートマージン候補は、バーコード記号 1 におけるバーの手前に、ある一定以上の幅の白バーが存在しているときに発生される。

【 0 0 1 5 】

カウントデータ値格納部 3 0 は、カウンタ部 4 0 から順次出力されるカウントデータ D 4 を、時系列的にすべて格納して保存する。カウントデータ値格納部 3 0 は、フリップフロップ回路や、D R A M 等のような記憶回路から構成される。スタートマージン格納部 9 及びカウントデータ値格納部 3 0 にはインターフェース部 1 1 が接続される。

【 0 0 1 6 】

インターフェース部 1 1 は、スタートマージン格納部 9 から読み出される S M F データ D 9 や、カウントデータ値格納部 3 0 から読み出した各キャラクタのバーコードデータ値 D 3 0 を C P U 3 3 に入力する。インターフェース部 1 1 にはデコードバッファ 2 5 が接続され、C P U 3 3 から出力されるバーコードデータ D 1 2 が格納される。

【 0 0 1 7 】

デコードバッファ 2 5 はカウントデータ値格納部 3 0 から出力された各キャラクタのバーコードデータ値 D 3 0 の演算途中のデータを格納もなされる。また、インターフェース部 1 1 には出力端子 OUT が接続され、出力端子 OUT にはデコード処理後のバーコードデータ D 1 2 が出力される。バーコードデータ D 1 2 は、例えば、デコードバッファ 2 5 からインターフェース部 1 1 を介して出力端子 OUT へ読み出される。

【 0 0 1 8 】

図 1 5 A ~ E は、従来例に係る 2 本加算データの算出例を示す説明図である。図 1 5 A に示す C L K 信号は図 1 4 に示したクロック発生部 3 からカウンタ部 4 0 へ出力される。図 1 5 B に示すバーコード記号 1 は、例えば、C O D E 1 2 8 規格のものである。C O D E 1 2 8 規格によれば、図 1 5 B に示す 1 キャラクタについて、3 本の黒部分のバー (1) , (3) , (5) 及び 3 本の白部分のスペース (2) , (4) , (6) から構成される。1 キャラクタの全幅は 1 1 モジュールから構成される。当該キャラクタ C 1 の黒部分のバー (1) は 4 モジュールから成り、バー (3) 及びバー (5) は各々 1 モジュールから成る。白部分のスペース (2) は 1 モジュールから成り、スペース (4) 及び (6) は各々 2 モジュールから成る。

【 0 0 1 9 】

図 1 5 C に示すバーコードデータ値 D 3 0 は、図 1 4 に示した光学読取部 2 から出力されるコード読取データ D 2 を、クロック発生部 3 から出力される C L K 信号に基づいてサンプリングし、当該 C L K 信号に基づいてコード読取データ D 2 をカウントし、コード読取データ D 2 を白部分のスペースおよび黒部分のバーを含むエレメントの幅を示すカウントデータ D 4 に変換することで得られる。

【 0 0 2 0 】

図 1 5 D に示す W 1 1 はキャラクタ C 1 の黒部分のバー (1) の幅である。W 1 2 はバー (1) に続くスペース (2) の幅である。W 1 3 はキャラクタ C 1 のスペース (2) に続くバー (3) の幅である。W 1 4 はキャラクタ C 1 のバー (3) に続くスペース (4) の幅である。W 1 5 はスペース (4) に続くバー (5) の幅である。W 1 6 はキャラクタ C 1 のバー (5) に続くスペース (6) の幅である。上述の 6 種類の幅を示すデータは 1 キャラクタのバーコードデータ値 D 3 0 を構成する。

【 0 0 2 1 】

図 1 5 E に示す W 2 1 はキャラクタ C 1 のバー (1) とスペース (2) の各々が C P U 3 3 によって加算された幅 (以下バー (1) + スペース (2) と記す) である。バー (1) + スペース (2) の加算幅を示すカウントデータ値は幅 W 2 1 データを成す。W 2 2 はキャラクタ C 1 のバー (1) に続くスペース (2) 及びバー (3) の各々が C P U 3 3 によって加算された幅である。スペース (2) + バー (3) の加算幅を示すカウントデータ値は幅 W 2 2 データを成す。

【 0 0 2 2 】

W 2 3 はキャラクタ C 1 のスペース (2) に続くバー (3) 及びスペース (4) の各々が C P U 3 3 によって加算された幅である。バー (3) + スペース (4) の加算幅を示すカウントデータ値は幅 W 2 3 データを成す。W 2 4 はキャラクタ C 1 のバー (3) に続くスペース (4) 及びバー (5) の各々が C P U 3 3 によって加算された幅である。スペース (4) + バー (5) の加算幅を示すカウントデータ値は幅 W 2 4 データを成す。W 2 5 はキャラクタ C 1 のスペース (4) に続くバー (5) 及びスペース (6) の各々が C P U 3 3 によって加算された幅である。バー (5) + スペース (6) の加算幅を示すカウントデータ値は幅 W 2 5 データを成す。上述の 5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データは 1

10

【 0 0 2 3 】

図 1 6 及び図 1 7 は、従来例に係る難読キャラクタのデコード例 (その 1 , 2) を示すフローチャートである。光学情報読取装置 2 0 では、C O D E 1 2 8 規格に基づくバーコード記号 1 を読み取って、そのスタートマージンを検索し、更に、スタートマージンに続くキャラクタを 2 本加算方式によりデコード処理してバーコードデータ D 1 2 を出力する場合を例に挙げる。その場合のバーコード記号 1 のバー及びスペースの各エレメントの誤差許容度 (最大誤差範囲) が、例えば、 ± 0.5 モジュールに設定される。

【 0 0 2 4 】

これらをデコード条件にして、図 1 6 に示すフローチャートのステップ S T 81 で C P U 3 3 はスタートマージンを検索する。このとき、光学読取部 2 は、バーコード記号 1 を読み取って、コード読取データ D 2 を発生する。コード読取データ D 2 は、光学読取部 2 からカウンタ部 4 0 へ出力される。クロック発生部 3 は、所定周波数のサンプリング用の C L K 信号をカウンタ部 4 0 に出力する。

20

【 0 0 2 5 】

カウンタ部 4 0 は光学読取部 2 から出力されるコード読取データ D 2 を入力すると共に、クロック発生部 3 から出力される C L K 信号を入力し、C L K 信号に基づいてコード読取データ D 2 をカウントし、コード読取データ D 2 を白部分のスペースおよび黒部分のバーを含むエレメントの 2 本加算幅を示すカウントデータ D 4 に変換する。カウントデータ D 4 は、バーコード記号 1 における各バー (黒部分) の幅およびスペース (白部分) の幅を示す数値データ列となる。カウントデータ D 4 は、カウントデータ値格納部 3 0 に保存される。

30

【 0 0 2 6 】

m 段データバッファ 5 はカウンタ部 4 0 から出力されるカウントデータ D 4 を、m 段分 (m は任意の整数) だけ一時格納する。m 段は、バーコード記号 1 の印刷状態により任意に設定されている。x 倍器 7 は、カウントデータ D 4 からスタートマージンを検索するに当たり、カウントデータ D 4 を何倍かを示す「x」が予め設定され、その x 倍の演算を行って、x 倍データ D 7 を出力する。

【 0 0 2 7 】

セレクト部 6 は、C P U 3 3 の選択制御を受けて、m 段データバッファ 5 に一時格納された m 段分のカウントデータ D 4 のうち、比較対象とするデータを選択する。比較器 8 はセレクト部 6 が選択した、m 段データバッファ 5 中の k 段前までの各カウントデータ D 4 と、x 倍器 7 から出力される x 倍データ D 7 とを比較する。

40

【 0 0 2 8 】

比較結果によって x 倍器 7 からの x 倍データ D 7 の方が小さいときは、スタートマージン候補があったと判定し、スタートマージン候補検索フラグ (以下で S M F データ D 9 という) を発生する。S M F データ D 9 はスタートマージン格納部 9 に出力される。スタートマージン格納部 9 には比較器 8 から出力される S M F データ D 9、そのとき比較対象としたカウントデータ値 = D 1 0 とが対応させて格納される。

【 0 0 2 9 】

ステップ S T 82 で C P U 3 3 は 1 キャラクタ分の幅データ (6 本) を抽出する。図 1 5

50

に示したキャラクタC 1によれば、黒部分のバー(1)の幅W 1 1がカウンタ部4 0により計数され、そのカウントデータ値が幅W 1 1データとなる。バー(1)に続く白部分のスペース(2)の幅W 1 2もカウンタ部4 0により計数され、そのカウントデータ値が幅W 1 2データとなる。スペース(2)に続くバー(3)の幅W 1 3はカウンタ部4 0により計数され、そのカウントデータ値が幅W 1 3データとなる。

【0030】

このバー(3)に続くスペース(4)の幅W 1 4もカウンタ部4 0により計数され、そのカウントデータ値は幅W 1 4データとなる。スペース(4)に続くバー(5)の幅W 1 5はカウンタ部4 0により計数され、そのカウントデータ値が幅W 1 5データとなる。バー(5)に続くスペース(6)の幅W 1 6もカウンタ部4 0により計数され、そのカウントデータ値が幅W 1 6データとなる。上述の6種類の幅W 1 1データ～幅W 1 6データはバーコードデータ値D 3 0を構成する。バーコードデータ値D 3 0はインターフェース部1 1を介してCPU 2 3に出力される。

10

【0031】

ステップS T83でCPU 3 3はキャラクタC 1の1モジュール幅を算出する。この例では、CPU 3 3は、1キャラクタ分の6本の幅W 1 1データ～幅W 1 6データのバーコードデータ値D 3 0をキャラクタのモジュール数に換算して2本加算パターンのモジュール数を求めている。図1 5 Eに示したように、カウンタ部4 0のカウントデータ値から、5種類の幅W 2 1データ～幅W 2 5データがCPU 3 3に順次出力される。

【0032】

20

例えば、CPU 3 3は幅W 2 1データを入力し、キャラクタC 1のバー(1)とスペース(2)とを加算した加算幅W 2 1のモジュール数を求める。モジュール数は全キャラクタ幅=1 1モジュールに対する加算幅W 2 1の比率、例えば、2 . 4を算出する。この場合、加算幅W 2 1のモジュール数は、バーやスペース等の各エレメントの最大誤差が0 . 5以内となるように、当該モジュール数を四捨五入して最も近い整数になされる。例えば、W 2 1のモジュール数が2 . 4の場合、W 2 1のモジュール数は「2」である。モジュール数が2 . 6の場合、W 2 1のモジュール数は「3」である。

【0033】

ステップS T84でCPU 3 3は1キャラクタ分のバーコードデータ値D 3 0を2本加算パターンに変換する。例えば、図1 5 Eに示したキャラクタC 1のバー(1)とスペース(2)の各々がCPU 3 3によって加算された加算幅W 2 1を示すカウントデータ値が幅W 2 1データとなる。キャラクタC 1のバー(1)に続くスペース(2)及びバー(3)の各々がCPU 3 3によって加算された加算幅W 2 2を示すカウントデータ値が幅W 2 2データとなる。

30

【0034】

キャラクタC 1のスペース(2)に続くバー(3)及びスペース(4)の各々がCPU 3 3によって加算された加算幅W 2 3を示すカウントデータ値が幅W 2 3データとなる。キャラクタC 1のバー(3)に続くスペース(4)及びバー(5)の各々がCPU 3 3によって加算された加算幅W 2 4を示すカウントデータ値が幅W 2 4データとなる。キャラクタC 1のスペース(4)に続くバー(5)及びスペース(6)の各々がCPU 3 3によって加算された加算幅W 2 5を示すカウントデータ値が幅W 2 5データとなる。

40

【0035】

その後、ステップS T85でCPU 3 3は、先に変換した2本加算パターンが一致するキャラクタを検索する。この例では、図1 4に示したカウンタ部4 0から出力されるバーコードデータ値D 3 0に基づいて、CPU 3 3が5種類の幅W 2 1データ～幅W 2 5データ(2本加算データ)に関して、先に変換されたモジュール値と、期待値キャラクタの理論値とが一致するキャラクタを検索する。従来方式では、この段階でキャラクタが決定する。

【0036】

ステップS T86でCPU 3 3は一致するキャラクタの存在有無に基づいて制御を分岐す

50

る。一致するキャラクタが存在する場合は、ステップ S T 87 で 2 本加算パターンが一致したキャラクタをデコード結果とする。その後、図 1 7 に示すステップ S T 88 に移行する。

【 0 0 3 7 】

上述のステップ S T 87 でデコード結果となされたキャラクタについては、ステップ S T 88 で、C P U 3 3 はマージン直後のキャラクタか否かに基づいて制御を分岐する。デコード結果となされたキャラクタが、マージン直後のキャラクタではない場合は、ステップ S T 89 に移行して C P U 3 3 はデコード結果がストップキャラクタか否かに基づいて制御を分岐する。

【 0 0 3 8 】

デコード結果がストップキャラクタではない場合は、ステップ S T 90 に移行して C P U 3 3 は、誤り検知 (パリティチェック) を実行して、チェックデジットが正しいか否かに基づいて制御を分岐する。チェックデジットが正しい場合は、ステップ S T 91 で C P U 3 3 は読み取りを完了し、デコードバッファ 2 5 のバーコードデータ D 1 2 を出力し、デコードバッファ 2 5 をクリアする (正常終了) 。その後、ステップ S T 95 に移行する。

【 0 0 3 9 】

なお、ステップ S T 88 でマージン直後のキャラクタである場合は、ステップ S T 92 に移行して C P U 3 3 は、デコード結果がスタートキャラクタか否かに基づいて制御を分岐する。上述のステップ S T 89 でデコード結果がストップキャラクタである場合及び、ステップ S T 92 でデコード結果がスタートキャラクタである場合は、図 1 6 に示すステップ S T 83 に移行する。ステップ S T 83 で C P U 3 3 は、次キャラクタ (6 本後ろのバーコードデータ値 D 3 0) のデコード処理を実行する。その後、ステップ S T 81 に戻る。

【 0 0 4 0 】

また、図 1 6 に示したステップ S T 86 で一致するキャラクタが存在しない場合、図 1 7 に示したステップ S T 92 でデコード結果がスタートキャラクタではない場合及び、ステップ S T 90 でチェックデジットが誤っている場合は、ステップ S T 94 に移行して C P U 3 3 は、読み取りエラー処理を実行する。読み取りエラー処理では次のマージンを検索する等が行われる。

【 0 0 4 1 】

その後、ステップ S T 95 に移行して、C P U 3 3 は終了を判別する。例えば、当該光学情報読取装置 2 0 に対して読取終了指令が検出され、読取終了指令が検出された場合は、バーコード記号 1 の読み取り制御を終了する。読取終了指令が検出されていない場合は、バーコード記号 1 の読み取り制御を継続するべく、ステップ S T 81 に戻って、上述した内容を繰り返すようになされる。

【 先行技術文献 】

【 特許文献 】

【 0 0 4 2 】

【 特許文献 1 】 特許 4 5 5 5 9 5 2 号公報 (第 4 頁 図 1)

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 4 3 】

ところで、従来例に係る光学情報読取装置 2 0 によれば、特許文献 1 に見られるようなバーコード記号読取装置の構成に、1 キャラクタに関して、2 本幅加算方式を導入して難読キャラクタをデコード可能な光学情報読取装置を作成しようとした場合、次のような問題がある。

i . 従来方式の光学情報読取装置 2 0 では、キャラクタパターン算出時、バー及びスペースを含む各エレメントの誤差許容度が ± 0.5 モジュール以内に設定され、1 キャラクタ分の 6 本の幅 W 1 1 データ ~ 幅 W 1 6 データのバーコードデータ値 D 3 0 がキャラクタのモジュール数に換算されて、2 本加算パターンのモジュール数が求められている。しかも、バーやスペース等の各エレメントの最大誤差が 0.5 以内となるように、2 本加算パターンのモジュール数が四捨五入して最も近い整数になされている。

10

20

30

40

50

【 0 0 4 4 】

従って、加算幅 W 2 1 のキャラクタのモジュール数の計数値が 2 . 4 の場合、本来、整数の候補として「 2 」と「 3 」となるにも関わらず、モジュール数が四捨五入されて最も近い整数として、モジュール数が「 2 」となされてしまう（ステップ S T 83 参照）。

【 0 0 4 5 】

因みに、期待値キャラクタとして、モジュール数が「 3 」であったような場合であって、印刷技術や貼付技術等によって、バーコード記号を構成するバーやスペースに、欠落や傷等が介在して、モジュール数の計数値が 2 . 4 となった場合に、 $\pm 1 . 0$ の誤差が介入する結果となって、バーコード記号を正確に読み取れないという問題がある。誤差が $\pm 0 . 5$ モジュールを越えるエレメントが存在した場合は、読み取りエラーとなっていた（ステップ S T 94 参照）。

10

【 0 0 4 6 】

ii . また、キャラクタパターン検索時、5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データ（2 本加算データ）に対して一致するキャラクタを検索しているが、誤差が $\pm 0 . 5$ モジュールを越えるエレメントを含むキャラクタは検索対象から除外されてしまう。従って、ステップ S T 83 で四捨五入によって誤差を含んだキャラクタを含むキャラクタがデコード結果と決定される場合が想定される。この結果、従来方式のバーコード読み取り方法によると、バーやスペース等のエレメントが傷やかすれた場合に、バーコードを誤読してしまい、バーコードデータを正確に出力することが困難となるという問題がある。

【 0 0 4 7 】

20

そこで、本発明は上述した課題を解決したものであって、従来方式のバーコード読み取り方法に比べてバーコードの誤読取を防止できるようにすると共に、エレメントが傷やかすれた場合であっても、正確なバーコードデータを出力できるようにした光学情報読取装置、光学情報読取方法、コンピュータが読み取り可能なプログラム及び記録媒体を提供することを目的とする。

【課題を解決するための手段】

【 0 0 4 8 】

上述の課題を解決するために、請求項 1 に記載の光学情報読取装置は、1 キャラクタ内で隣接するバー及びスペースを含む n ($n \geq 2$) 本のエレメントの幅を加算して 2 ~ n 本加算エレメント幅を各々求め、1 キャラクタ幅に対する前記 2 ~ n 本加算エレメント幅の各々の比率を算出する演算手段と、前記演算手段によって算出された前記比率から、前記 2 ~ n 本加算エレメント幅の各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出する抽出手段と、前記抽出手段によって抽出された各々の前記 2 ~ n 本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較する比較手段と、前記候補キャラクタと最も相関が強い期待値キャラクタを検索する検索手段とを備え、前記演算手段は、当該対象キャラクタの端の前記 2 ~ n 本加算エレメント幅を各々求める際に、前記対象キャラクタの前後のキャラクタのバー及びスペースを含むエレメントの幅を 1 本以上 $n - 1$ 本以下加算範囲を拡張することを特徴とするものである。

30

【 0 0 4 9 】

40

請求項 1 に係る光学情報読取装置によれば、バーコード記号を読み取ってそのデコード結果を出力する場合に、演算手段は、1 キャラクタ内で隣接するバー及びスペースを含む n ($n \geq 2$) 本のエレメントの幅を加算して 2 ~ n 本加算エレメント幅を各々求め、この 2 ~ n 本加算エレメント幅の各々を 1 キャラクタのモジュール数に換算して当該 2 ~ n 本加算エレメント幅のモジュール数を求める。抽出手段は、演算手段によって求められた 2 ~ n 本加算エレメント幅の各々のモジュール数が誤差 1 以内の整数となるキャラクタを候補キャラクタとして抽出する。これらを前提にして、検索手段が抽出手段によって抽出された各々の 2 ~ n 本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索するようになる。

50

【 0 0 5 0 】

この検索によって、複数の候補キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータを当該バーコード記号の読取時のデコード結果として出力できるようになるので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

【 0 0 5 2 】

請求項2に記載の光学情報読取装置は、請求項1において、前記抽出手段は、当該対象キャラクタの端の前記2～n本加算エレメント幅を各々求める際に、前記1キャラクタ分のn本の幅データと、当該対象キャラクタの前後の1本以上n-1本以下の幅データとを抽出することを特徴とするものである。

10

【 0 0 5 3 】

請求項3に記載の光学情報読取装置は、請求項1において、前記検索手段は、前記抽出手段から得られた複数の前記対象キャラクタの各々の前記2～n本加算エレメント幅に対して2種類の前記対象キャラクタの何れかと一致する対象キャラクタを検索し、前記候補キャラクタの数を絞り込むことを特徴とするものである。

【 0 0 5 4 】

請求項4に記載の光学情報読取装置は、請求項1において、前記検索手段は、前記候補キャラクタのモジュール数の実測値と、前記期待値キャラクタのモジュール数の理想値との間の誤差を算出し、前記誤差を合計して得られる誤差値が最小となる前記候補キャラクタと最も相関が強い期待値キャラクタを検索することを特徴とするものである。

20

【 0 0 5 5 】

請求項5に記載の光学情報読取装置は、請求項4において、前記検索手段は、前記誤差値が最小となる前記候補キャラクタと一致する前記期待値キャラクタを検出することを特徴とするものである。

【 0 0 5 6 】

請求項6に記載の光学情報読取装置は、請求項1において、前記1キャラクタ内で隣接するバー及びスペースを含む $n(n \geq 2)$ 本のエレメントの幅を加算した合計値と、前記1キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて当該1キャラクタ分の幅データが適切か否かを判別する比較手段を備えることを特徴とするものである。

【 0 0 5 7 】

30

請求項7に記載の光学情報読取方法は、光学情報読取装置が、1キャラクタ内で隣接するバー及びスペースを含む $n(n \geq 2)$ 本のエレメントの幅を加算して2～n本加算エレメント幅を各々求めるステップと、求めた前記2～n本加算エレメント幅の各々の、1キャラクタ幅に対する比率を算出するステップと、算出した前記比率から、前記2～n本加算エレメント幅の各々のモジュール数が誤差1以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出するステップと、抽出された各々の前記2～n本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索するステップとを実行し、当該対象キャラクタの端の前記2～n本加算エレメント幅を各々求める際に、前記対象キャラクタの前後のキャラクタのバー及びスペースを含むエレメントの幅を1本以上n-1本以下加算することを特徴とするものである。

40

【 0 0 5 9 】

請求項8に記載の光学情報読取方法は、請求項7において、前記2～n本加算エレメント幅を各々求める際に、前記1キャラクタ分のn本の幅データと、当該キャラクタの前後の1本以上n-1本以下の幅データとを抽出することを特徴とするものである。

【 0 0 6 0 】

請求項9に記載の光学情報読取方法は、請求項8において、抽出された複数の前記対象キャラクタの各々の前記2～n本加算エレメント幅に対して2種類の前記対象キャラクタの何れかと一致する対象キャラクタを検索し、前記候補キャラクタの数を絞り込むことを特徴とするものである。

50

【 0 0 6 1 】

請求項 1 0 に記載の光学情報読取方法は、請求項 7 において、前記候補キャラクタのモジュール数の実測値と、前記期待値キャラクタのモジュール数の理想値との間の誤差を算出し、前記誤差を合計して得られる誤差値が最小となる前記候補キャラクタと最も相関が強い期待値キャラクタを抽出することを特徴とするものである。

【 0 0 6 2 】

請求項 1 1 に記載の光学情報読取方法は、請求項 1 0 において、前記誤差値が最小となる前記候補キャラクタと一致する前記期待値キャラクタを検出することを特徴とするものである。

【 0 0 6 3 】

請求項 1 2 に記載の光学情報読取方法は、請求項 7 において、前記 1 キャラクタ内で隣接するバー及びスペースを含む n ($n \geq 2$) 本のエレメントの幅を加算した合計値と、前記 1 キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて当該 1 キャラクタ分の幅データが適切か否かを判別することを特徴とするものである。

【 0 0 6 4 】

請求項 1 3 に記載のコンピュータが読み取り可能なプログラムは、請求項 7 乃至 1 2 に記載の光学情報読取方法を実行するための指令を組み合わせたことを特徴とするものである。

【 0 0 6 5 】

請求項 1 4 に記載のコンピュータが読み取り可能な記録媒体は、請求項 7 乃至 1 2 に記載の光学情報読取方法を実行するためのプログラムが記述されていることを特徴とするものである。

【 発明の効果 】

【 0 0 6 6 】

請求項 1 に係る光学情報読取装置及び、請求項 7 に係る光学情報読取方法によれば、 $2 \sim n$ 本加算エレメント幅の各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んで候補キャラクタが抽出され、ここに抽出された各々の $2 \sim n$ 本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索する検索手段を備えるものである。

【 0 0 6 7 】

この構成によって、複数の候補キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータを当該バーコード記号の読取時のデコード結果として出力できるようになるので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

【 0 0 6 8 】

更に、当該対象キャラクタの $2 \sim n$ 本加算エレメント幅を各々求める際に、対象キャラクタの前後のキャラクタのバー及びスペースを含むエレメントの幅を加算するので、複数の候補キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータを当該バーコード記号の読取時のデコード結果として出力できるようになる。

【 0 0 6 9 】

請求項 2 に記載の光学情報読取装置及び、請求項 8 に記載の光学情報読取方法によれば、当該対象キャラクタの前記 $2 \sim n$ 本加算エレメント幅を各々求める際に、前記 1 キャラクタ分の n 本の幅データと、当該対象キャラクタの前後の 1 本以上 $n - 1$ 本以下の幅データとを抽出するので、複数の候補キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータを当該バーコード記号の読取時のデコード結果として出力できるようになる。

【 0 0 7 0 】

また、請求項 2 に係る光学情報読取装置及び請求項 8 に係る光学情報読取方法によれば

10

20

30

40

50

、1キャラクタ分のn本の幅データと、当該キャラクタの前後の1本以上n-1本以下の幅データとを抽出するので、2～n本加算エレメント幅の各々のモジュール数が誤差1以内の整数となる候補キャラクタを従来方法に比べてより多く抽出できるようになり、第1番目に相関が強いものをデコード結果とするだけでなく、第2番目、第3番目・・・に相関が強いものがデコード結果となる確率が高くなるというようにデコード結果の候補キャラクタを複数出力できるようになる。

【0071】

請求項3に記載の光学情報読取装置及び、請求項9に記載の光学情報読取方法によれば、抽出された複数の前記対象キャラクタの各々の前記2～n本加算エレメント幅に対して2種類の前記対象キャラクタの何れかと一致する対象キャラクタを検索し、候補キャラクタの数を絞り込むので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

10

【0072】

請求項4に記載の光学情報読取装置及び、請求項10に記載の光学情報読取方法によれば、誤差値が最小となる候補キャラクタと最も相関が強い期待値キャラクタを検索するので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

【0073】

請求項5に記載の光学情報読取装置及び、請求項11に記載の光学情報読取方法によれば、誤差値が最小となる候補キャラクタと一致する期待値キャラクタを検出するので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

20

【0074】

請求項6に記載の光学情報読取装置及び、請求項12に記載の光学情報読取方法によれば、当該1キャラクタ分の幅データが適切か否かの判別結果に基づいてキャラクタのバーの消失や分割等を判断できるようになる。複数のスキヤンのデコード結果を組み合わせることが可能となり、汚れや、かすれがひどいバーコードにおいても読み取りが可能となる。

【0075】

請求項13に係るコンピュータが読み取り可能なプログラム及び請求項14に係る記録媒体によれば、本発明に係る光学情報読取方法を実行するための指令を組み合わせただけで、複数の候補キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータを当該バーコード記号の読取時のデコード結果として出力できるようになる。従って、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

30

【図面の簡単な説明】

【0076】

【図1】本発明に係る実施形態としてのバーコード記号読取装置100の構成例を示すブロック図である。

【図2】(A)～(G)は第1の実施例に係る1本データ、2～6本加算データの算出例を示す説明図である。

40

【図3】第1の実施例に係る難読キャラクタのデコード例(その1)を示すフローチャートである。

【図4】難読キャラクタのデコード例(その2)を示すフローチャートである。

【図5】(A)～(G)は、第1の実施例に係る1本データ、2本～6本加算データの他の算出例(その1)を示す説明図である。

【図6】(A)～(G)は、第1の実施例に係る1本データ、2本～6本加算データの他の算出例(その2)を示す説明図である。

【図7】(A)～(C)は、CODE 128のバーコード記号例、その欠損例及びその再構成例を示す説明図である。

50

【図 8】第 2 の実施例に係る難読キャラクタのデコード例（その 1）を示すフローチャートである。

【図 9】難読キャラクタのデコード例（その 2）を示すフローチャートである。

【図 10】難読キャラクタのデコード例（その 3）を示すフローチャートである。

【図 11】第 3 の実施例に係る難読キャラクタのデコード例（その 1）を示すフローチャートである。

【図 12】難読キャラクタのデコード例（その 2）を示すフローチャートである。

【図 13】難読キャラクタのデコード例（その 3）を示すフローチャートである。

【図 14】従来例に係る光学情報読取装置 20 の構成例を示すブロック図である。

【図 15】（A）～（D）は、2 本加算データの算出例を示す説明図である。

【図 16】難読キャラクタのデコード例（その 1）を示すフローチャートである。

【図 17】難読キャラクタのデコード例（その 2）を示すフローチャートである。

【発明を実施するための形態】

【0077】

以下、図面を参照しながら、本発明に係る光学情報読取装置、光学情報読取方法、コンピュータが読み取り可能なプログラム及び記録媒体について説明する。図 1 に示すバーコード記号読取装置 100 は光学情報読取装置の一例を構成し、バーコード記号 1 を読み取って、そのスタートマージンを検索し、更に、スタートマージンに続くキャラクタをデコード処理してバーコードデータ D12 を出力するものである。

【0078】

バーコード記号読取装置 100 は、光学読取部 2、クロック発生部 3、カウンタ部 4、m 段データバッファ 5、x 倍器 7、セクタ部 6、比較器 8、スタートマージン格納部 9、幅データ格納部 10、インターフェース部 11、マイクロプロセッサ 12 及びデコードバッファ 25 を有して構成される。なお、従来例と同じ名称及び同じ符号のものは同じ機能を有するのでその説明を省略する。

【0079】

カウンタ部 4 は光学読取部 2 から出力されるコード読取データ D2 を入力すると共に、クロック発生部 3 から出力される CLK 信号を入力し、当該 CLK 信号に基づいてコード読取データ D2 をカウントし、コード読取データ D2 を白部分のスペースおよび黒部分のバーを含むエレメントの 1 本幅や、n 本加算幅を示すカウントデータ D4（バーコードデータ値：数値）に変換する。

【0080】

カウンタ部 4 には幅データ格納部 10 が接続される。幅データ格納部 10 は、カウンタ部 4 から順次出力される 1 本幅、n 本加算幅の対象キャラクタに係るカウントデータ D4 を、時系列的にすべて格納して保存する。幅データ格納部 10 は、フリップフロップ回路や、DRAM 等のような記憶回路から構成される。幅データ格納部 10 は、1 本幅、n 本加算幅の対象キャラクタに係るカウントデータ D4 を各キャラクタの幅データ D10（バーコードデータ値 D30 に対応）として出力する。スタートマージン格納部 9 及び幅データ格納部 10 にはインターフェース部 11 が接続される。

【0081】

マイクロプロセッサ 12 は、インターフェース部 11 を介して入力される SMF データ D9 や、各キャラクタの幅データ D10 によって、バーコードの認識処理を実行する。マイクロプロセッサ 12 は、プログラム格納用の読み出し専用メモリ（Read Only Memory：以下 ROM 21 という）、ワークメモリ用の随時情報の書き込み読み出し可能なメモリ（Random Access Memory：以下 RAM 22 という）、及び中央処理ユニット（Central Processing Unit；以下 CPU 23 という）及び期待値格納部 24 を含むマイクロコンピュータを構成している。

【0082】

ROM 21 は記録媒体の一例を構成する。ROM 21 にはコンピュータが読み取り可能なプログラムであって、本発明に係る光学情報読取方法を実行するための指令を組み合わ

10

20

30

40

50

せたプログラムデータD_pが格納される。その内容は、バーコード記号読取装置100が、1キャラクタ内で隣接するバー及びスペースを含む $n(n \geq 2)$ 本のエレメントの幅を加算して2～n本加算エレメント幅を各々求めるステップと、求めた2～n本加算エレメント幅の各々を1キャラクタのモジュール数に換算して当該2～n本加算エレメント幅のモジュール数を求めるステップと、ここで求めた2～n本加算エレメント幅の各々のモジュール数が誤差1以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出するステップと、抽出された各々の2～n本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索するステップとを実行するプログラムである。ここに候補キャラクタとは対象キャラクタを絞り込んだキャラクタをいう。プログラムデータD_pには、システム起動用のシステムプログラムデータも含まれる。

10

【0083】

RAM22にはバーコード記号読取装置100を制御するためのシステムプログラムの他に、光学情報読取方法を実行するための演算や、抽出、検索、比較等の制御を実行する際の制御情報が一時格納される。例えば、電源がオンされると、電源オン情報を検出したCPU23は、ROM21からシステムプログラムを読み出してRAM22に展開し、システムを起動して、当該バーコード記号読取装置100全体を制御するようになされる。

【0084】

この例で、CPU23は、少なくとも、演算部23aや、抽出部23b、比較部23c、検索部23d等の4つの機能を有している。演算部23aは演算手段の一例を構成し、各対象キャラクタの幅データD10を入力し、幅データD10に基づいて1キャラクタ内で隣接するバー及びスペースを含む $n(n \geq 2)$ 本のエレメントの幅を加算して2～n本加算エレメント幅を各々求める。演算部23aは、求めた2～n本加算エレメント幅の各々を1キャラクタのモジュール数に換算して当該2～n本加算エレメント幅のモジュール数を求める。

20

【0085】

抽出部23bは、演算部23aによって求められた2～n本加算エレメント幅（以下で2～n本加算パターンともいう）の各々のモジュール数が誤差1以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出する。抽出部23bは、2～n本加算パターンを各々求める際に、1キャラクタ分のn本の幅データD10と、対象キャラクタの前後の1本以上の幅データD10とを抽出する。

30

【0086】

この例では、対象キャラクタの抽出基準を従来方式の誤差0.5以内の整数から誤差1以内の整数へと範囲を拡大し、1キャラクタ分のn本の幅データD10と、対象キャラクタの前後の1本以上の幅データD10とを抽出するようにした。従って、2～n本加算エレメント幅の各々のモジュール数が誤差1以内の整数となる対象キャラクタを従来方法に比べて、より多く抽出できるようになる。しかも、第1番目に相関が強いものをデコード結果とするだけでなく、第2番目、第3番目・・・に相関が強いものがデコード結果となる確率が高くなるというようにデコード結果の候補キャラクタを複数出力できるようになる。これにより、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

40

【0087】

比較部23cは、抽出部23bによって抽出された各々の2～n本加算エレメント幅のモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタ（理論値）とを比較する。比較部23cは判別機能を有しており、1キャラクタ内で隣接するバー及びスペースを含む $n(n \geq 2)$ 本のエレメントの幅を加算した合計値と、1キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて当該1キャラクタ分の幅データD10が適切か否かを判別する。

【0088】

この判別によって、キャラクタのバーの消失や分割等を判断できるようになる。また、

50

複数のスキヤンのデコード結果を組み合わせることが可能となり、汚れや、かすれがひどいバーコードにおいても読み取りが可能となる。また、バーの消失や分割等の判断結果に基づいて対象キャラクタを絞ることができ、計算量の軽減が図れ、処理時間の高速化につながる。

【 0 0 8 9 】

検索部 2 3 d 検索部 2 3 d は、候補キャラクタと最も相関が強い期待値キャラクタを検索する。例えば、上述の演算部 2 3 a が候補キャラクタのモジュール数の実測値と、期待値キャラクタのモジュール数の理想値との間の誤差を算出すると、検索部 2 3 d が、演算部 2 3 a によって算出された誤差を合計して得られる誤差値が最小となる候補キャラクタと最も相関が強い期待値キャラクタを検索する。この例では、誤差値が最小となる候補

10

【 0 0 9 0 】

マイクロプロセッサ 1 2 には期待値格納部 2 4 が接続される。期待値格納部 2 4 には期待値キャラクタ（理論値）が格納されている。期待値キャラクタは CODE 1 2 8 規格のキャラクタの各エレメントを構成するモジュール数（理論値）である。これらにより、バーコード記号読取装置 1 0 0 を構成する。CPU 2 3 内の演算部 2 3 a、抽出部 2 3 b、比較部 2 3 c 及び、検索部 2 3 d はソフトウェアでも、ハードウェアでもどちらで構築してもよい。

20

【 実施例 1 】

【 0 0 9 1 】

続いて、図 2 A ~ G を参照して、第 1 の実施例に係る 1 本データ、2 ~ 6 本加算データの算出例について説明する。この実施例では、CODE 1 2 8 規格のバーコード記号 1 について説明する。CODE 1 2 8 規格によれば、図 2 A に示す 1 キャラクタは、3 本の黒部分のバー（1）、（3）、（5）及び 3 本の白部分のスペース（2）、（4）、（6）から構成される。図中、C 1 は対象キャラクタであり、C 0 は対象キャラクタ C 1 の 1 つ前のキャラクタであり、C 2 はその 1 つ後ろのキャラクタである。

【 0 0 9 2 】

1 キャラクタの全幅は 1 1 モジュールから構成される。この例で、黒部分のバー（1）は 4 モジュールから成り、バー（3）及びバー（5）は各々 1 モジュールから成る。白部分のスペース（2）は 1 モジュールから成り、スペース（4）及び（6）は各々 2 モジュールから成る。

30

【 0 0 9 3 】

スタートコードは 3 種類あり、それぞれ 1 0 3 種類のコードパターンが表される。CODE 1 2 8 規格では、アスキー 1 2 8 文字を表現できるようになっている。スタートコード C を使用すれば、1 キャラクタで数字 2 桁を表現でき、非常に高い印字密度で表現できる。CODE 1 2 8 のバーコード記号 1 は 8 キャラクタから構成される（図 5 A 参照）。バーに用いるモジュール数の和は偶数パリティであり、スペースに用いるモジュール数の和は奇数パリティとなっている。偶数パリティ及び奇数パリティを利用した自己機能を持っている。

40

【 0 0 9 4 】

図 2 B に示す W 1 1 は対象キャラクタ C 1 の黒部分のバー（1）の幅である。バー（1）の幅はカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 1 データを成す。W 1 2 はバー（1）に続くスペース（2）の幅である。スペース（2）の幅もカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 2 データを成す。W 1 3 は対象キャラクタ C 1 のスペース（2）に続くバー（3）の幅である。バー（3）の幅はカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 3 データを成す。

【 0 0 9 5 】

W 1 4 は対象キャラクタ C 1 のバー（3）に続くスペース（4）の幅である。スペース（4）の幅もカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 4 データを成

50

す。W 1 5 はスペース (4) に続くバー (5) の幅である。バー (5) の幅はカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 5 データを成す。W 1 6 は対象キャラクタ C 1 のバー (5) に続くスペース (6) の幅である。スペース (6) の幅もカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 6 データを成す。上述の 6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データは 1 キャラクタの 1 本データであり、幅データ D 1 0 を構成する。

【 0 0 9 6 】

図 2 C に示す W 2 1 は対象キャラクタ C 1 のバー (1) とスペース (2) の各々が演算部 2 3 a によって加算された幅 (以下バー (1) + スペース (2) と記す) である。バー (1) + スペース (2) の加算幅を示すカウントデータ値は幅 W 2 1 データを成す。W 2 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) の加算幅を示すカウントデータ値は幅 W 2 2 データを成す。

10

【 0 0 9 7 】

W 2 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算された幅である。バー (3) + スペース (4) の加算幅を示すカウントデータ値は幅 W 2 3 データを成す。W 2 4 は対象キャラクタ C 1 のバー (3) に続くスペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。スペース (4) + バー (5) の加算幅を示すカウントデータ値は幅 W 2 4 データを成す。W 2 5 は対象キャラクタ C 1 のスペース (4) に続くバー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。バー (5) + スペース (6) の加算幅を示すカウントデータ値は幅 W 2 5 データを成す。上述の 5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データは 1 キャラクタの 2 本加算データであり、幅データ D 1 0 を構成する。

20

【 0 0 9 8 】

図 2 D に示す W 3 1 は対象キャラクタ C 1 のバー (1) 、スペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算された幅である。バー (1) + スペース (2) + バー (3) の加算幅を示すカウントデータ値は幅 W 3 1 データを成す。W 3 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) 、バー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) + スペース (4) の加算幅を示すカウントデータ値は幅 W 3 2 データを成す。

30

【 0 0 9 9 】

W 3 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) 、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。バー (3) + スペース (4) + バー (5) の加算幅を示すカウントデータ値は幅 W 3 3 データを成す。W 3 4 は対象キャラクタ C 1 のバー (3) に続くスペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。スペース (4) + バー (5) + スペース (6) の加算幅を示すカウントデータ値は幅 W 3 4 データを成す。上述の 4 種類の幅 W 3 1 データ ~ 幅 W 3 4 データは 1 キャラクタの 3 本加算データであり、幅データ D 1 0 を構成する。

【 0 1 0 0 】

図 2 E に示す W 4 1 は対象キャラクタ C 1 のバー (1) 、スペース (2) 、バー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算された幅である。バー (1) + スペース (2) + バー (3) + スペース (4) の加算幅を示すカウントデータ値は幅 W 4 1 データを成す。W 4 2 の対象キャラクタ C 1 はバー (1) に続くスペース (2) 、バー (3) 、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) + スペース (4) + バー (5) の加算幅を示すカウントデータ値は幅 W 4 2 データを成す。

40

【 0 1 0 1 】

W 4 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) 、スペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。バー (

50

3) + スペース (4) + バー (5) + スペース (6) の加算幅を示すカウントデータ値は幅 W 4 3 データを成す。上述の 3 種類の幅 W 4 1 データ ~ 幅 W 4 3 データは 1 キャラクタの 4 本加算データであり、幅データ D 1 0 を構成する。

【 0 1 0 2 】

図 2 F に示す W 5 1 は対象キャラクタ C 1 のバー (1)、スペース (2)、バー (3)、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。バー (1) + スペース (2) + バー (3) + スペース (4) + バー (5) の加算幅を示すカウントデータ値は幅 W 5 1 データを成す。

【 0 1 0 3 】

W 5 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2)、バー (3)、スペース (4)、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) + スペース (4) + バー (5) + スペース (6) の加算幅を示すカウントデータ値は幅 W 5 2 データを成す。上述の 2 種類の幅 W 5 1 データ及び幅 W 5 2 データは 1 キャラクタの 5 本加算データであり、幅データ D 1 0 を構成する。

【 0 1 0 4 】

図 2 G に示す W 6 1 は対象キャラクタ C 1 のバー (1)、スペース (2)、バー (3)、スペース (4)、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。バー (1) + スペース (2) + バー (3) + スペース (4) + バー (5) + スペース (6) の加算幅を示すカウントデータ値は幅 W 6 1 データを成す。1 種類の幅 W 6 1 データは 1 キャラクタの 6 本加算データであり、幅データ D 1 0 を構成する。

【 0 1 0 5 】

上述のカウンタ部 4 から出力されるカウントデータ値から、6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データ (1 本データ)、5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データ (2 本加算データ)、4 種類の幅 W 3 1 データ ~ 幅 W 3 4 データ (3 本加算データ)、3 種類の幅 W 4 1 データ ~ 幅 W 4 3 データ (4 本加算データ)、2 種類の幅 W 5 1 データ及び幅 W 5 2 データ (5 本加算データ) 及び、1 種類の幅 W 6 1 データ (6 本加算データ) の合計で、2 1 種類の 1 本データ、2 ~ 6 本加算データが得られる。

【 0 1 0 6 】

続いて、本発明に係る光学情報読取方法について、図 3 及び図 4 を参照し、第 1 の実施例に係る難読キャラクタのデコード例を説明する。この実施例では、CODE 1 2 8 規格に基づくバーコード記号 1 を読み取って、そのスタートマージンを検索し、更に、スタートマージンに続く対象キャラクタをデコード処理してバーコードデータ D 1 2 を出力する場合を例に挙げる。その場合のバーコード記号 1 のバー及びスペースの各エレメントの誤差許容度 (最大誤差範囲) が、 ± 1.0 モジュールに設定される。バーコード記号 1 については図 2 に示した対象キャラクタ C 1 について説明する (キャラクタデコードアルゴリズム) 。

【 0 1 0 7 】

これらをデコード条件にして、まず、図 3 に示すフローチャートのステップ S T 1 でマイクロプロセッサ 1 2 はスタートマージンを検索する。このとき、光学読取部 2 は、バーコード記号 1 を読み取って、コード読取データ D 2 を発生する。コード読取データ D 2 は、光学読取部 2 からカウンタ部 4 へ出力される。クロック発生部 3 は、所定周波数のサンプリング用の C L K 信号をカウンタ部 4 に出力する。

【 0 1 0 8 】

カウンタ部 4 は光学読取部 2 から出力されるコード読取データ D 2 を入力すると共に、クロック発生部 3 から出力される C L K 信号を入力し、C L K 信号に基づいてコード読取データ D 2 をカウントし、コード読取データ D 2 を白部分のスペースおよび黒部分のバーを含むエレメントの 1 本幅及び 6 本加算幅を示すカウントデータ D 4 に変換する。カウントデータ D 4 は、バーコード記号 1 における各バー (黒部分) の幅およびスペース (白部分) の幅を示す数値データ列となる。カウントデータ D 4 は、幅データ格納部 1 0 に保存される。

10

20

30

40

50

【 0 1 0 9 】

m 段データバッファ 5 はカウンタ部 4 から出力されるカウントデータ D 4 を、m 段分 (m は任意の整数) だけ一時格納する。m 段は、バーコード記号 1 の印刷状態により任意に設定されている。x 倍器 7 は、カウントデータ D 4 からスタートマージンを検索するに当たり、カウントデータ D 4 を何倍かを示す「 x 」が予め設定され、その x 倍の演算を行って、x 倍データ D 7 を出力する。

【 0 1 1 0 】

セクタ部 6 は、マイクロプロセッサ 1 2 の選択制御を受けて、m 段データバッファ 5 に一時格納された m 段分のカウントデータ D 4 のうち、比較対象とするデータを選択する。比較器 8 はセクタ部 6 が選択した、m 段データバッファ 5 中の k 段前までの各カウントデータ D 4 と、x 倍器 7 から出力される x 倍データ D 7 とを比較する。

10

【 0 1 1 1 】

比較結果によって x 倍器 7 からの x 倍データ D 7 の方が小さいときは、スタートマージン候補があったと判定し、スタートマージン候補検索フラグ (以下で S M F データ D 9 という) を発生する。S M F データ D 9 はスタートマージン格納部 9 に出力される。スタートマージン格納部 9 には比較器 8 から出力される S M F データ D 9、そのとき比較対象としたカウントデータ値 = D 1 0 とが対応させて格納される。

【 0 1 1 2 】

ステップ S T 2 で C P U 2 3 は 1 キャラクタ分の幅データ D 1 0 (6 本) と、その前後の 5 本ずつ幅データ D 1 0 を抽出する。図 2 B に示した対象キャラクタ C 1 によれば、黒部分のバー (1) の幅 W 1 1 がカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 1 データとなる。バー (1) に続く白部分のスペース (2) の幅 W 1 2 もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 2 データとなる。スペース (2) に続くバー (3) の幅 W 1 3 はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 3 データとなる。

20

【 0 1 1 3 】

このバー (3) に続くスペース (4) の幅 W 1 4 もカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 4 データとなる。スペース (4) に続くバー (5) の幅 W 1 5 はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 5 データとなる。バー (5) に続くスペース (6) の幅 W 1 6 もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 6 データとなる。上述の 6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データは幅データ D 1 0 を構成する。幅データ D 1 0 はインターフェース部 1 1 を介して C P U 2 3 に出力される。

30

【 0 1 1 4 】

C P U 2 3 では、演算部 2 3 a が各対象キャラクタの幅データ D 1 0 を入力し、幅データ D 1 0 に基づいて 1 キャラクタ内で隣接するバー及びスペースを含む n (n ≥ 2) 本のエレメントの幅を加算して 2 ~ n 本加算パターンを各々求める。この例では n = 6 である。2 ~ n 本加算エレメント幅を各々求める際に、1 キャラクタ分の 6 本の幅データ D 1 0 と、対象キャラクタの前後の 5 本の幅データ D 1 0 とを抽出する。

【 0 1 1 5 】

例えば、図 2 C に示した対象キャラクタ C 1 のバー (1) とスペース (2) の各々が演算部 2 3 a によって加算され、その加算幅 W 2 1 を示すカウントデータ値が幅 W 2 1 データとなる。対象キャラクタ C 1 のバー (1) に続くスペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算され、その加算幅 W 2 2 を示すカウントデータ値が幅 W 2 2 データとなる。対象キャラクタ C 1 のスペース (2) に続くバー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算され、その加算幅 W 2 3 を示すカウントデータ値が幅 W 2 3 データとなる。

40

【 0 1 1 6 】

対象キャラクタ C 1 のバー (3) に続くスペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算され、その加算幅 W 2 4 を示すカウントデータ値が幅 W 2 4 データと

50

なる。対象キャラクタ C 1 のスペース (4) に続くバー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算され、その加算幅 W 2 5 を示すカウントデータ値が幅 W 2 5 データとなる。上述の 5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データは幅データ D 1 0 を構成する。

【 0 1 1 7 】

図 2 D に示した対象キャラクタ C 1 のバー (1) 、スペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算され、その加算幅 W 3 1 を示すカウントデータ値が幅 W 3 1 データとなる。対象キャラクタ C 1 のバー (1) に続くスペース (2) 、バー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算され、その加算幅 W 3 2 を示すカウントデータ値が幅 W 3 2 データとなる。

10

【 0 1 1 8 】

対象キャラクタ C 1 のスペース (2) に続くバー (3) 、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算され、その加算幅 W 3 3 を示すカウントデータ値が幅 W 3 3 データとなる。対象キャラクタ C 1 のバー (3) に続くスペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算され、その加算幅 W 3 4 を示すカウントデータ値が幅 W 3 4 データとなる。上述の 4 種類の幅 W 3 1 データ ~ 幅 W 3 4 データは幅データ D 1 0 を構成する。

【 0 1 1 9 】

図 2 E に示した対象キャラクタ C 1 のバー (1) 、スペース (2) 、バー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算され、その加算幅 W 4 1 を示すカウントデータ値が幅 W 4 1 データとなる。対象キャラクタ C 1 はバー (1) に続くスペース (2) 、バー (3) 、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算され、その加算幅 W 4 2 を示すカウントデータ値が幅 W 4 2 データとなる。

20

【 0 1 2 0 】

対象キャラクタ C 1 のスペース (2) に続くバー (3) 、スペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算され、その加算幅 W 4 3 を示すカウントデータ値が幅 W 4 3 データとなる。上述の 3 種類の幅 W 4 1 データ ~ 幅 W 4 3 データは幅データ D 1 0 を構成する。

【 0 1 2 1 】

図 2 F に示した対象キャラクタ C 1 のバー (1) 、スペース (2) 、バー (3) 、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算され、その加算幅 W 5 1 を示すカウントデータ値が幅 W 5 1 データとなる。対象キャラクタ C 1 のバー (1) に続くスペース (2) 、バー (3) 、スペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算され、その加算幅 W 5 2 を示すカウントデータ値が幅 W 5 2 データを成す。上述の 2 種類の幅 W 5 1 データ及び幅 W 5 2 データは幅データ D 1 0 を構成する。

30

【 0 1 2 2 】

図 2 G に示した対象キャラクタ C 1 のバー (1) 、スペース (2) 、バー (3) 、スペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算され、その加算幅 W 6 1 を示すカウントデータ値が幅 W 6 1 データとなる。1 種類の幅 W 6 1 データは幅データ D 1 0 を構成する。

40

【 0 1 2 3 】

上述のカウンタ部 4 から出力されるカウントデータ値から、6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データ (1 本データ) 、5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データ (2 本加算データ) 、4 種類の幅 W 3 1 データ ~ 幅 W 3 4 データ (3 本加算データ) 、3 種類の幅 W 4 1 データ ~ 幅 W 4 3 データ (4 本加算データ) 、2 種類の幅 W 5 1 データ及び幅 W 5 2 データ (5 本加算データ) 及び、1 種類の幅 W 6 1 データ (6 本加算データ) の合計で、2 1 種類の 1 本データ、2 ~ 6 本加算データ が得られる。

【 0 1 2 4 】

ステップ S T 3 で C P U 2 3 は対象キャラクタ C 1 の 1 モジュール幅を算出する。この

50

例では、演算部 23a は、先に求めた 2 ~ 6 本加算パターン（エレメント幅）の各々を 1 キャラクタのモジュール数に換算して 2 ~ 6 本加算パターンのモジュール数を求める。図 2B ~ G に示したように、カウンタ部 4 から出力されるカウントデータ値から、6 種類の幅 W11 データ ~ 幅 W16 データ、5 種類の幅 W21 データ ~ 幅 W25 データ、4 種類の幅 W31 データ ~ 幅 W34 データ、3 種類の幅 W41 データ ~ 幅 W43 データ、2 種類の幅 W51 データ及び幅 W52 データ及び、1 種類の幅 W61 データが CPU 23 に順次出力される。

【0125】

例えば、演算部 23a は幅 W21 データを入力し、対象キャラクタ C1 のバー（1）とスペース（2）とを加算した加算幅 W21 のモジュール数を求める。モジュール数は全キャラクタ幅 = 11 モジュールに対する加算幅 W21 の比率、例えば、2.4 を算出する。この場合、加算幅 W21 のモジュール数は、バーやスペース等の各エレメントの最大誤差が 1.0 以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出するので、加算幅 W21 のモジュール数は「2」又は「3」となる。

【0126】

この例では、各エレメントの最大誤差範囲を ±1.0 モジュール以内に拡張したので、各エレメントの誤差許容度が 1.0 以内まで許容量を増加できるようになる。許容量を増加したので、対象キャラクタ一致を採る条件を増加できる。これにより、バーコード記号 1 の誤読取を防止できるようになる。

【0127】

ステップ ST4 で CPU 23 は抽出された 21 種類の幅データ D10 を 1 ~ 6 本加算パターン（エレメント幅）に変換する。この例では、6 種類の幅 W11 データ ~ 幅 W16 データが 1 キャラクタの 1 本データに変換され、1 本データに基づく対象キャラクタ C1 の被認識パターンとして、1 本パターンが作成される。5 種類の幅 W21 データ ~ 幅 W25 データは 1 キャラクタの 2 本加算データに変換され、2 本加算データに基づく対象キャラクタ C1 の 2 本加算パターン（被認識パターン）が作成される。4 種類の幅 W31 データ ~ 幅 W34 データは 1 キャラクタの 3 本加算データに変換され、3 本加算データに基づく対象キャラクタ C1 の 3 本加算パターンが作成される。

3 種類の幅 W41 データ ~ 幅 W43 データは 1 キャラクタの 4 本加算データに変換され、4 本加算データに基づく対象キャラクタ C1 の 4 本加算パターンが作成される。2 種類の幅 W51 データ及び幅 W52 データは 1 キャラクタの 5 本加算データに変換され、5 本加算データに基づく対象キャラクタ C1 の 5 本加算パターンが作成される。1 種類の幅 W61 データは 1 キャラクタの 6 本加算データに変換され、6 本加算データに基づく対象キャラクタ C1 の 6 本加算パターンが作成される。

【0128】

ステップ ST5 で CPU 23 は対象キャラクタを絞り込むために、1 本パターン、2 ~ 6 本加算パターンの全てに対してモジュール数の誤差が 1 以内の対象キャラクタを検索して候補キャラクタを決定する。このとき、CPU 23 では抽出部 23b が、演算部 23a によって求められた 1 本パターン、2 ~ 6 本加算パターンの各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出する。その際に、抽出部 23b は、演算部 23a によって求められた 1 本パターン、2 ~ 6 本加算パターンの 1 キャラクタ分の 6 本の幅データ D10 と、対象キャラクタの前後の（6 - 1）本の幅データ D10 とを抽出する。

【0129】

ステップ ST6 で CPU 23 は、候補キャラクタが 1 つ以上存在するか否かに基づいて制御を分岐する。対象キャラクタが 1 つ以上（複数）存在する場合は、ステップ ST7 で 1 ~ 6 本加算パターンの実測値とその理想値との間の誤差を算出し、この誤差の合計が最小となった候補キャラクタをデコード結果とする。この誤差値が最小となる候補キャラクタは、期待値キャラクタと最も相関が強いキャラクタである。この例では、総合的な誤差が最も少ない対象キャラクタを候補キャラクタとして選択できるようになる。

【 0 1 3 0 】

このとき、ＣＰＵ２３では、期待値キャラクタ（理論値）が期待値格納部２４からＣＰＵ２３の比較部２３ｃへ読み出される。比較部２３ｃは、抽出部２３ｂによって抽出された各々の２～６本加算パターン（エレメント幅）のモジュール数の候補キャラクタと、候補キャラクタを評価するための期待値キャラクタ（理論値）とを比較する。この比較結果は検索部２３ｄへ出力される。

【 0 1 3 1 】

検索部２３ｄでは、候補キャラクタと最も相関が強い期待値キャラクタが検索される。例えば、上述の演算部２３ａが候補キャラクタのモジュール数の実測値と、期待値キャラクタのモジュール数の理想値との間の誤差を算出すると、検索部２３ｄが、演算部２３ａによって算出された誤差を合計して得られる誤差値が最小となる候補キャラクタと最も相関が強い期待値キャラクタを検索する。この例では、誤差値が最小となる候補キャラクタと最も相関が強い期待値キャラクタを検索するので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。

10

【 0 1 3 2 】

以後は従来方式と同じ内容となる。例えば、図４に示すステップＳＴ８でＣＰＵ２３は、デコード結果の候補キャラクタがマージン直後の対象キャラクタか否かに基づいて制御を分岐する。デコード結果の候補キャラクタがマージン直後の対象キャラクタである場合は、ステップＳＴ９に移行してＣＰＵ２３はデコード結果の候補キャラクタがストップキャラクタか否かに基づいて制御を分岐する。

20

【 0 1 3 3 】

デコード結果の候補キャラクタがストップキャラクタではない場合は、ステップＳＴ１０に移行してＣＰＵ２３は誤り検知（パリティチェック）を実行し、チェックデジットが正しいか否かに基づいて制御を分岐する。チェックデジットが正しい場合は、読み取りを完了し、ステップＳＴ１１でＣＰＵ２３はデコード結果となされた候補キャラクタのバーコードデータＤ１２を出力する。

【 0 1 3 4 】

なお、ステップＳＴ８でＣＰＵ２３はデコード結果の候補キャラクタがマージン直後の対象キャラクタでない場合は、ステップＳＴ１２に移行して、ＣＰＵ２３はデコード結果の候補キャラクタがスタートキャラクタか否かに基づいて制御を分岐する。

30

【 0 1 3 5 】

上述のステップＳＴ９でＣＰＵ２３はデコード結果の候補キャラクタがストップキャラクタである場合及び、ステップＳＴ１２でデコード結果の候補キャラクタがスタートキャラクタである場合は、図３に示すステップＳＴ１３に移行する。ステップＳＴ１３で、ＣＰＵ２３は次キャラクタ（６本後ろのデータ）のデコード処理を実行する。その後、ステップＳＴ２に戻る。

【 0 1 3 6 】

また、ステップＳＴ１２でデコード結果の候補キャラクタがスタートキャラクタでない場合及び、ステップＳＴ１０でチェックデジットが誤っている場合は、ステップＳＴ１４に移行して、ＣＰＵ２３は読み取りエラー処理を実行する。読み取りエラー処理では次のマージンを検索する等が行われる。キャラクタ認識がエラーとなった場合に、エラーキャラクタを仮に配置してバーコードデータを作成するようにしてもよい。

40

【 0 1 3 7 】

その後、ステップＳＴ１５に移行して、ＣＰＵ２３は終了を判別する。例えば、当該バーコード記号読取装置１００に対して読取終了指令が検出され、読取終了指令が検出された場合は、バーコード記号１の読み取り制御を終了する。読取終了指令が検出されていない場合は、バーコード記号１の読み取り制御を継続するべく、ステップＳＴ１に戻って、上述した内容を繰り返すようになされる。

【 0 1 3 8 】

このように第１の実施例に係るバーコード記号読取装置１００によれば、バーコード記

50

号 1 を読み取ってそのデコード結果を出力する場合に、演算部 2 3 a は、1 キャラクタ内で隣接するバー及びスペースを含む $n = 6 (n - 2)$ 本のエレメントの幅を加算して 2 ~ 6 本加算パターン (エレメント幅) を各々求める。演算部 2 3 a は、2 ~ 6 本加算パターンの各々を 1 キャラクタのモジュール数に換算して当該 2 ~ 6 本加算パターンのモジュール数を求める。

【 0 1 3 9 】

抽出部 2 3 b は、演算部 2 3 a によって求められた 2 ~ 6 本加算パターンの各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んで候補キャラクタを抽出する。これらを前提にして、検索部 2 3 d が抽出部 2 3 b によって抽出された各々の 2 ~ 6 本加算パターンのモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索するようになる。

10

【 0 1 4 0 】

この検索によって、複数の対象キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータ D 1 2 を当該バーコード記号 1 の読取時のデコード結果として出力できるようになるので、従来方式のバーコード読み取り方法に比べて大幅にバーコード記号 1 の誤読取を防止できるようになる。

【 0 1 4 1 】

しかも、誤差の範囲を従来方式の ± 0.5 から ± 1.0 に拡大し、候補キャラクタを増加させている。そして、1 キャラクタ分の 6 本の幅データ D 1 0 と、当該キャラクタ C 1 の前後の 5 本の幅データ D 1 0 とを抽出するので、2 ~ 6 本加算パターンの各々のモジュール数が誤差 1 以内の整数となる対象キャラクタを絞り込んだ候補キャラクタを従来方法に比べてより多く抽出できるようになる。

20

【 0 1 4 2 】

また、第 1 番目に相関が強い候補キャラクタをデコード結果とするだけでなく、第 2 番目、第 3 番目・・・に相関が強い候補キャラクタがデコード結果となる確率が高くなるというようにデコード結果の候補キャラクタを複数出力できるようになる。これにより、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。これにより、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。なお、誤差の範囲を設定する際の閾値を可変することで、幅データ D 1 0 の信頼性を選択することも可能となる。

30

【 0 1 4 3 】

続いて、図 5 A ~ G を参照して、第 1 の実施例に係る 1 本データ、2 本加算データの他の算出例 (その 1) について説明する。この例では、対象キャラクタ C 1 の両側のキャラクタ C 0, C 2 まで加算範囲を拡張したものである。対象キャラクタ C 1 と隣接する 1 つ前のキャラクタ C 0 のスペース (6) 及びバー (5) の 2 エレメントまで加算範囲を拡張し、1 つ後ろのキャラクタ C 2 のバー (1) 及びスペース (2) の 2 エレメントまで加算範囲を拡張する場合を前提とする。

【 0 1 4 4 】

図 5 A に示す対象キャラクタ C 1 の 1 つ前にはキャラクタ C 0 が設けられ、対象キャラクタ C 1 の 1 つ後ろにはキャラクタ C 2 が設けられる。各々のキャラクタ C 0 ~ C 2 は、その各々が全幅 11 モジュールから構成される。3 個のバー (1), (3), (5) 及び 3 個のスペース (2), (4), (6) から構成される。バー (1) のモジュール数は「4」であり、バー (3), (5) のモジュール数は「1」である。また、スペース (2) のモジュール数は「1」であり、スペース (4), (6) のモジュール数は「2」である。これらのキャラクタ C 0 ~ C 2 を有したバーコードの場合を例に挙げて、対象キャラクタの作成する場合を説明する。

40

【 0 1 4 5 】

図 5 B に示す対象キャラクタ C 1 の W 1 1 は、黒部分のバー (1) の幅である。バー (1) の幅はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 1 データを成す

50

。W 1 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) の幅である。スペース (2) の幅もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 2 データを成す。W 1 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) の幅である。バー (3) の幅はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 3 データを成す。

【 0 1 4 6 】

W 1 4 は対象キャラクタ C 1 のバー (3) に続くスペース (4) の幅である。スペース (4) の幅もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 4 データを成す。W 1 5 は対象キャラクタ C 1 のスペース (4) に続くバー (5) の幅である。バー (5) の幅はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 5 データを成す。W 1 6 は対象キャラクタ C 1 のバー (5) に続くスペース (6) の幅である。スペース (6) の幅もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 6 データを成す。上述の 6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データは 1 キャラクタの 1 本データであり、幅データ D 1 0 を構成する。」

図 5 C に示す W2-1 は、1 つ前の対象キャラクタ C 0 のスペース (6) と対象キャラクタ C 1 のバー (1) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W2-1 データを成す。W 2 1 は 1 つ前のキャラクタ C 0 のスペース (6) に続く対象キャラクタ C 1 のバー (1) とスペース (2) の各々が演算部 2 3 a によって加算された幅 (以下バー (1) + スペース (2) と記す) である。バー (1) + スペース (2) の加算幅を示すカウントデータ値が幅 W 2 1 データを成す。W 2 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) の加算幅を示すカウントデータ値が幅 W 2 2 データを成す。

【 0 1 4 7 】

W 2 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算された幅である。バー (3) + スペース (4) の加算幅を示すカウントデータ値が幅 W 2 3 データを成す。W 2 4 は対象キャラクタ C 1 のバー (3) に続くスペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。スペース (4) + バー (5) の加算幅を示すカウントデータ値が幅 W 2 4 データを成す。

【 0 1 4 8 】

W 2 5 は対象キャラクタ C 1 のスペース (4) に続くバー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。バー (5) + スペース (6) の加算幅を示すカウントデータ値が幅 W 2 5 データを成す。W 2 6 はバー (5) に続くスペース (6) 及び 1 つ後ろのキャラクタ C 2 のバー (1) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W 2 6 データを成す。上述の 7 種類の幅 W2-1 データ ~ 幅 W 2 6 データは 1 キャラクタの 2 本加算データであり、幅データ D 1 0 を構成する。

【 0 1 4 9 】

図 5 D に示す W3-2 は、1 つ前のキャラクタ C 0 のバー (5) 、スペース (6) 及び、対象キャラクタ C 1 のバー (1) の各々が演算部 2 3 a によって加算された幅である。バー (5) + スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W3-2 データを成す。W3-1 は、1 つ前のキャラクタ C 0 のスペース (6) 、対象キャラクタ C 1 のバー (1) 及び、スペース (2) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) + スペース (2) の加算幅を示すカウントデータ値が幅 W3-1 データを成す。

【 0 1 5 0 】

W 3 1 は対象キャラクタ C 1 のバー (1) 、スペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算された幅である。バー (1) + スペース (2) + バー (3) の加算幅を示すカウントデータ値が幅 W 3 1 データを成す。W 3 2 は対象キャラクタ C 1 のバ

ー(1)に続くスペース(2)、バー(3)及びスペース(4)の各々が演算部23aによって加算された幅である。スペース(2)+バー(3)+スペース(4)の加算幅を示すカウントデータ値が幅W32データを成す。

【0151】

W33は対象キャラクタC1のスペース(2)に続くバー(3)、スペース(4)及びバー(5)の各々が演算部23aによって加算された幅である。バー(3)+スペース(4)+バー(5)の加算幅を示すカウントデータ値が幅W33データを成す。W34は対象キャラクタC1のバー(3)に続くスペース(4)、バー(5)及びスペース(6)の各々が演算部23aによって加算された幅である。スペース(4)+バー(5)+スペース(6)の加算幅を示すカウントデータ値が幅W34データを成す。

10

【0152】

W35は、対象キャラクタC1のスペース(4)に続くバー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された幅である。バー(5)+スペース(6)+バー(1)の加算幅を示すカウントデータ値が幅W35データを成す。W36は、対象キャラクタC1のバー(3)に続くスペース(6)の1つ後ろのキャラクタC2のバー(1)及びスペース(2)の各々が演算部23aによって加算された幅である。スペース(6)+バー(1)+スペース(2)の加算幅を示すカウントデータ値が幅W36データを成す。上述の8種類の幅W3-2データ~幅W36データは1キャラクタの3本加算データであり、幅データD10を構成する。

【0153】

20

図5Eに示すW4-2は、1つ前のキャラクタC0のバー(5)、スペース(6)、対象キャラクタC1のバー(1)及びスペース(2)の各々が演算部23aによって加算された幅である。バー(5)+スペース(6)+バー(1)+スペース(2)の加算幅を示すカウントデータ値が幅W4-2データを成す。W4-1は、1つ前のキャラクタC0のスペース(6)、対象キャラクタC1のバー(1)、スペース(2)及びバー(3)の各々が演算部23aによって加算された幅である。スペース(6)+バー(1)+スペース(2)+バー(3)の加算幅を示すカウントデータ値が幅W4-1データを成す。

【0154】

W41は対象キャラクタC1のバー(1)、スペース(2)、バー(3)及びスペース(4)の各々が演算部23aによって加算された幅である。バー(1)+スペース(2)+バー(3)+スペース(4)の加算幅を示すカウントデータ値が幅W41データを成す。W42は対象キャラクタC1のバー(1)に続くスペース(2)、バー(3)、スペース(4)及びバー(5)の各々が演算部23aによって加算された幅である。スペース(2)+バー(3)+スペース(4)+バー(5)の加算幅を示すカウントデータ値が幅W42データを成す。

30

【0155】

W43は対象キャラクタC1のスペース(2)に続くバー(3)、スペース(4)、バー(5)及びスペース(6)の各々が演算部23aによって加算された幅である。バー(3)+スペース(4)+バー(5)+スペース(6)の加算幅を示すカウントデータ値が幅W43データを成す。W44は、対象キャラクタC1のバー(3)に続くスペース(4)、バー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された幅である。スペース(4)+バー(5)+スペース(6)+バー(1)の加算幅を示すカウントデータ値が幅W44データを成す。

40

【0156】

W45は、対象キャラクタC1の白のスペース(4)に続くバー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)及び、スペース(2)の幅である。バー(5)+スペース(6)+バー(1)+スペース(2)の加算幅を示すカウントデータ値が幅W45データを成す。上述の7種類の幅W4-2データ~幅W45データは、1キャラクタの4本加算データであり、幅データD10を構成する。

【0157】

50

図5 Fに示すW5-2は、1つ前のキャラクタC0のバー(5)、スペース(6)、対象キャラクタC1のバー(1)、スペース(2)及びバー(3)の各々が演算部23aによって加算された幅である。バー(5)+スペース(6)+バー(1)+スペース(2)+バー(3)の加算幅を示すカウントデータ値が幅W5-2データを成す。W5-1は、1つ前のキャラクタC0のスペース(6)+キャラクタC1のバー(1)+スペース(2)+バー(3)+スペース(4)の各々が演算部23aによって加算された幅である。スペース(6)+バー(1)+スペース(2)+バー(3)+スペース(4)の加算幅を示すカウントデータ値が幅W5-1データを成す。

【0158】

W51は対象キャラクタC1のバー(1)、スペース(2)、バー(3)、スペース(4)及びバー(5)の各々が演算部23aによって加算された幅である。バー(1)+スペース(2)+バー(3)+スペース(4)+バー(5)の加算幅を示すカウントデータ値が幅W51データを成す。W52は対象キャラクタC1のバー(1)に続くスペース(2)、バー(3)、スペース(4)、バー(5)及びスペース(6)の各々が演算部23aによって加算された幅である。スペース(2)+バー(3)+スペース(4)+バー(5)+スペース(6)の加算幅を示すカウントデータ値が幅W52データを成す。

【0159】

W53は、対象キャラクタC1のスペース(2)に続くバー(3)、スペース(4)、バー(5)、スペース(6)及び1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された幅である。バー(3)+スペース(4)+バー(5)+スペース(6)+バー(1)の加算幅を示すカウントデータ値が幅W53データを成す。W54は、対象キャラクタC1のバー(3)に続くスペース(4)、バー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)及びスペース(2)の各々が演算部23aによって加算された幅である。スペース(4)+バー(5)+スペース(6)+バー(1)+スペース(2)の加算幅を示すカウントデータ値が幅W54データを成す。上述の6種類の幅W5-2データ及び幅W54データは1キャラクタの5本加算データであり、幅データD10を構成する。

【0160】

図5 Gに示すW6-2は、1つ前のキャラクタC0のバー(5)、スペース(6)、対象キャラクタC1のバー(1)、スペース(2)、バー(3)及びスペース(4)の各々が演算部23aによって加算された幅である。バー(5)+スペース(6)+キャラクタC1のバー(1)+スペース(2)+バー(3)+スペース(4)の加算幅を示すカウントデータ値が幅W6-2データを成す。

【0161】

W6-1は、1つ前のキャラクタC0のスペース(6)、対象キャラクタC1のバー(1)、スペース(2)、バー(3)、スペース(4)及びバー(5)の各々が演算部23aによって加算された幅である。スペース(6)+バー(1)+スペース(2)+バー(3)+スペース(4)+バー(5)の加算幅を示すカウントデータ値が幅W6-1データを成す。

【0162】

W61は対象キャラクタC1のバー(1)、スペース(2)、バー(3)、スペース(4)、バー(5)及びスペース(6)の各々が演算部23aによって加算された幅である。バー(1)+スペース(2)+バー(3)+スペース(4)+バー(5)+スペース(6)の加算幅を示すカウントデータ値が幅W61データを成す。

【0163】

W62は対象キャラクタC1のバー(1)に続くスペース(2)、バー(3)、スペース(4)、バー(5)、スペース(6)及び1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された幅である。スペース(2)+バー(3)+スペース(4)+バー(5)+スペース(6)+バー(1)の加算幅を示すカウントデータ値が幅W62データを成す。

【 0 1 6 4 】

W 6 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3)、スペース (4)、バー (5)、スペース (6)、1 つ後ろのキャラクタ C 2 のバー (1) 及びスペース (2) の各々が演算部 2 3 a によって加算された幅である。バー (3) + スペース (4) + バー (5) + スペース (6) + バー (1) + スペース (2) の加算幅を示すカウントデータ値が幅 W 6 3 データを成す。上述の 5 種類の幅 W 6-2 データ及び幅 W 6 3 データは 1 キャラクタの 6 本加算データであり、幅データ D 1 0 を構成する。

【 0 1 6 5 】

上述のカウンタ部 4 から出力されるカウントデータ値に基づいて、6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データ (1 本データ)、7 種類の幅 W 2-1 データ ~ 幅 W 2 6 データ (2 本加算データ)、8 種類の幅 W 3-2 データ ~ 幅 W 3 6 データ (3 本加算データ)、7 種類の幅 W 4-2 データ ~ 幅 W 4 5 データ (4 本加算データ)、6 種類の幅 W 5-2 データ及び幅 W 5 4 データ (5 本加算データ) 及び、5 種類の幅 W 6-2 データ及び幅 W 6 3 データ (6 本加算データ) の合計で、3 9 種類の 1 本データ、2 ~ 6 本加算データ が得られる。

【 0 1 6 6 】

続いて、図 6 A ~ G を参照して、第 1 の実施例に係る 1 本データ、2 本 ~ 6 本加算データの他の算出例 (その 2) について説明する。この例では、対象キャラクタ C 1 の両側のキャラクタ C 0、C 2 まで加算範囲を拡張する場合であって、図 6 A に示す対象キャラクタ C 1 と隣接する 1 つ前のキャラクタ C 0 のスペース (6)、バー (5)、スペース (4)、バー (3) 及びスペース (2) の 5 エLEMENT まで加算範囲を拡張し、かつ、1 つ後ろのキャラクタ C 2 のバー (1)、スペース (2)、バー (3)、スペース (4) 及びバー (5) の 5 エLEMENT まで加算範囲を拡張する場合を前提とする。これらのキャラクタ C 0 ~ C 2 を有したバーコード記号 1 の場合を例に挙げて、対象キャラクタを作成する場合を説明する。

【 0 1 6 7 】

図 6 B に示す W 1 1 は対象キャラクタ C 1 の黒部分のバー (1) の幅である。バー (1) の幅はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 1 データを成す。W 1 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) の幅である。スペース (2) の幅もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 2 データを成す。W 1 3 はスペース (2) に続くバー (3) の幅である。バー (3) の幅はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 3 データを成す。

【 0 1 6 8 】

W 1 4 は対象キャラクタ C 1 のバー (3) に続くスペース (4) の幅である。スペース (4) の幅もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 4 データを成す。W 1 5 は対象キャラクタ C 1 のスペース (4) に続くバー (5) の幅である。バー (5) の幅はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 5 データを成す。W 1 6 は対象キャラクタ C 1 のバー (5) に続くスペース (6) の幅である。スペース (6) の幅もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 6 データを成す。上述の 6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データは 1 キャラクタの 1 本データであり、幅データ D 1 0 を構成する。

【 0 1 6 9 】

図 6 C に示す W 2-1 は、1 つ前のキャラクタ C 0 のスペース (6) と対象キャラクタ C 1 のバー (1) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W 2-1 データを成す。W 2 1 は 1 つ前のキャラクタ C 0 のスペース (6) に続くバー (1) とスペース (2) の各々が演算部 2 3 a によって加算された幅 (以下バー (1) + スペース (2) と記す) である。バー (1) + スペース (2) の加算幅を示すカウントデータ値が幅 W 2 1 データを成す。W 2 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) の加算幅を示すカウントデータ値が幅 W 2 2 データを成す。

【 0 1 7 0 】

W 2 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算された幅である。バー (3) + スペース (4) の加算幅を示すカウントデータ値が幅 W 2 3 データを成す。W 2 4 は対象キャラクタ C 1 のバー (3) に続くスペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。スペース (4) + バー (5) の加算幅を示すカウントデータ値が幅 W 2 4 データを成す。W 2 5 は対象キャラクタ C 1 のスペース (4) に続くバー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。バー (5) + スペース (6) の加算幅を示すカウントデータ値が幅 W 2 5 データを成す。W 2 6 は対象キャラクタ C 1 のバー (5) に続くスペース (6) 及び 1 つ後ろのキャラクタ C 2 のバー (1) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W 2 6 データを成す。上述の 7 種類の幅 W2-1 データ ~ 幅 W 2 6 データは 1 キャラクタの 2 本加算データであり、幅データ D 1 0 を構成する。

10

【 0 1 7 1 】

図 6 D に示す W3-2 は、 1 つ前のキャラクタ C 0 のバー (5) 、スペース (6) 及び、対象キャラクタ C 1 のバー (1) の各々が演算部 2 3 a によって加算された幅である。バー (5) + スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W3-2 データを成す。W3-1 は、 1 つ前のキャラクタ C 0 のスペース (6) 対象キャラクタ C 1 のバー (1) 及び、スペース (2) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) + スペース (2) の加算幅を示すカウントデータ値が幅 W3-1 データを成す。

20

【 0 1 7 2 】

W 3 1 は対象キャラクタ C 1 のバー (1) 、スペース (2) 及びバー (3) の各々が演算部 2 3 a によって加算された幅である。バー (1) + スペース (2) + バー (3) の加算幅を示すカウントデータ値が幅 W 3 1 データを成す。W 3 2 は対象キャラクタ C 1 のバー (1) に続くスペース (2) 、バー (3) 及びスペース (4) の各々が演算部 2 3 a によって加算された幅である。スペース (2) + バー (3) + スペース (4) の加算幅を示すカウントデータ値が幅 W 3 2 データを成す。

【 0 1 7 3 】

W 3 3 は対象キャラクタ C 1 のスペース (2) に続くバー (3) 、スペース (4) 及びバー (5) の各々が演算部 2 3 a によって加算された幅である。バー (3) + スペース (4) + バー (5) の加算幅を示すカウントデータ値が幅 W 3 3 データを成す。W 3 4 はバー (3) に続くスペース (4) 、バー (5) 及びスペース (6) の各々が演算部 2 3 a によって加算された幅である。スペース (4) + バー (5) + スペース (6) の加算幅を示すカウントデータ値が幅 W 3 4 データを成す。

30

【 0 1 7 4 】

W 3 5 は、対象キャラクタ C 1 のスペース (4) に続くバー (5) 、スペース (6) 、 1 つ後ろのキャラクタ C 2 のバー (1) の各々が演算部 2 3 a によって加算された幅である。バー (5) + スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W 3 5 データを成す。W 3 6 は、対象キャラクタ C 1 のバー (3) に続くスペース (6) の 1 つ後ろのキャラクタ C 2 のバー (1) 及びスペース (2) の各々が演算部 2 3 a によって加算された幅である。スペース (6) + バー (1) + スペース (2) の加算幅を示すカウントデータ値が幅 W 3 6 データを成す。上述の 8 種類の幅 W3-2 データ ~ 幅 W 3 6 データは 1 キャラクタの 3 本加算データであり、幅データ D 1 0 を構成する。

40

【 0 1 7 5 】

図 6 E に示す W4- 3 は、 1 つ前のキャラクタ C 0 のスペース (4) 、バー (5) 、スペース (6) 及び、バー (1) の各々が演算部 2 3 a によって加算された幅である。スペース (4) + バー (5) + スペース (6) + バー (1) の加算幅を示すカウントデータ値が幅 W4-3 データを成す。W4-2 は、 1 つ前のキャラクタ C 0 のバー (5) 、スペース (6) 、対象キャラクタ C 1 のバー (1) 及びスペース (2) の各々が演算部 2 3 a によって加

50

算された幅である。バー(5) + スペース(6) + バー(1) + スペース(2)の加算幅を示すカウントデータ値が幅W4-2データを成す。

【0176】

W4-1は、1つ前のキャラクタC0のスペース(6)、対象キャラクタC1のバー(1)、スペース(2)及びバー(3)の各々が演算部23aによって加算された幅である。スペース(6) + バー(1) + スペース(2) + バー(3)の加算幅を示すカウントデータ値が幅W4-1データを成す。W41は対象キャラクタC1のバー(1)、スペース(2)、バー(3)及びスペース(4)の各々が演算部23aによって加算された幅である。バー(1) + スペース(2) + バー(3) + スペース(4)の加算幅を示すカウントデータ値が幅W41データを成す。W42は対象キャラクタC1のバー(1)に続くスペース(2)、バー(3)、スペース(4)及びバー(5)の各々が演算部23aによって加算された幅である。スペース(2) + バー(3) + スペース(4) + バー(5)の加算幅を示すカウントデータ値が幅W42データを成す。

10

【0177】

W43は対象キャラクタC1のスペース(2)に続くバー(3)、スペース(4)、バー(5)及びスペース(6)の各々が演算部23aによって加算された幅である。バー(3) + スペース(4) + バー(5) + スペース(6)の加算幅を示すカウントデータ値が幅W43データを成す。W44は、対象キャラクタC1のバー(3)に続くスペース(4)、バー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された幅である。スペース(4) + バー(5) + スペース(6) + バー(1)の加算幅を示すカウントデータ値が幅W44データを成す。

20

【0178】

W45は、対象キャラクタC1の白のスペース(4)に続くバー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)及び、スペース(2)の各々が演算部23aによって加算された幅である。バー(5) + スペース(6) + バー(1) + スペース(2)の加算幅を示すカウントデータ値が幅W45データを成す。W46は、対象キャラクタC1のバー(5)に続くスペース(6)、1つ後ろのキャラクタC2のバー(1)、スペース(2)及びバー(3)の各々が演算部23aによって加算された幅である。スペース(6) + バー(1) + スペース(2) + バー(3)の加算幅を示すカウントデータ値が幅W46データを成す。上述の9種類の幅W4-3データ～幅W46データは1キャラクタの4本加算データであり、幅データD10を構成する。

30

【0179】

図6Fに示すW5-4は、1つ前のキャラクタC0のバー(3)、スペース(4)、バー(5)、スペース(6) + キャラクタC1のバー(1)の各々が演算部23aによって加算された幅である。バー(3) + スペース(4) + バー(5) + スペース(6) + バー(1)の加算幅を示すカウントデータ値が幅W5-4データを成す。W5-3は、1つ前のキャラクタC0のスペース(4)、バー(5)、スペース(6)、対象キャラクタC1のバー(1)及びスペース(2)の各々が演算部23aによって加算された幅である。スペース(4) + バー(5) + スペース(6) + バー(1) + スペース(2)の加算幅を示すカウントデータ値が幅W5-3データを成す。

40

【0180】

W5-2は、1つ前のキャラクタC0のバー(5)、スペース(6)、対象キャラクタC1のバー(1)、スペース(2)及びバー(3)の各々が演算部23aによって加算された幅である。バー(5) + スペース(6) + バー(1) + スペース(2) + バー(3)の加算幅を示すカウントデータ値が幅W5-2データを成す。W5-1は、1つ前のキャラクタC0のスペース(6)、対象キャラクタC1のバー(1)、スペース(2)、バー(3)及びスペース(4)の各々が演算部23aによって加算された幅である。スペース(6) + バー(1) + スペース(2) + バー(3) + スペース(4)の加算幅を示すカウントデータ値が幅W5-1データを成す。

【0181】

50

W5 1は対象キャラクタC 1のバー(1)、スペース(2)、バー(3)、スペース(4)及びバー(5)の各々が演算部2 3 aによって加算された幅である。バー(1) + スペース(2) + バー(3) + スペース(4) + バー(5)の加算幅を示すカウントデータ値が幅W5 1データを成す。W5 2は対象キャラクタC 1のバー(1)に続くスペース(2)、バー(3)、スペース(4)、バー(5)及びスペース(6)の各々が演算部2 3 aによって加算された幅である。スペース(2) + バー(3) + スペース(4) + バー(5) + スペース(6)の加算幅を示すカウントデータ値が幅W5 2データを成す。

【0 1 8 2】

W5 3は、対象キャラクタC 1のスペース(2)に続くバー(3)、スペース(4)、バー(5)、スペース(6)及び1つ後ろのキャラクタC 2のバー(1)の各々が演算部2 3 aによって加算された幅である。バー(3) + スペース(4) + バー(5) + スペース(6) + バー(1)の加算幅を示すカウントデータ値が幅W5 3データを成す。W5 4は、対象キャラクタC 1のバー(3)に続くスペース(4)、バー(5)、スペース(6)、1つ後ろのキャラクタC 2のバー(1)及びスペース(2)の各々が演算部2 3 aによって加算された幅である。スペース(4) + バー(5) + スペース(6) + バー(1) + スペース(2)の加算幅を示すカウントデータ値が幅W5 4データを成す。

【0 1 8 3】

W5 5は、対象キャラクタC 1のスペース(4)に続くバー(5)、スペース(6)、1つ後ろのキャラクタC 2のバー(1)、スペース(2)及び、バー(3)の各々が演算部2 3 aによって加算された幅である。バー(5) + スペース(6) + バー(1) + スペース(2) + バー(3)の加算幅を示すカウントデータ値が幅W5 5データを成す。W5 6は、対象キャラクタC 1のバー(5)に続くスペース(6)、1つ後ろのキャラクタC 2のバー(1)、スペース(2)、バー(3)及びスペース(4)の各々が演算部2 3 aによって加算された幅である。スペース(6) + バー(1) + スペース(2) + バー(3) + スペース(4)の加算幅を示すカウントデータ値が幅W5 6データを成す。上述の10種類の幅W5-4データ及び幅W5 6データは1キャラクタの5本加算データであり、幅データD 1 0を構成する。

【0 1 8 4】

図6 Gに示すW6-5は、1つ前のキャラクタC 0のスペース(2)、バー(3)、スペース(4)、バー(5)、スペース(6)及び、対象キャラクタC 1のバー(1)の各々が演算部2 3 aによって加算された幅である。スペース(2) + バー(3) + スペース(4) + バー(5) + スペース(6) + バー(1)の加算幅を示すカウントデータ値が幅W6-5データを成す。W6-4は、1つ前のキャラクタC 0のバー(3)、スペース(4)、バー(5)、スペース(6)、対象キャラクタC 1のバー(1)及びスペース(2)の各々が演算部2 3 aによって加算された幅である。バー(3) + スペース(4) + バー(5) + スペース(6) + バー(1) + スペース(2)の加算幅を示すカウントデータ値が幅W6-4データを成す。

【0 1 8 5】

W6-3は、1つ前のキャラクタC 0のスペース(4)、バー(5)、スペース(6)、対象キャラクタC 1のバー(1)、スペース(2)及びバー(3)の各々が演算部2 3 aによって加算された幅である。スペース(4) + バー(5) + スペース(6) + バー(1) + スペース(2) + バー(3)の加算幅を示すカウントデータ値が幅W6-3データを成す。W6-2は、1つ前のキャラクタC 0のバー(5) + スペース(6) + キャラクタC 1のバー(1) + スペース(2) + バー(3) + スペース(4)の各々が演算部2 3 aによって加算された幅である。バー(5) + スペース(6) + キャラクタC 1のバー(1) + スペース(2) + バー(3) + スペース(4)の加算幅を示すカウントデータ値が幅W6-2データを成す。

【0 1 8 6】

W6-1は、1つ前のキャラクタC 0のスペース(6) + キャラクタC 1のバー(1) + スペース(2) + バー(3) + スペース(4) + バー(5)の各々が演算部2 3 aによ

て加算された幅である。スペース(6) + キャラクタC1のバー(1) + スペース(2) + バー(3) + スペース(4) + バー(5)の加算幅を示すカウントデータ値が幅W6-1データを成す。

【0187】

W61は対象キャラクタC1のバー(1)、スペース(2)、バー(3)、スペース(4)、バー(5)及びスペース(6)の各々が演算部23aによって加算された幅である。バー(1) + スペース(2) + バー(3) + スペース(4) + バー(5) + スペース(6)の加算幅を示すカウントデータ値が幅W61データを成す。

【0188】

W62は対象キャラクタC1のバー(1)に続くスペース(2)、バー(3)、スペース(4)、バー(5)、スペース(6)及び、1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された幅である。スペース(2) + バー(3) + スペース(4) + バー(5) + スペース(6) + バー(1)の加算幅を示すカウントデータ値が幅W62データを成す。

10

【0189】

W63は対象キャラクタC1のスペース(2)に続くバー(3)、スペース(4)、バー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)及びスペース(2)の各々が演算部23aによって加算された幅である。バー(3) + スペース(4) + バー(5) + スペース(6) + バー(1) + スペース(2)の加算幅を示すカウントデータ値が幅W63データを成す。

20

【0190】

W64は、対象キャラクタC1のバー(3)に続くスペース(4)、バー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)、スペース(2)及びバー(3)の各々が演算部23aによって加算された幅である。スペース(4) + バー(5) + スペース(6) + 1つ後ろのキャラクタC2のバー(1) + スペース(2) + バー(3)の加算幅を示すカウントデータ値が幅W64データを成す。

【0191】

W65は、対象キャラクタC1のスペース(4)に続くバー(5)、スペース(6)、1つ後ろのキャラクタC2のバー(1)、スペース(2)、バー(3)及びスペース(4)の各々が演算部23aによって加算された幅である。バー(5) + スペース(6) + 1つ後ろのキャラクタC2のバー(1) + スペース(2) + バー(3) + スペース(4)の加算幅を示すカウントデータ値が幅W65データを成す。

30

【0192】

W66は、対象キャラクタC1のバー(5)に続くスペース(6)、1つ後ろのキャラクタC2のバー(1)、スペース(2)、バー(3)、スペース(4)及びバー(5)の各々が演算部23aによって加算された幅である。スペース(6) + 1つ後ろのキャラクタC2のバー(1) + スペース(2) + バー(3) + スペース(4) + バー(5)の加算幅を示すカウントデータ値が幅W66データを成す。上述の11種類の幅W6-5データ及び幅W66データは1キャラクタの6本加算データであり、幅データD10を構成する。

【0193】

この例では、図1に示したカウンタ部4から出力されるカウントデータ値に基づいて、6種類の幅W11データ～幅W16データ(1本データ)、7種類の幅W2-1データ～幅W26データ(2本加算データ)、8種類の幅W3-2データ～幅W36データ(3本加算データ)、9種類の幅W4-3データ～幅W46データ(4本加算データ)、10種類の幅W5-4データ及び幅W56データ(5本加算データ)及び、11種類の幅W6-5データ及び幅W66データ(6本加算データ)の合計で、51種類の幅W6-5データ及び幅W66データが得られる。

40

【実施例2】

【0194】

続いて、図7～図10を参照して、第2の実施例に係るデコード例について説明する。

50

ここで、図7A～Cを参照して、CODE 128のバーコード記号例、その欠損例及びその再構成例について説明をする。図7Aに示すCODE 128のバーコード記号1は、バーやスペース等の欠落や傷等を生じていないものである。バーコード記号1は8キャラクタから構成され、例えば、バーコードデータD12は、数字の「12345678」を表現する。

【0195】

図7Bに示すCODE 128のバーコード記号1'は、バーやスペース等に欠落や傷等が生じているものである。バーコード記号1'は図7Aに示したCODE 128のバーコード記号1に生じたものと仮定する。図中の3本の横向き破線は、閾値th1～th3である。各々の閾値th1～th3は、スキャン方向と直交する上・中・下段に設定されている。

10

【0196】

閾値th1を基準にしてバーコード記号1'をスキャンして、データを読み込むと、傷等によりキャラクタ「3」及び「4」が欠落している。閾値th2を基準にしてバーコード記号1'をスキャンしてデータを読み込むと、傷等によりキャラクタ「3」～「6」が欠落している。閾値th3を基準にしてバーコード記号1'をスキャンしてデータを読み込むと、傷等によりキャラクタ「7」及び「8」が欠落している。

【0197】

図7Cに示すように閾値th1を基準にしたデコード結果は、「12??5678」を示すバーコードデータD12が出力される。「3」及び「4」がデコードされていない。閾値th2を基準にしたデコード結果は、「12????78」を示すバーコードデータD12が出力される。「3」～「6」がデコードされていない。閾値th3を基準にしたデコード結果は、「123456??」を示すバーコードデータD12が出力される。「7」及び「8」がデコードされていない。

20

【0198】

第2の実施例では、図7Bに示したようにバーやスペース等に欠落や傷等が生じている場合であっても、「12??5678」や、「12????78」、「123456??」等を「12345678」を表現するバーコードデータD12に再構成するようになされる。

【0199】

この例では、スキャン方向と直交する上・中・下段に次々とスキャンを行って、エラーキャラクタを補正し、エラーキャラクタが無くなった時点でCD(チェックデジット)の照合を行って、バーコードデータD12を作成するようにした。これにより、かすれ、汚れ等により、エレメントの一部が欠落しているバーコード記号1であってもその読み取りが可能となる。

30

【0200】

また、第1の実施例では1キャラクタ内の幅データD10のみを利用した場合について説明したが、1キャラクタ内の端部の幅データD10が利用される回数が少ない。例えば、対象キャラクタC1のバー(1)の幅W11データは2本加算パターンの算出時、加算幅W21でのみ利用されるが、スペース(2)の幅W12データは、2本加算パターンの算出時、加算幅W21と加算幅W22とで利用されている。そこで、第2の実施例では当該キャラクタC1に隣接するキャラクタC0、C2も含めたエレメント部分も含めて計算を行うようにした。

40

【0201】

続いて、図8～図10を参照して、第2の実施例に係る難読キャラクタのデコード例(その1～3)について説明する。この実施例では、パターン認識する加算範囲を両側の2キャラクタのエレメントまで拡張した。加算範囲は、図7Aに示した対象キャラクタと隣接する1つ前のキャラクタC0のスペース(6)、バー(5)、スペース(4)、バー(3)及びスペース(2)の5エレメントまで拡張し、かつ、1つ後ろのキャラクタC2のバー(1)、スペース(2)、バー(3)、スペース(4)及びバー(5)の5エレメントまで拡張する場合を前提とする。この加算範囲の拡張によって、51種類の1本データ

50

、 2 ～ 6 本加算データからデコード結果を得る場合を前提とする（マルチスキャンリコンストラクション）。

【 0 2 0 2 】

これらをデコード条件にして、図 8 に示すフローチャートのステップ S T 21 で C P U 2 3 はスタートマージンを検索する。このとき、光学読取部 2 は、第 1 の実施例と同様にし、バーコード記号 1 を読み取って、コード読取データ D 2 を発生する。コード読取データ D 2 は、光学読取部 2 からカウンタ部 4 へ出力される。クロック発生部 3 は、所定周波数のサンプリング用の C L K 信号をカウンタ部 4 に出力する。

【 0 2 0 3 】

カウンタ部 4 は光学読取部 2 から出力されるコード読取データ D 2 を入力すると共に、クロック発生部 3 から出力される C L K 信号を入力し、C L K 信号に基づいてコード読取データ D 2 をカウントし、コード読取データ D 2 を白部分のスペースおよび黒部分のバーを含むエレメントの 1 本幅及び 2 本加算幅を示すカウントデータ D 4 に変換する。カウントデータ D 4 は、バーコード記号 1 における各バー（黒部分）の幅およびスペース（白部分）の幅を示す数値データ列となる。カウントデータ D 4 は、幅データ格納部 1 0 に保存される。なお、スタートマージンの検索については、その説明を省略する。

【 0 2 0 4 】

ステップ S T 22 で C P U 2 3 は、1 キャラクタ分の幅データ（6 本）を抽出する。図 1 5 に示したキャラクタ C 1 によれば、黒部分のバー（1）の幅 W 1 1 がカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 1 データとなる。バー（1）に続く白部分のスペース（2）の幅 W 1 2 もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 2 データとなる。スペース（2）に続くバー（3）の幅 W 1 3 はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 3 データとなる。

【 0 2 0 5 】

このバー（3）に続くスペース（4）の幅 W 1 4 もカウンタ部 4 により計数され、そのカウントデータ値は幅 W 1 4 データとなる。スペース（4）に続くバー（5）の幅 W 1 5 はカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 5 データとなる。バー（5）に続くスペース（6）の幅 W 1 6 もカウンタ部 4 により計数され、そのカウントデータ値が幅 W 1 6 データとなる。上述の 6 種類の幅 W 1 1 データ～幅 W 1 6 データはバーコードデータ値 D 3 0 を構成する。バーコードデータ値 D 3 0 はインターフェース部 1 1 を介して C P U 2 3 に出力される。

【 0 2 0 6 】

ステップ S T 23 で C P U 2 3 はバーの分割、消失の有無に基づいて制御を分岐する。バーの分割、消失の有無については、6 本の幅の合計と 1 キャラクタ幅とを比較し、1 キャラクタ分の幅データ D 1 0 が適切かを判断する。ここでは、1 キャラクタ内で隣接するバー及びスペースを含む 6 本のエレメントの幅を加算した合計値と、1 キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて 1 キャラクタ分の幅データ D 1 0 が適切か否かを判別する。

【 0 2 0 7 】

この例で、C P U 2 3 では比較部 2 3 c が 1 キャラクタ内で隣接する 6 本のバー（1）、スペース（2）、バー（3）、スペース（4）、バー（5）及びスペース（6）の幅 W 1 ～ W 6 の合計と、1 キャラクタの全幅を示す全幅値とを比較する。比較部 2 3 c による比較結果によって、1 キャラクタ分の幅データ D 1 0 が適切か否かが判別でき、この判別に基づいてキャラクタのバーの消失や分割等を判断できるようになる。また、複数のスキャンのデコード結果を組み合わせることが可能となり、汚れや、かすれがひどいバーコードにおいても読み取りが可能となる。

【 0 2 0 8 】

また、当該キャラクタ C 1 の前後のキャラクタ C 0 , C 2 の全幅からキャラクタフレームを判定できるので、ボイドやスポット等によるエレメントに過不足が有っても、デコードを継続できるようになる。また、バーの消失や分割等の判断結果に基づいて対象キャラ

10

20

30

40

50

クタを絞ることができ、計算量の軽減が図れ、処理時間の高速化につながる。

【0209】

バーの分割、消失が無い場合は、ステップS T24でCPU23は1モジュール幅を算出し、1キャラクタ幅を記憶する。このとき、演算部23aは、6種類の幅W11データ～幅W16データに基づいて当該キャラクタC1の1モジュール幅を算出する。1キャラクタ幅は、図7Bに示した1キャラクタ内で隣接する6本のバー(1)、スペース(2)、バー(3)、スペース(4)、バー(5)及びスペース(6)の幅W1～W6を合計して求められる。

【0210】

ステップS T25でCPU23は1キャラクタ分の幅データD10を2本加算パターンに変換する。例えば、図7Cに示した1つ前のキャラクタC0のスペース(6)と対象キャラクタC1のバー(1)の各々が演算部23aによって加算された加算幅W2-1を示すカウントデータ値が幅W2-1データとなる。

10

【0211】

同様にして、1つ前のキャラクタC0のスペース(6)に続くバー(1)とスペース(2)の各々が演算部23aによって加算された加算幅W21を示すカウントデータ値が幅W21データとなる。対象キャラクタC1のバー(1)に続くスペース(2)及びバー(3)の各々が演算部23aによって加算された加算幅W22を示すカウントデータ値が幅W22データとなる。

20

【0212】

対象キャラクタC1のスペース(2)に続くバー(3)及びスペース(4)の各々が演算部23aによって加算された加算幅W23を示すカウントデータ値が幅W23データとなる。対象キャラクタC1のバー(3)に続くスペース(4)及びバー(5)の各々が演算部23aによって加算された加算幅W24を示すカウントデータ値が幅W24データとなる。

【0213】

対象キャラクタC1のスペース(4)に続くバー(5)及びスペース(6)の各々が演算部23aによって加算された加算幅W25を示すカウントデータ値が幅W25データとなる。対象キャラクタC1のバー(5)に続くスペース(6)及び1つ後ろのキャラクタC2のバー(1)の各々が演算部23aによって加算された加算幅W26を示すカウントデータ値が幅W26データとなる。上述の7種類の幅W2-1データ～幅W26データは1キャラクタの2本加算データであり、幅データD10から構成される。

30

【0214】

その後、ステップS T26でCPU23は、先に変換した2本加算パターンが一致するキャラクタを検索する。この例では、図1に示したカウンタ部4から出力されるカウントデータ値に基づいて、検索部23dが、6種類の幅W11データ～幅W16データ(1本データ)、7種類の幅W2-1データ～幅W26データ(2本加算データ)、8種類の幅W3-2データ～幅W36データ(3本加算データ)、9種類の幅W4-3データ～幅W46データ(4本加算データ)、10種類の幅W5-4データ及び幅W56データ(5本加算データ)及び、11種類の幅W6-5データ及び幅W66データ(6本加算データ)の51種類の1本データ、2～6本加算データC1に対して一致するキャラクタを検索する。その際に各エレメントの幅に対して2種類の対象キャラクタのどちらかと一致する対象キャラクタを検索し、対象キャラクタを絞り込む。この絞り込みにより、およそ1～4程度の候補キャラクタが検索される。

40

【0215】

ステップS T27でCPU23は一致するキャラクタの存在有無に基づいて制御を分岐する。一致するキャラクタが存在する場合は、ステップS T28で2本加算パターンが一致した対象キャラクタをデコード結果とする。その後、図9に示すステップS T31に移行する。

【0216】

50

なお、ステップ S T 23 でバーの分割、消失が有る場合、及び、ステップ S T 27 で一致するキャラクタが存在しない場合は、ステップ S T 29 で C P U 2 3 は現在のキャラクタをエラーキャラクタとする。その後、ステップ S T 30 で C P U 2 3 は次のキャラクタ (1 キャラクタ幅分後ろのデータ) のデコード処理に移行する。その後、ステップ S T 22 に移行して、上述した内容を繰り返す。

【 0 2 1 7 】

上述のステップ S T 28 でデコード結果となされた候補キャラクタについては、図 9 に示すステップ S T 31 で C P U 2 3 はマージン直後のキャラクタか否かに基づいて制御を分岐する。デコード結果となされた候補キャラクタが、マージン直後のキャラクタではない場合は、ステップ S T 32 に移行して C P U 2 3 はデコード結果がストップキャラクタか否かに基づいて制御を分岐する。デコード結果がストップキャラクタである場合は、ステップ S T 33 に移行して C P U 2 3 は、デコード結果にエラーキャラクタが存在しないか否かに基づいて制御を分岐する。

10

【 0 2 1 8 】

デコード結果にエラーキャラクタが存在しない場合は、ステップ S T 34 に移行して、C P U 2 3 はデコードバッファ 2 5 に現在 (今) のスキャンのデコード結果を上書きする。その後、ステップ S T 35 に移行して C P U 2 3 は誤り検知 (パリティチェック) を実行して、チェックデジットが正しいか否かに基づいて制御を分岐する。チェックデジットが正しい場合は、ステップ S T 36 で C P U 2 3 は読み取りを完了し、デコードバッファ 2 5 のバーコードデータ D 1 2 を出力し、デコードバッファ 2 5 をクリアする (正常終了) 。その後、ステップ S T 45 に移行する。

20

【 0 2 1 9 】

なお、ステップ S T 31 でマージン直後のキャラクタである場合は、ステップ S T 37 に移行して C P U 2 3 は、デコード結果がスタートキャラクタか否かに基づいて制御を分岐する。上述のステップ S T 37 でデコード結果がスタートキャラクタでない場合及び、ステップ S T 32 でデコード結果がストップキャラクタでない場合は、ステップ S T 38 に移行する。ステップ S T 38 で C P U 2 3 は、次キャラクタ (6 本後ろの幅データ D 1 0) のデコード処理を実行する。その後、ステップ S T 22 に戻る。

【 0 2 2 0 】

また、ステップ S T 33 でデコード結果にエラーキャラクタが存在している場合は、図 1 0 に示すステップ S T 39 で C P U 2 3 はデコードバッファ 2 5 の長さでデコードデータ長とが等しいか否かに基づいて制御を分岐する。デコードバッファ 2 5 の長さでデコードデータ長とが等しい場合は、ステップ S T 41 で、デコードバッファ 2 5 のエラーキャラクタ部分のデコードが成功していたらデータを書き換える。その後、ステップ S T 42 に移行して、デコードバッファ 2 5 にエラーキャラクタが存在しないか否かに基づいて制御を分岐する。デコードバッファ 2 5 にエラーキャラクタが存在していない場合は、ステップ S T 35 に戻る。

30

【 0 2 2 1 】

上述のステップ S T 35 でチェックデジットが誤っている場合は、図 1 0 に示すステップ S T 43 でデコードバッファ 2 5 をクリアした後、図 9 に示したステップ S T 37 でデコード結果がスタートキャラクタではない場合、図 1 0 に示したステップ S T 39 でデコードバッファ 2 5 の長さでデコードデータ長とが異なる場合は、ステップ S T 40 でデコードバッファ 2 5 をクリアした後、及び、ステップ S T 42 でデコードバッファ 2 5 にエラーキャラクタが存在する場合は、ステップ S T 44 に移行して C P U 2 3 は、読み取りエラー処理を実行する。読み取りエラー処理では次のマージンを検索する等が行われる。

40

【 0 2 2 2 】

また、キャラクタ認識がエラーとなった場合に、エラーキャラクタを仮に配置してバーコードデータを作成するようにしてもよい。その後、図 9 に示すステップ S T 45 に移行して、C P U 2 3 は終了を判別する。例えば、当該バーコード記号読取装置 1 0 0 に対して読取終了指令が検出され、読取終了指令が検出された場合は、バーコード記号 1 の読み取

50

り制御を終了する。読取終了指令が検出されていない場合は、バーコード記号 1 の読み取り制御を継続するべく、ステップ S T 21 に戻って、上述した内容を繰り返すようになされる。

【 0 2 2 3 】

このように第 2 の実施例に係るバーコード記号読取装置 1 0 0 によれば、パターン認識する加算範囲を対象キャラクタの両側の 2 キャラクタ C 0 , C 2 のエレメントまで拡張した。すなわち、加算範囲に関して、図 7 A に示した対象キャラクタと隣接する 1 つ前のキャラクタ C 0 のスペース (6)、バー (5)、スペース (4)、バー (3) 及びスペース (2) の 5 エレメントまで拡張し、かつ、1 つ後ろのキャラクタ C 2 のバー (1)、スペース (2)、バー (3)、スペース (4) 及びバー (5) の 5 エレメントまで拡張したので、5 1 種類の対象キャラクタの中からデコード結果となる候補キャラクタを決定できるようになる。

10

【 0 2 2 4 】

従って、キャラクタ内の幅データだけでパターン認識を行う従来方式に比べて、対象キャラクタの両側の 2 キャラクタ C 0 , C 2 のエレメントも含めた部分についても計算に参画されるので、対象キャラクタの端部に、欠落、傷等の誤差が多く含まれるパターンであっても、誤読し難くなった。これにより、バー、スペース等のエレメント (エレメント) の幅の最大誤差が ± 0.5 モジュール以上あっても、キャラクタとして認識できる可能性が高くなった。しかも、対象キャラクタを絞ることによって、計算量の軽減が図れ、パターン認識処理に係る時間が短縮され、データ処理の高速化につながる。

20

【 実施例 3 】

【 0 2 2 5 】

続いて、図 1 1 ~ 図 1 3 を参照して、第 3 の実施例に係る難読キャラクタのデコード例 (その 1 ~ 3) について説明する。この実施例では、第 1 の実施例で説明したキャラクタデコードアルゴリズムと、第 2 の実施例で説明したマルチスキャンリコンストラクションとを併合した処理が行われる。これらをデコード条件にして、図 1 1 に示すフローチャートのステップ S T 51 でスタートマージンを検索する。このとき、光学読取部 2 は、第 1 の実施例と同様にして、バーコード記号 1 を読み取って、コード読取データ D 2 を発生する。コード読取データ D 2 は、光学読取部 2 からカウンタ部 4 へ出力される。クロック発生部 3 は、所定周波数のサンプリング用の C L K 信号をカウンタ部 4 に出力する。

30

【 0 2 2 6 】

カウンタ部 4 は光学読取部 2 から出力されるコード読取データ D 2 を入力すると共に、クロック発生部 3 から出力される C L K 信号を入力し、C L K 信号に基づいてコード読取データ D 2 をカウントし、コード読取データ D 2 を白部分のスペースおよび黒部分のバーを含むエレメントの 1 本幅及び 2 本加算幅を示すカウントデータ D 4 に変換する。カウントデータ D 4 は、バーコード記号 1 における各バー (黒部分) の幅およびスペース (白部分) の幅を示す数値データ列となる。カウントデータ D 4 は、幅データ格納部 1 0 に保存される。なお、スタートマージンの検索については、その説明を省略する。

【 0 2 2 7 】

ステップ S T 52 で C P U 2 3 は 6 本の幅の合計と 1 キャラクタ幅とを比較し、1 キャラクタ分の幅データ D 10 が適切かを判断する。ここでは、1 キャラクタ内で隣接するバー及びスペースを含む 6 本のエレメントの幅を加算した合計値と、1 キャラクタの全幅を示す全幅値とを比較し、比較結果に基づいて 1 キャラクタ分の幅データ D 10 が適切か否かを判別する。

40

【 0 2 2 8 】

この例で、C P U 2 3 では比較部 2 3 c が 1 キャラクタ内で隣接する 6 本のバー (1)、スペース (2)、バー (3)、スペース (4)、バー (5) 及びスペース (6) の幅 W 1 ~ W 6 の合計と、1 キャラクタの全幅を示す全幅値とを比較する。比較部 2 3 c による比較結果によって、1 キャラクタ分の幅データ D 10 が適切か否かが判別でき、この判別に基づいてキャラクタ候補を絞ることができ、計算量の軽減が図れ、処理時間の高速化に

50

つながる。

【 0 2 2 9 】

ステップ S T 53 で C P U 2 3 はバーの分割、消失の有無に基づいて制御を分岐する。バーの分割、消失の有無については、比較部 2 3 c による比較結果を利用する。対象キャラクタ C 1 の前後のキャラクタ C 0 , C 2 の全幅からキャラクタフレームを判定できるので、ボイドやスポット等によるエレメントに過不足が有っても、デコードを継続できるようになる。

【 0 2 3 0 】

バーの分割、消失が無い場合は、ステップ S T 54 で C P U 2 3 は 1 モジュール幅を算出し、1 キャラクタ幅を記憶する。このとき、演算部 2 3 a は、6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データに基づいて対象キャラクタ C 1 の 1 モジュール幅を算出する。1 キャラクタ幅は、図 7 B に示した 1 キャラクタ内で隣接する 6 本のバー (1)、スペース (2)、バー (3)、スペース (4)、バー (5) 及びスペース (6) の幅 W 1 ~ W 6 を合計して求められる。

10

【 0 2 3 1 】

ステップ S T 55 で C P U 2 3 は抽出された 2 1 種類の幅データ D 1 0 を 1 ~ 6 本加算パターン (エレメント幅) に変換する。この例では、6 種類の幅 W 1 1 データ ~ 幅 W 1 6 データが 1 キャラクタの 1 本データに変換され、1 本データに基づく対象キャラクタ C 1 の被認識パターンとして、1 本パターンが作成される。5 種類の幅 W 2 1 データ ~ 幅 W 2 5 データは 1 キャラクタの 2 本加算データに変換され、2 本加算データに基づく対象キャラクタ C 1 の 2 本加算パターン (被認識パターン) が作成される。4 種類の幅 W 3 1 データ ~ 幅 W 3 4 データは 1 キャラクタの 3 本加算データに変換され、3 本加算データに基づく対象キャラクタ C 1 の 3 本加算パターンが作成される。

20

3 種類の幅 W 4 1 データ ~ 幅 W 4 3 データは 1 キャラクタの 4 本加算データに変換され、4 本加算データに基づく対象キャラクタ C 1 の 4 本加算パターンが作成される。2 種類の幅 W 5 1 データ及び幅 W 5 2 データは 1 キャラクタの 5 本加算データに変換され、5 本加算データに基づく対象キャラクタの 5 本加算パターンが作成される。1 種類の幅 W 6 1 データは 1 キャラクタの 6 本加算データに変換され、6 本加算データに基づく対象キャラクタ C 4 の 6 本加算パターンが作成される。

【 0 2 3 2 】

30

ステップ S T 56 で C P U 2 3 はキャラクタを絞り込むために、1 本パターン、2 ~ 6 本加算パターンの全てに対してモジュール数の誤差が 1 以内のキャラクタを検索して候補キャラクタを決定する。このとき、C P U 2 3 では抽出部 2 3 b が、演算部 2 3 a によって求められた 1 本パターン、2 ~ 6 本加算パターンの各々のモジュール数が誤差 1 以内の整数となる対象キャラクタ C 1 等を絞り込んで候補キャラクタを抽出する。その際に、抽出部 2 3 b は、演算部 2 3 a によって求められた 1 本パターン、2 ~ 6 本加算パターンの 1 キャラクタ分の 6 本の幅データ D 1 0 と、対象キャラクタの前後の (6 - 1) 本の幅データ D 1 0 とを抽出する。

【 0 2 3 3 】

ステップ S T 57 で C P U 2 3 は、候補キャラクタが 1 つ以上存在するか否かに基づいて制御を分岐する。候補キャラクタが 1 つ以上 (複数) 存在する場合は、ステップ S T 58 で 1 ~ 6 本加算パターンの実測値とその理想値との間の誤差を算出し、この誤差の合計が最小となった候補キャラクタをデコード結果とする。この誤差値が最小となる候補キャラクタは、期待値キャラクタと最も相関が強いキャラクタである。この例では、総合的な誤差が最も少ない対象キャラクタ C 1 等を候補キャラクタとして選択できるようになる。

40

【 0 2 3 4 】

このとき、C P U 2 3 では、期待値キャラクタ (理論値) が期待値格納部 2 4 から C P U 2 3 の比較部 2 3 c へ読み出される。比較部 2 3 c は、抽出部 2 3 b によって抽出された各々の 2 ~ 6 本加算パターン (エレメント幅) のモジュール数の候補キャラクタと、候補キャラクタを評価するための期待値キャラクタ (理論値) とを比較する。この比較結果

50

は検索部 2 3 d へ出力される。

【 0 2 3 5 】

検索部 2 3 d では、候補キャラクタと最も相関が強い期待値キャラクタが検索される。例えば、上述の演算部 2 3 a が候補キャラクタのモジユール数の実測値と、期待値キャラクタのモジユール数の理想値との間の誤差を算出すると、検索部 2 3 d が、演算部 2 3 a によって算出された誤差を合計して得られる誤差値が最小となる候補キャラクタと最も相関が強い期待値キャラクタを検索する。

【 0 2 3 6 】

この例では、誤差値が最小となる候補キャラクタと最も相関が強い期待値キャラクタを検索するので、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。なお、ステップ S T 57 で候補キャラクタが存在しない場合は、ステップ S T 59 で現在のキャラクタをエラーキャラクタとする。その後、ステップ S T 60 で次のキャラクタ（キャラクタ幅分後ろのデータ）のデコード処理に移行する。

【 0 2 3 7 】

上述のステップ S T 58 でデコード結果となされた候補キャラクタについては、図 1 2 に示すステップ S T 61 で C P U 2 3 はマージン直後の対象キャラクタが否かに基づいて制御を分岐する。デコード結果となされた候補キャラクタが、マージン直後の対象キャラクタではない場合は、ステップ S T 62 に移行して C P U 2 3 はデコード結果がストップキャラクタが否かに基づいて制御を分岐する。デコード結果がストップキャラクタである場合は、ステップ S T 63 に移行して C P U 2 3 は、デコード結果にエラーキャラクタが存在しないか否かに基づいて制御を分岐する。

【 0 2 3 8 】

デコード結果にエラーキャラクタが存在しない場合は、ステップ S T 64 に移行して、C P U 2 3 はデコードバッファ 2 5 に現在（今）のスキヤンのデコード結果を上書きする。その後、ステップ S T 65 に移行して C P U 2 3 は誤り検知（パリティチェック）を実行して、チェックデジットが正しいか否かに基づいて制御を分岐する。チェックデジットが正しい場合は、ステップ S T 66 で C P U 2 3 は読み取りを完了し、デコードバッファ 2 5 のバーコードデータ D 1 2 を出力し、デコードバッファ 2 5 をクリアする（正常終了）。その後、ステップ S T 65 に移行する。

【 0 2 3 9 】

なお、ステップ S T 61 でマージン直後の対象キャラクタである場合は、ステップ S T 67 に移行して C P U 2 3 は、デコード結果がスタートキャラクタが否かに基づいて制御を分岐する。上述のステップ S T 67 でデコード結果がスタートキャラクタである場合及び、ステップ S T 62 でデコード結果がストップキャラクタである場合は、ステップ S T 68 に移行する。ステップ S T 68 で C P U 2 3 は、次キャラクタ（6 本後ろの幅データ D 1 0 ）のデコード処理を実行する。その後、ステップ S T 52 に戻る。

【 0 2 4 0 】

また、ステップ S T 63 でデコード結果にエラーキャラクタが存在している場合は、図 1 3 に示すステップ S T 69 で C P U 2 3 はデコードバッファ 2 5 の長さでデコードデータ長とが等しいか否かに基づいて制御を分岐する。デコードバッファ 2 5 の長さでデコードデータ長とが等しい場合は、ステップ S T 71 で、デコードバッファ 2 5 のエラーキャラクタ部分のデコードが成功していたらデータを書き換える。その後、ステップ S T 72 に移行して、デコードバッファ 2 5 にエラーキャラクタが存在しないか否かに基づいて制御を分岐する。デコードバッファ 2 5 にエラーキャラクタが存在していない場合は、ステップ S T 65 に戻る。

【 0 2 4 1 】

上述のステップ S T 65 でチェックデジットが誤っている場合は、図 1 3 に示すステップ S T 73 でデコードバッファ 2 5 をクリアした後、図 1 2 に示したステップ S T 67 でデコード結果がスタートキャラクタではない場合、図 1 3 に示したステップ S T 69 でデコードバッファ 2 5 の長さでデコードデータ長とが異なる場合は、ステップ S T 70 でデコードバッ

ファ25をクリアした後、及び、ステップS T72でデコードバッファ25にエラーキャラクタが存在する場合は、ステップS T74に移行してCPU23は、読み取りエラー処理を実行する。読み取りエラー処理では次のマージンを検索する等が行われる。

【0242】

また、キャラクタ認識がエラーとなった場合に、エラーキャラクタを仮に配置してバーコードデータを作成するようにしてもよい。その後、図12に示すステップS T75に移行して、CPU23は終了を判別する。例えば、当該バーコード記号読取装置100に対して読取終了指令が検出され、読取終了指令が検出された場合は、バーコード記号1の読み取り制御を終了する。読取終了指令が検出されていない場合は、バーコード記号1の読み取り制御を継続するべく、ステップS T51に戻って、上述した内容を繰り返すようになさ

10

【0243】

このように、第3の実施例に係るバーコード記号読取装置100によれば、バーコード記号1を読み取ってそのデコード結果を出力する場合に、第1の実施例で説明したキャラクタデコードアルゴリズムと、第2の実施例で説明したマルチスキャンリコンストラクションとを併合した処理が行われる。これを前提にして、検索部23dが抽出部23bによって抽出された各々の2～6本加算パターンのモジュール数の候補キャラクタと、当該候補キャラクタを評価するための期待値キャラクタとを比較し、当該候補キャラクタと最も相関が強い期待値キャラクタを検索するようになる。

【0244】

20

この検索によって、複数の対象キャラクタから抽出された候補キャラクタと最も相関が強い期待値キャラクタのバーコードデータD12を当該バーコード記号1の読取時のデコード結果として出力できるようになるので、従来方式のバーコード読み取り方法に比べて大幅にバーコード記号1の誤読取を防止できるようになる。

【0245】

しかも、誤差の範囲を従来方式の ± 0.5 から ± 1.0 に拡大し、対象キャラクタを増加させている。そして、1キャラクタ分の6本の幅データD10と、対象キャラクタの前後の5本の幅データD10とを抽出するので、2～6本加算パターンの各々のモジュール数が誤差1以内の整数となる対象キャラクタを絞り込んだ候補キャラクタを従来方法に比べてより多く抽出できるようになる。また、候補キャラクタを絞ることによって、計算量の軽減が図れ、パターン認識処理に係る時間が短縮され、データ処理の高速化につながる。これにより、従来方式のバーコード読み取り方法に比べて大幅にバーコードの誤読取を防止できるようになる。なお、誤差の範囲を設定する際の閾値を可変することで、幅データD10の信頼性を選択することも可能となる。

30

【0246】

この例では、バーコード記号の規格に関して、CODE128の場合について説明したが、これに限られることはなく、CODE39、EAN/UPC、インターリーブド2オブ5、コーダバー、RSS、RSSリミテッド、RSSエクスパンデド等の各々の方式についても、本発明を同様にして適用できるようになる。

【産業上の利用可能性】

40

【0247】

本発明は、バーコード記号を読み取ってバーコードデータを出力するバーコード記号読取装置に適用して極めて好適である。

【符号の説明】

【0248】

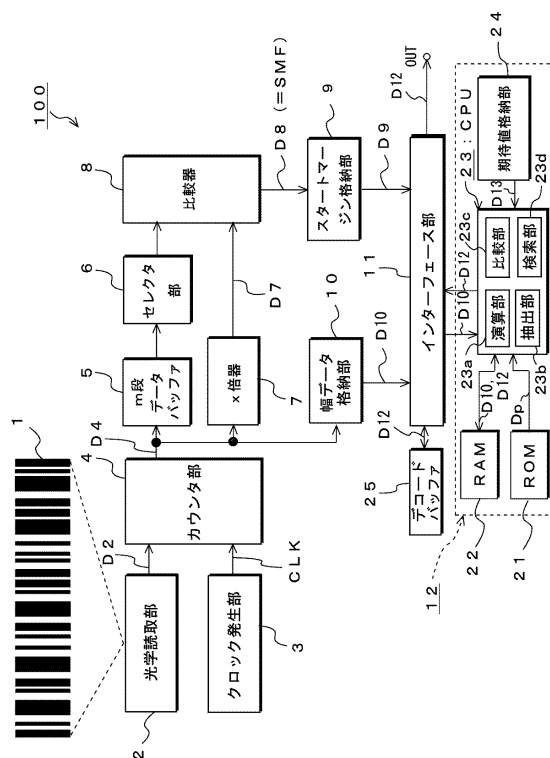
2・・・光学情報読取部、3・・・クロック発生部、4・・・カウンタ部、5・・・m段データバッファ、6・・・セレクタ部、7・・・x倍器、8・・・比較器、・・・スタートマージン格納部、10・・・幅データ格納部、11・・・インターフェース部、12・・・マイクロプロセッサ、21・・・ROM、22・・・RAM、23・・・CPU、23a・・・演算部（演算手段）、23b・・・抽出部（抽出手段）、23c・・・

50

比較部、23d・・・検索部、24・・・期待値格納部、25・・・デコードバッファ及び、100・・・バーコード記号読取装置（光学情報読取装置）

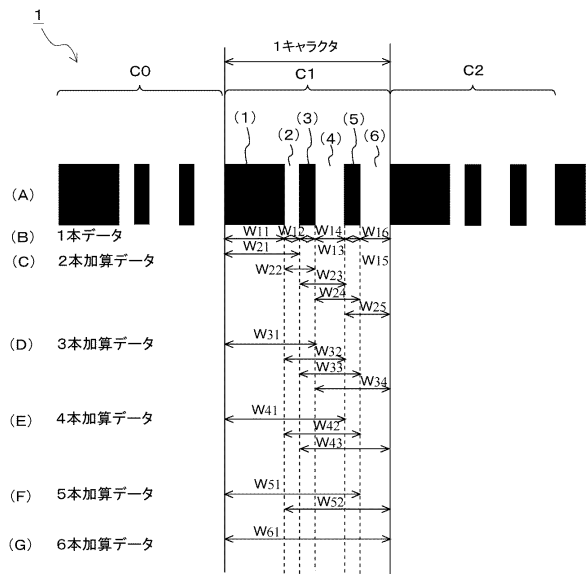
【図1】

実施形態としてのバーコード記号読取装置100の構成例



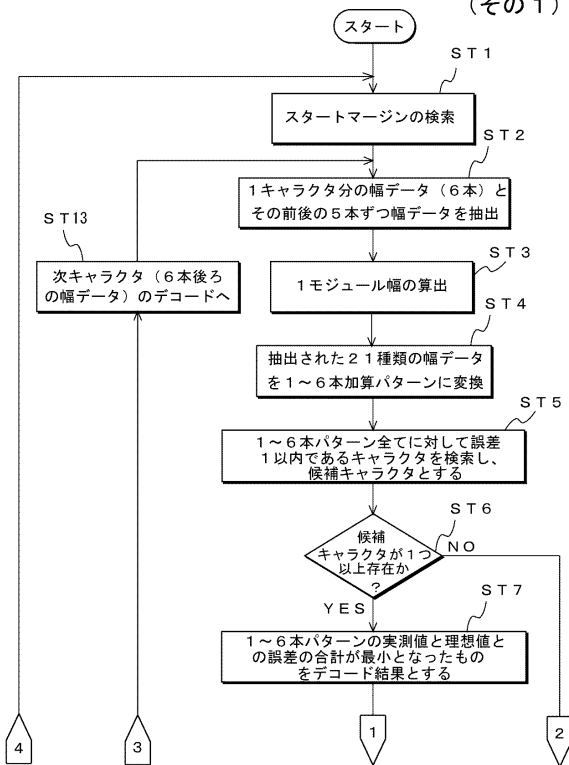
【図2】

第1の実施例に係る1本、2～6本加算データの算出例



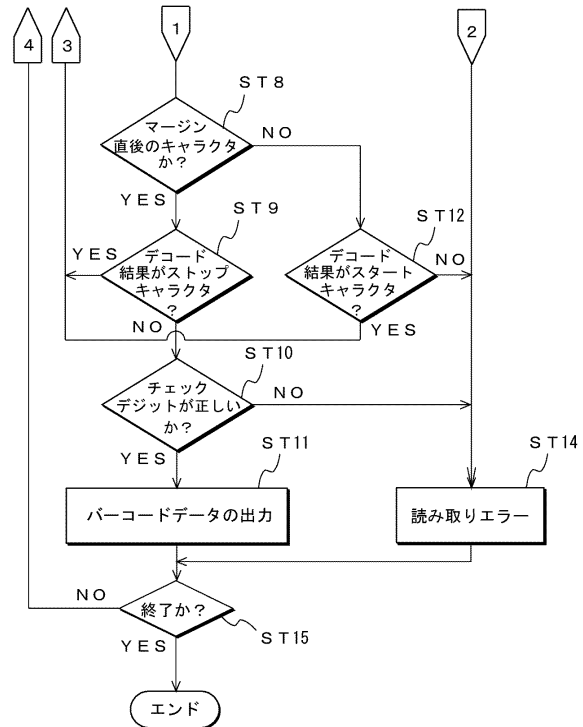
【図 3】

第 1 の実施例に係る難読キャラクタの解釈例
(その 1)



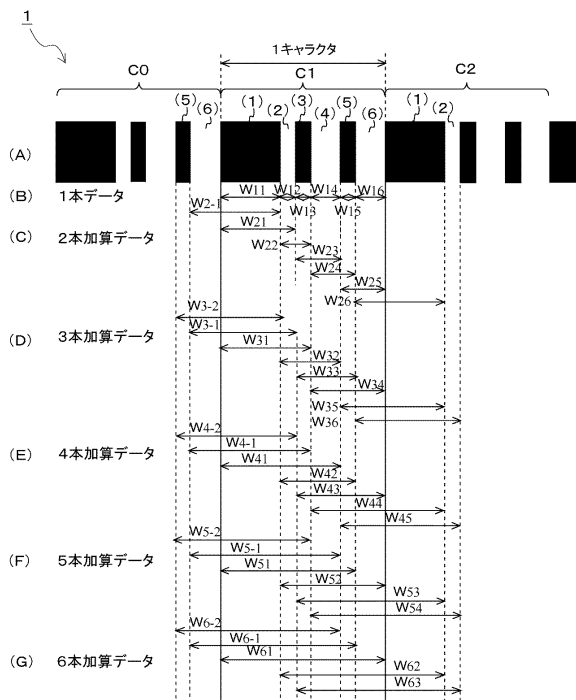
【図 4】

第 1 の実施例に係る難読キャラクタ
の解釈例 (その 2)



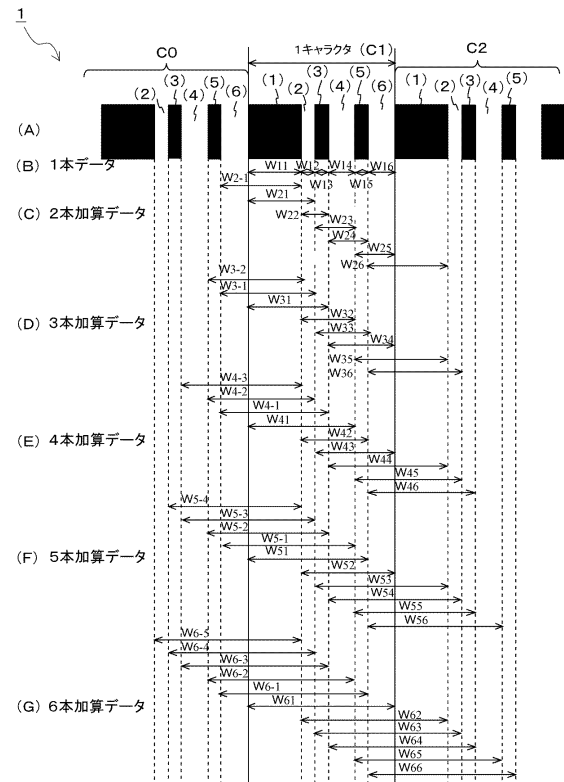
【図 5】

1 本データ、2 ~ 6 本加算データの他の算出例 (その 1)



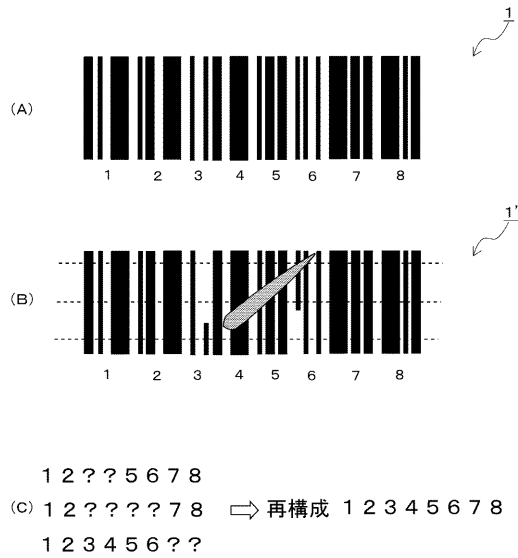
【図 6】

1 本データ、2 ~ 6 本加算データの他の算出例 (その 2)

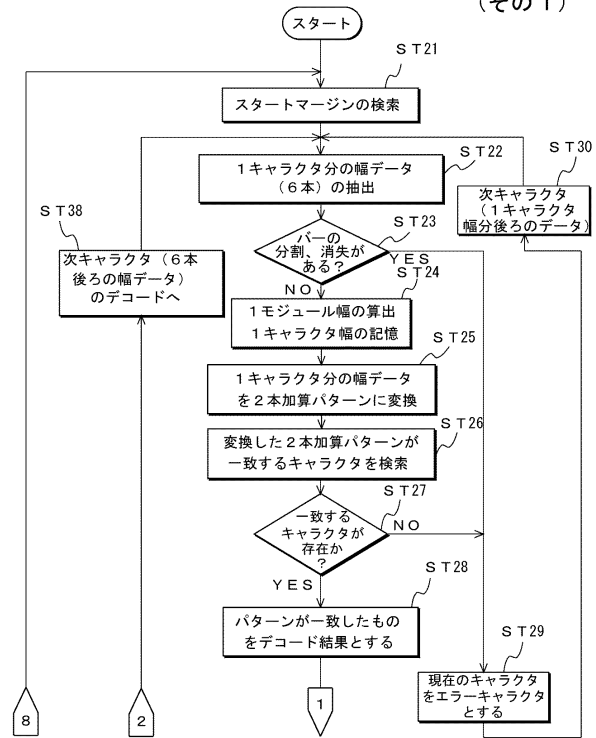


【図 7】

第2の実施例に係るバーコードデータの再生例

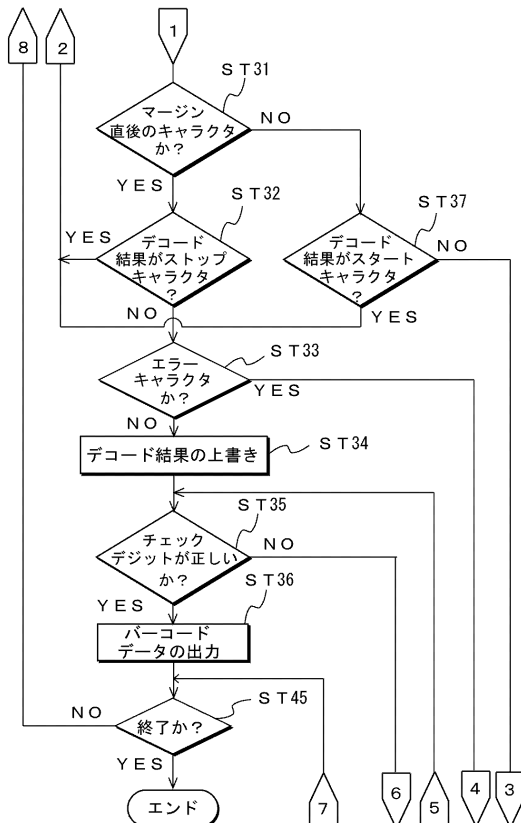


【図 8】

第2の実施例に係る難読キャラクタの解読例
(その 1)

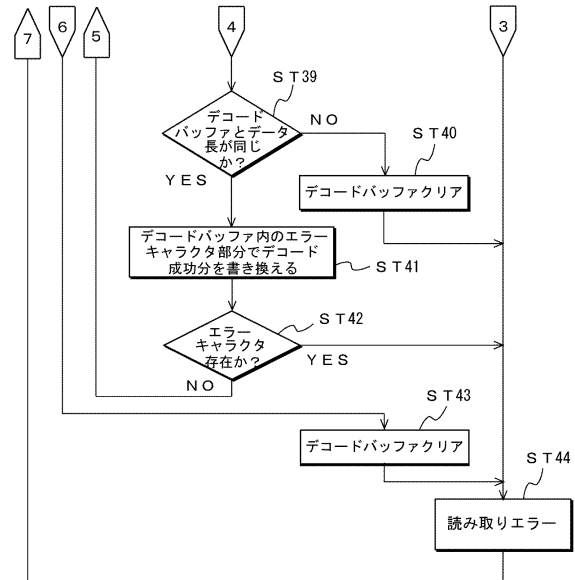
【図 9】

第2の実施例に係る難読キャラクタの解読例 (その 2)



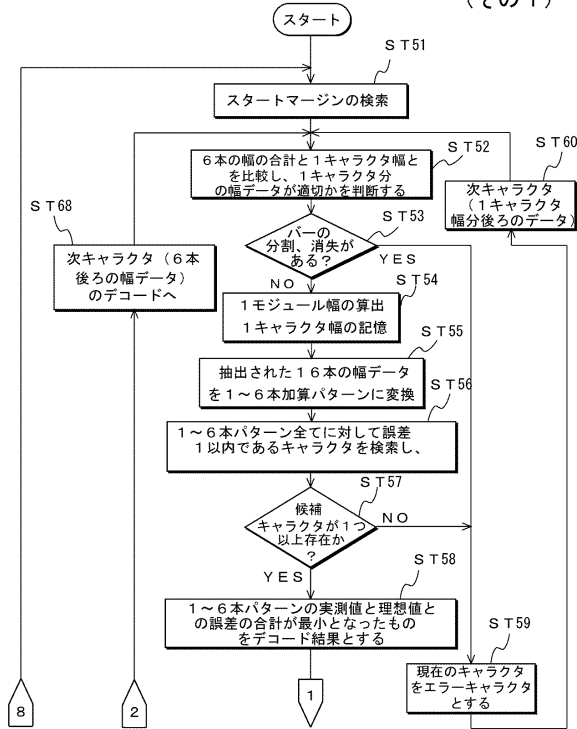
【図 10】

第2の実施例に係る難読キャラクタの解読例 (その 3)



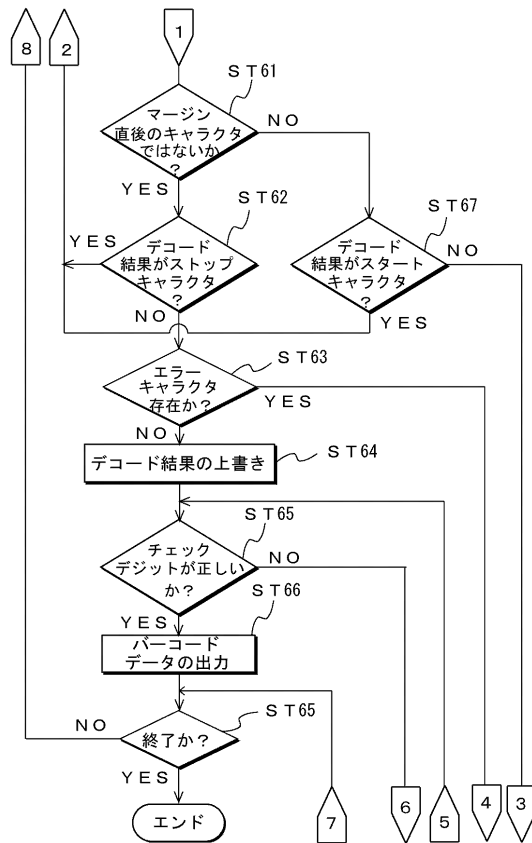
【図 1 1】

第3の実施例に係る難読キャラクタの解読例
(その1)



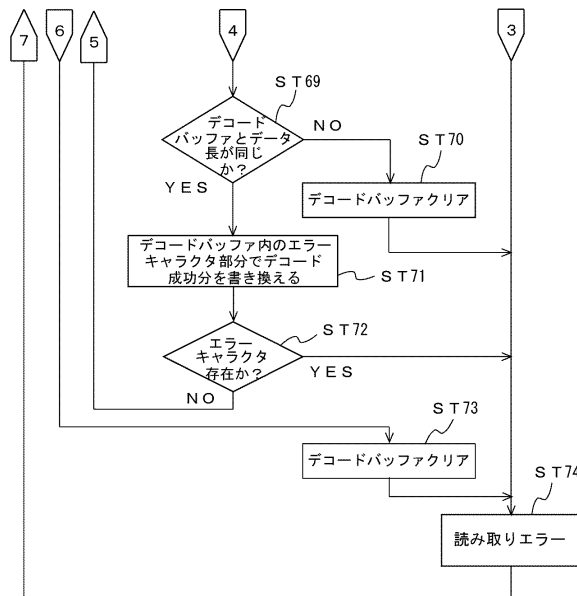
【図 1 2】

第3の実施例に係る難読キャラクタの解読例
(その2)



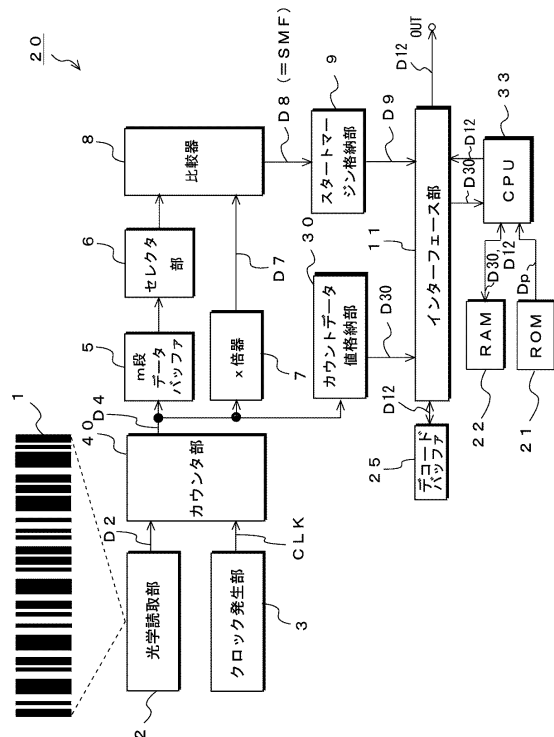
【図 1 3】

第2の実施例に係る難読キャラクタの解読例
(その3)



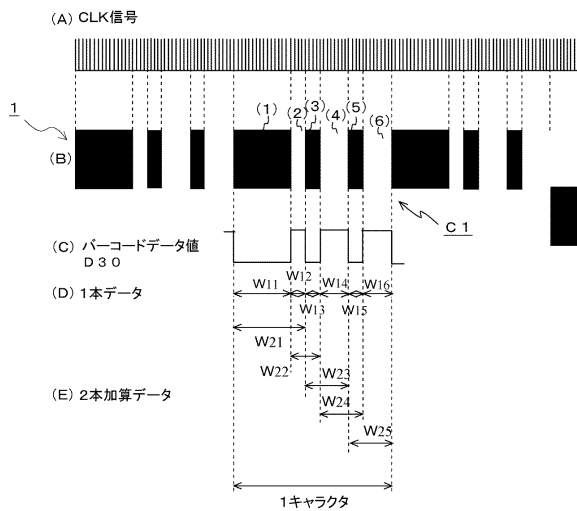
【図 1 4】

従来例に係る光学情報読取装置20の構成例



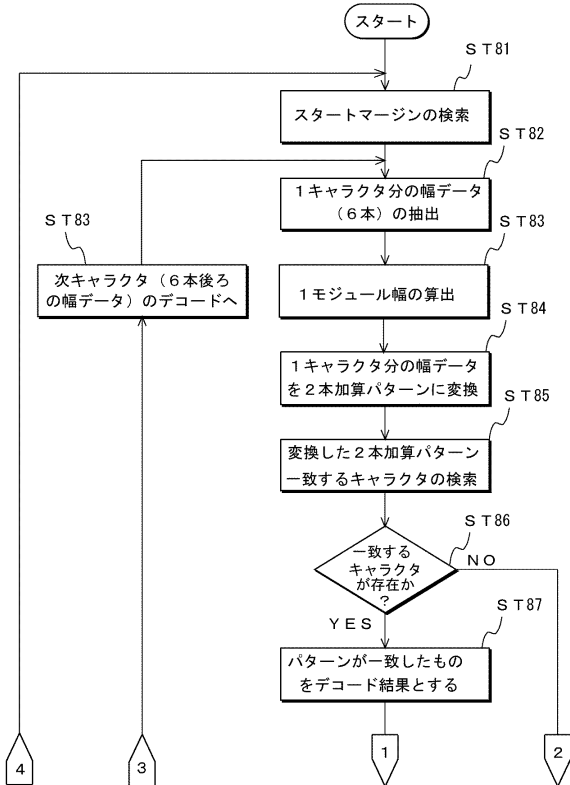
【図 15】

従来例に係る2本加算データの算出例



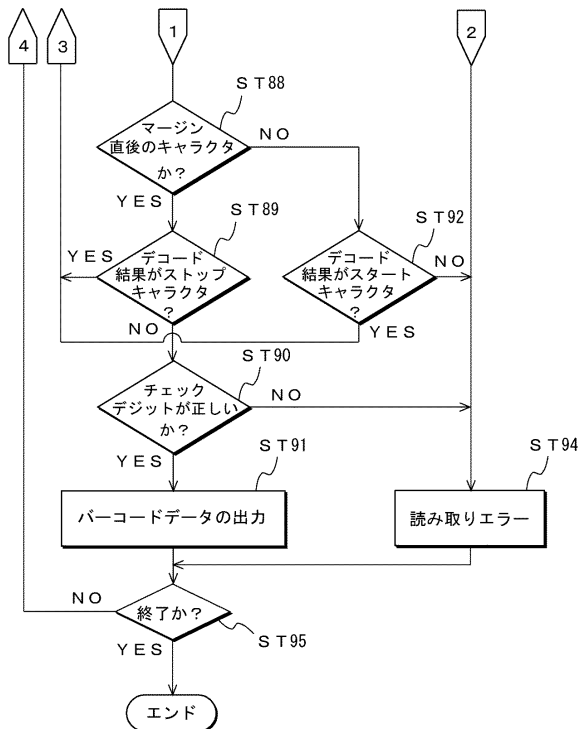
【図 16】

従来例に係る難読キャラクタの解読例



【図 17】

従来例に係る難読キャラクタの解読例



フロントページの続き

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

- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 K 7 / 0 0 - 7 / 1 4