

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7547534号
(P7547534)

(45)発行日 令和6年9月9日(2024.9.9)

(24)登録日 令和6年8月30日(2024.8.30)

(51)国際特許分類		F I			
G 0 6 K	7/14 (2006.01)	G 0 6 K	7/14	0 7 3	
G 0 6 K	7/10 (2006.01)	G 0 6 K	7/10	3 7 2	
		G 0 6 K	7/14	0 1 3	

請求項の数 12 外国語出願 (全19頁)

(21)出願番号	特願2023-43250(P2023-43250)	(73)特許権者	591005615 ジック アーゲー
(22)出願日	令和5年3月17日(2023.3.17)		ドイツ連邦共和国 7 9 1 8 3 パルトキ ルヒ エルヴィーン - ジック - シュトラ ーセ 1
(65)公開番号	特開2023-153742(P2023-153742 A)	(74)代理人	110001069 弁理士法人京都国際特許事務所
(43)公開日	令和5年10月18日(2023.10.18)	(72)発明者	パスカル シューラー ドイツ連邦共和国、7 9 1 8 3 パルト キルヒ、エルヴィーン - ジック - シュト ラーセ 1、シーオー ジック アーゲー
審査請求日	令和5年5月1日(2023.5.1)	(72)発明者	シュテファン ツォップフ ドイツ連邦共和国、7 9 1 8 3 パルト キルヒ、エルヴィーン - ジック - シュト ラーセ 1、シーオー ジック アーゲー
(31)優先権主張番号	22166611		最終頁に続く
(32)優先日	令和4年4月5日(2022.4.5)		
(33)優先権主張国・地域又は機関	欧州特許庁(EP)		

(54)【発明の名称】 光学コードの読み取り

(57)【特許請求の範囲】

【請求項1】

コンピュータ実装された、多数の文字を有する文字列を備えるメッセージを符号化した光学コード(20)読み取り方法であって、

前記光学コード(20)を含む画像データを取得する手順、

前記画像データを評価して前記メッセージを読み取る手順、

読み取られたメッセージの文字とスキーマとを比較する手順であって、該スキーマが、前記メッセージ中の少なくとも1つの位置に対して、読み取り対象の光学コード(20)中で該位置において期待される固定文字を含む、という手順、及び、

前記スキーマとの比較に基づいて、間違っ読み取られたメッセージを認識する及び/又は該間違っ読み取られたメッセージを訂正する手順を含む方法において、

前記スキーマが前記メッセージの少なくとも1つの位置に対して読み取り対象の光学コード(20)中で該位置において期待される可変文字を含み、可変文字が、取り得る文字の部分範囲上で設定されているが、固定文字ではないこと、

前記光学コード(20)が検査合計を備え、前記メッセージの読み取りの妥当性を該検査合計に基づいて確認すること、及び

ある可変文字の位置において、該可変文字が取り得る文字の部分範囲に入らない文字又は読み取り不能の文字を、前記検査合計を用いて訂正し、訂正された文字が前記可変文字が

取り得る文字の部分範囲に入る場合にのみ、その結果を、正しく読み取られたメッセージとして受け入れること

を特徴とする方法。

【請求項 2】

前記スキーマがコード長を備えている及び/又は位置毎に1つの固定文字又は1つの可変文字を含んでいる、請求項1に記載の方法。

【請求項 3】

複数のスキーマのうち適用すべきスキーマとして、前記読み取られたメッセージ内で最も良く見つかる固定文字群を持つスキーマを選択する、請求項1又は2に記載の方法。

【請求項 4】

前記スキーマの固定文字群の最小限の割合、特に少なくとも半分以上が前記メッセージ中で正しく読み取られた場合に、前記スキーマの残りの固定文字を前記メッセージ内に引き継ぐ、請求項1又は2に記載の方法。

10

【請求項 5】

前記スキーマの長さとは一致しない長さを有するメッセージ、又は、ある可変文字の位置に該可変文字が取り得る文字の部分範囲内にはない文字を有するメッセージを、誤読に分類する、請求項1又は2に記載の方法。

【請求項 6】

前記スキーマが、特にASCIIコードにより、印刷不能文字、特殊文字、数字、表音文字、小文字及び大文字という、可変文字が取り得る文字の各部分範囲のうち少なくとも1つを備えている、請求項1又は2に記載の方法。

20

【請求項 7】

前記スキーマが、位置毎に許容文字を示す正規表現として表される、請求項1又は2に記載の方法。

【請求項 8】

前記スキーマを学習するために多数の読み取られたメッセージを評価し、その際、特に、該読み取られたメッセージを稼働中に取得する又はログファイルから読み出す、請求項1に記載の方法。

【請求項 9】

前記スキーマを、前記読み取られたメッセージの各位置における読み取られた文字の分布から学習する、請求項8に記載の方法。

30

【請求項 10】

常に同じ文字が読み取られる位置においてはその該当する固定文字を学習し、様々な文字が読み取られる位置においてはそれらの文字により形成される部分範囲を可変文字として学習する、請求項8又は9に記載の方法。

【請求項 11】

前記スキーマをまず空の領域で位置毎に初期化し、それぞれある位置で読み取られた最初の文字を該位置のために保存し、ある位置で該位置に対して最初の文字より後で読み取られた文字が該位置に対してまだ知られていなかった文字であれば、該位置のための部分範囲を前記読み取られた文字が含まれるように拡大する、請求項8又は9に記載の方法。

40

【請求項 12】

受信光から画像データを生成するための少なくとも1つの受光素子(24)と、請求項1に記載の光学コード(20)読み取り方法が実装された制御及び評価ユニット(26)とを備える光電式コードリーダー(10)。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、請求項1及び14のプレアンブルにそれぞれ記載の光学コード読み取り方法及び光電式コードリーダーに関する。

【背景技術】

50

【 0 0 0 2 】

コードリーダは、スーパーマーケットのレジ、荷物の自動識別、郵便物の仕分け、空港での手荷物の発送等、物流での利用が知られている。コードスキャナの場合、読み取り光線が回転ミラー又はポリゴンミラーホイールでコードを横切るように案内される。カメラベースのコードリーダは画像センサを用いて物体の画像をその表面にあるコードとともに撮影し、画像解析ソフトウェアがその画像からコード情報を抽出する。

【 0 0 0 3 】

ある重要な一群の利用形態では、コードを付した物体がコードリーダの近くを通過するように搬送される。そして、走査型コードリーダがその読み取り領域内に次々に入ってくるコードをそれぞれ検出する。他方、カメラベースのコードリーダでは、ライン走査カメラがコード情報を持つ物体画像を相対運動に伴って次々に1ラインずつ読み取る。2次元画像センサを用いる場合は、撮像レートと搬送速度に応じて多少重なり合う複数の画像データが規則的に取得される。物体をコンベア上で任意の向きに配置できるようにするため、読み取りトンネルに複数のコードリーダを設け、物体を複数の側又は全ての側から撮影できるようにすることも多い。走査型コードリーダも拡散反射光を捕らえ、最終的には複数の画像ラインを捕らえてそれらを物体画像に合成することができるが、このためには実際には画像センサが好まれる。このような物体画像内でコード領域を識別して一次元又は二次元のコードを読み取ることができる。

【 0 0 0 4 】

コードリーダや読み取りトンネルにとっては読み取り率が高いことが最も重要な品質基準の1つとなっている。読み取りエラーになれば、手動での再スキャンや事後的な仕分け等、コストのかかるエラー除去作業が必要になる。このようなエラーの原因となり得るのは、コードそのものの品質、不利な読み取り状況（例えば反射を生じさせるフィルムの下にコードがある場合）、そして最後に、グレースケール値の二値化の過程での評価エラーや計算された走査位置が不正確であるという形の評価エラー等である。

【 0 0 0 5 】

読み取りエラーを避けるため、Data Matrix、QR、Aztec、Maxicode、Dot-Code等の2次元コードやPDF417、MicroPDF等の積層型コード（Stapelcode）に対応した従来のデコードはリードソロモン訂正を備えている。これにより、大面積の欠陥があるコードでさえ正しく読み取られる強力な訂正処理が可能となる。それにより、そうでなければ読み取りができなかったコード（No Read）をなおも読み取ることができるだけでなく、いわゆる誤読（Mis Read）も回避される。誤読の場合、コードが間違っただけでなく、それが気付かれない。リードソロモン訂正ではそれは事実上生じない。そのエラー訂正能力を超えると、該当するコードは単にそれ以上デコードできない（No Read）だけである。

【 0 0 0 6 】

ところがバーコードにはリードソロモン訂正は利用できない。ここで「バーコード」という概念は1次元バーコードと理解すべきであるが、文献ではそれに反して2次元コードもバーコードと呼ばれることがある。バーコードの読み取りは普通、検査合計又はチェックサムを用いて検証される。読み取られたペイロードから算出された検査合計と、一緒に読み取られた検査合計が一致していれば、デコードは正しいものと想定される。違いがあれば、それはペイロード又は検査合計中の少なくとも1つの文字が間違っただけであることを意味しており、それは通例、読み取りエラー（No Read）となる。

【 0 0 0 7 】

検査合計を用いてデコード不能の又は欠陥のある単一のペイロード文字を訂正することができる。即ち、検査合計法を逆にしたりやり方で、該当のペイロード文字を、ペイロードから算出される検査合計と読み取られた検査合計とが一致するような値に事後的に設定するのである。この形の訂正は単一の文字に限定されており、それより広い面積で損なわれたコードには応用できない。その場合は読み取りエラー（No Read）のままである。

【 0 0 0 8 】

10

20

30

40

50

その上、検査合計も気付かれずに間違っ読み取られている可能性がある。そうなる
、デコード不能な又は欠陥のあるペイロード文字が訂正の試みにより間違っ値に設定さ
れる結果となる。それどころか、ペイロード文字は全て完全に正しかったのに、そも
間違っ検査合計を用いた訂正の試みによって初めてエラーが入り込む、ということさ
あり得る。同様のことは検査合計が正しくても起こり得る。それは、2つ以上のペイ
ロード文字が気付かれずに間違っデコードされ、それらのエラーが累積した結果、偶
然に正しい検査合計となる、という場合である。従って、バーコード用の検査合計
に基づく簡単な訂正法は読み取り率が高まると同時に誤読 (M i s R e a d) のリスクも高まる。

【 0 0 0 9 】

従って検査合計では単独のエラーしか確実に捕らえられず、故にバーコード用の従来
の補正は複数のエラーがある状況に弱い。しかし実際には1つのコードが複数箇所で損
なわれていることは全く異例ではない。例えば、紙袋のような軟らかい包装の場合に
下地が大きく波打っていたり、バーコードラベルの一部が厚紙包装の継ぎ目に貼り付
けられていたり、多数の局所的で平面的な欠陥がある高反射性フィルムの下にバー
コードがあったりする。

【 0 0 1 0 】

特許文献1に、検査処理の前に行われる予備訂正においてコードの少なくとも1箇所で
コードワードを該位置に対応する既知のコードワードで置き換えるという光学コード
読み取り方法が紹介されている。その既知のコードワードはパラメータ設定されたり、
上位システムのデータベースにより予め決められたり、コードの読み取り履歴から学
習したりする。これは後段のリードソロモン訂正を補完するために考えられた予備訂
正である。この予備訂正はコードワードが分かっていない可変的なコード部分につ
いては何の改善ももたらさない。可変的な部分におけるエラー訂正は後段のリード
ソロモン訂正の担当であり、特許文献1の利点は、予備訂正の後にはエラー訂正能
力が全てリードソロモン訂正のために利用でき、その一部が既知のコードワード
の訂正のために先に消費されることがない、ということにある。従って、リードソ
ロモン訂正ができないバーコードの場合、少なくともコードの可変的な部分に関し
ては、検査合計を用いた非常に限定的な訂正しかできない場合について述べた全
ての欠点が残っている。

【 先行技術文献 】

【 特許文献 】

【 0 0 1 1 】

【 文献 】 EP 3 428 835 B1

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 0 1 2 】

故に本発明の課題は、より確実且つ頑強な光学コード読み取り方法を示すこと
である。

【 課題を解決するための手段 】

【 0 0 1 3 】

この課題は請求項1及び14にそれぞれ記載の光学コード読み取り方法及び光電式
コードリーダにより解決される。念のために説明しておく、ここで関係するのは自
動的に実行される方法、特にコンピュータで実装される方法である。人間に見させ
ることは、その情報の多さから、いずれにせよあまりに過大な要求である。光学
コードはメッセージを含んでいる。それは該光学コードにより伝えられるべき平
文であり、該コード内に符号化されている。このメッセージは多数の文字を有す
る文字列を備えている。メッセージは好ましくはペイロード文字の他に少なくと
も1つの検査文字、例えば検査合計 (チェックサム) も含んでいる。コード、そ
してメッセージを読み取るため、まず冒頭で説明したいずれかの公知のやり方
で、光学コードを含む画像データを取得する。そしてその画像データ中で好まし
くは前処理によりコード領域を見つけ出す。前処理では画像データが例えばコン
トラストに基づいてセグメント化される。そしてコード領域中の各コードがデコ
ードされ、それによりメッセージが読み取られる。

10

20

30

40

50

【0014】

読み取られたメッセージの文字がスキーマと比較される。スキーマには読み取るべきコードに関する予備知識が入っている。それには固定部分が属している。即ち、スキーマにより、メッセージの少なくとも1箇所において特定の固定文字が期待される。つまり、そこにどの文字があるべきか、若しくはメッセージがその箇所でどのような内容になっているべきかが分かっている。ある使用状況においてそれぞれ独自の固定部分を持って現れる複数のコードファミリーが存在する場合、複数のスキーマがあってもよく、この場合、まず、適切なスキーマ（例えば現在のコードと最も一致するもの）を識別して使用することが好ましい。スキーマによれば、該スキーマの固定部分がメッセージ中で期待通りに読み取られなかったときに、間違っ

10

【0015】

本発明の出発点となる基本思想は、スキーマに可変部分を追加的に設けることにある。メッセージの少なくとも1箇所について、決まった文字に固定されてはいないものの、全ての取り得る文字のうち部分範囲内でのみ変わる可変文字が待ち受けられる。可変文字の典型例は数字又は表音文字である。メッセージ中の読み取られた文字とスキーマとの比較には可変部分も含まれる。可変文字の箇所においてスキーマと一致しない文字がメッセージ中で読み取られた場合、それはエラーと認識される。例えば、スキーマによれば表音文字があるはずの箇所で数字が読み取られる。このエラーはそのまま出力すること（Mis Read）もできるし、例えば検査合計に基づいて訂正を試みることもできる。各々の定義から固定文字と可変文字は互いに排他的であり、従ってスキーマによれば各々の位置は異なっている。

20

【0016】

本発明には、読み取り率が更に高まり、しかもそれが読み取りなし（No Read）と誤読（Mis Read）のいずれにも当てはまるという利点がある。読み取りなしになると、例えば改めて手動でコードを取得するといった骨の折れる事後処理が必要となり、誤読になると、仕分け、発送等、コードにより制御される割り当ての際、気付かれることなく非常に高つくエラーが生じることになる可能性さえある。本発明による訂正は単独のエラーに限定されない。欠落した文字を補うこともでき、その際にコード長を再び正しく再構成することができる。特に、誤読が訂正失敗後に気付かれずに残るリスクが大幅に低減される。後でより詳しく説明するように、スキーマは、効率的な文法を用いて、例えばユーザインターフェイスにおいて分かりやすくパラメータ設定したり、それどころか自動的にデコード工程又は適宜のログファイルから学習したりすることができる。本発明はバーコードに特に適している。それは、バーコードの場合は後段でリードソロモン訂正を利用できないため、特許文献1の予備訂正では不十分だからである。もっとも、本発明をリードソロモン訂正の代わり又はその補完として2次元コードに用いることは依然として可能である。リードソロモン訂正はそれだけでも頑強且つ強力ではあるが、そのエラー訂正能力は無限ではない。

30

40

【0017】

スキーマはコード長を備えている及び/又は位置毎に1つの固定文字又は1つの可変文字を含んでいることが好ましい。従って、このスキーマにはコード長の情報、即ちメッセージの文字の総数が属している。スキーマは完全であること、つまり、そのスキーマから、メッセージの位置毎に、そこにどの固定文字があるか、或いは、取り得る文字のどの部分範囲がそこに可変文字としてあり得るかが分かること好ましい。この完全性にはコード長が暗に入っているが、それを追加的にスキーマの明示的なパラメータとすることもできる。また、スキーマは不完全であることも可能であり、その場合、基本的なコード仕様の制限内で完全に自由な文字が少なくとも1つある。これは特にスキーマの学習中に現れる一時的な状態とすることができる。

50

【 0 0 1 8 】

複数のスキーマのうち適用すべきスキーマとして、読み取られたメッセージ内で最も良く見つかる固定文字群を持つスキーマを選択することが好ましい。ある使用状況において期待される様々なコードファミリーに対応したスキーマのプールが存在する可能性ができる。その場合、どのスキーマを用いて、間違っ て読み取られたメッセージの検査を行うか、或いは読み取られたメッセージを訂正するかという選択を行う。そのためにスキーマの固定部分を一種の指紋と解することができる。読み取られたメッセージに最も合う指紋を持つスキーマが用いられる。理想的な場合には、あるスキーマの固定部分全体がメッセージ内で読み取られる。部分的にしか一致しない場合は、例えば、スキーマと一致して読み取られた固定文字の数及び / 又はそのように読み取られた固定文字の最も高い割合を通じて評価を行うことができる。

10

【 0 0 1 9 】

コードが検査合計を備え、メッセージの読み取りの妥当性を該検査合計に基づいて確認することが好ましい。この場合、メッセージはペイロード文字の他に少なくとも1つの検査文字を含んでいる。読み取られたペイロード文字から検査合計を計算する。それには、最初にコードを生成した時に検査合計の計算に用いられたアルゴリズムと同じアルゴリズムが用いられる。計算した検査合計と読み取られた検査合計が一致したら、メッセージは妥当なものと認められる。これにより誤読 (M i s R e a d) が捕らえられる。

【 0 0 2 0 】

ある可変文字の位置において、該可変文字が取り得る文字の部分範囲に入らない文字又は読み取り不能の文字を、検査合計を用いて訂正し、訂正された文字が前記可変文字が取り得る文字の部分範囲に入る場合にのみ、その結果を、正しく読み取られたメッセージとして受け入れることが好ましい。ここでは、可変部分において検査合計に基づく訂正が行われる。冒頭で、この訂正は単独のエラーに対してしか可能ではなく、複数のエラーがある場合には追加の誤読 (M i s R e a d) が生じ得ることを説明した。検査合計は基本的に、メッセージの完全性を検証したり、訂正には最終的に検査合計との完全な一致が必須であるから該訂正のために考慮したりすることしかできない。ここでスキーマにより追加的に安全性を確保できる。それは、訂正したメッセージはなおもそのスキーマを満たさなければならないからである。これにより追加の誤読の少なくとも一部が捕らえられる。

20

【 0 0 2 1 】

スキーマの固定文字群の最小限の割合、特に少なくとも半分がメッセージ中で正しく読み取られた場合に、スキーマの残りの固定文字をメッセージ内に引き継ぐようにすると好ましい。これもまた固定部分を利用した訂正に該当する。ここではそれに、読み取られたメッセージ中にスキーマの固定部分が十分な信頼性で見つかるという追加条件が課される。そのために、固定文字のうち最小限の割合、特に少なくとも半分がスキーマと一致して読み出されることが要求される。メッセージの固定文字を固定部分の文字で上書きしてもよいが、それによりメッセージが長くなるという意味でそれらの文字を挿入してもよい。この訂正が必要であるという指示は、検査合計を用いた妥当性確認が失敗したときに出すことができる。検査合計を用いてメッセージの妥当性が確認された場合は、効率上の理由から、固定部分の検査と場合によっては訂正を省くことができる。逆に、訂正の後で再び検査合計に基づいて妥当性を確認したりコード長を検査したりすることが好ましい。なおもエラーがあれば、それを読み取りエラー (N o R e a d) として出力する。検査合計を用いて可変部分で訂正を前もって試みることもなお考えられるが、それは誤読 (M i s R e a d) を誘発するという既に説明したリスクを伴う。正しく読み取られた固定文字が最小限の割合に達しなかった場合も読み取りエラー (N o R e a d) とすることが好ましい。その場合、スキーマへの割り当てを行っても、訂正を試みるにはあまりに不確実である。この点に関しては最小限の割合を通じて頑強性を調節することができる。

30

40

【 0 0 2 2 】

スキーマの長さとは一致しない長さを有するメッセージ、又は、ある可変文字の位置に該可変文字が取り得る文字の部分範囲内にはない文字を有するメッセージは、誤読に分類する

50

ことが好ましい。このステップは訂正の後で行うことが好ましいが、特に最初から検査合計に基づいてメッセージの妥当性確認を行うことができた場合、訂正から独立して行うことも可能である。コード長と、可変文字に関するスキーマの事前設定とにより、多重エラーの場合の誤読を少なくとも部分的にまだ見つけ出すことができる。原則的には、複数の文字が間違っている、可変文字がスキーマを満たしていれば、読み取られた検査合計が算出される可能性があるが、そのような多重エラーの可能性は明らかに低減されている。それは言わば、検査合計はそれとして正しく読み取られたが、その検査合計自体が多重エラーの一部である、という場合に当たる。

【 0 0 2 3 】

スキーマは、印刷不能文字、特殊文字、数字、表音文字、小文字及び大文字という、可変文字が取り得る文字の各部分範囲のうち少なくとも1つを備えていることが好ましい。これらはメッセージ中で考えられる文字の区分又はクラスの特によく好適な例である。原則的には区分は全く任意に行うことができる。ただ、上記のような意味論的なクラスはユーザにとって理解しやすく、その結果、その使い方をより簡単に診断して最適化することができる。その上、コード中の規則性も実際には同様に任意の区分よりむしろ意味論的なクラスの形になっている。コードが取り得る文字はASCIIコードの0～127で表現されることがよくある。先に挙げた部分範囲はASCIIコード中にも見つけることができる。

【 0 0 2 4 】

スキーマは、位置毎に許容文字を示す正規表現として表されることが好ましい。正規表現はユーザがスキーマを理解し、場合によっては編集することを容易にする。しかも内部処理が簡単になり、デコーダのプログラミング時にエラーが発生しにくくなる。代わりにスキーマを独自に定義することも考えられるが、それは正規表現の持つ分かりやすさと形式的な規則性に少なくとも近似的に達していることが好ましい。

【 0 0 2 5 】

スキーマを学習するために多数の読み取られたメッセージを評価し、その際、特に、該読み取られたメッセージを稼働中に取得する又はログファイルから読み出すことが好ましい。確かに、スキーマを直接決めておくことは基本的には考えられよう。それは特にグラフィカルユーザインターフェイスにおけるパラメータ設定によってもよいし、データ記憶媒体若しくはネットワークを用いた読み取りによってもよい。スキーマの診断又は改良にとってそれは全く有益である。しかし、本実施形態では自動的に学習を行い、以てユーザをこの業務から解放するものである。学習は読み取られたメッセージに基づいて行い、そのメッセージは好ましくは現在又は過去の稼働（特にログファイル）から得ることができる。更に稼働中にスキーマの適合化を行うことも考えられる。スキーマを学習する基になったメッセージが正しく読み取られたメッセージであるということが分かっているならば有利である。加えて、最初から正しく読み取られた読み取り結果だけ、特に、検査合計に基づく訂正を受けなかった読み取り結果だけを用いることが好ましい。ただし、文字が欠落しているメッセージ及び/又は誤った文字を含んでいるメッセージが少しあったとしても、例えば統計的な方法により、読み取られたメッセージからスキーマを導き出すことは可能である。学習の際には、1つのコードファミリーからのコードだけを入力すること、或いは、複数のコードファミリーのために複数のスキーマを学習することができるように、読み取られたメッセージの仕分けを行うことが好ましい。仕分けの基準はコード長とすることができるが、スキーマそのものの一部、特に異なる固定部分でもよい。そうすると、学習の進行中に1つのスキーマから固定部分の違いによって2つ以上のスキーマが作成されることが起こり得る。

【 0 0 2 6 】

スキーマは、読み取られたメッセージの各位置における読み取られた文字の分布から学習することが好ましい。これによれば、メッセージの各位置にどの文字が現れるかが確定される。加えて、ある文字が各位置においてどのような頻度で読み取られたかを数えることができる。

【 0 0 2 7 】

10

20

30

40

50

常に同じ文字が読み取られる位置においてはその該当する固定文字を学習し、様々な文字が読み取られる位置においてはそれらの文字により形成される部分範囲を可変文字として学習することが好ましい。これによれば、ある位置で繰り返し変わらずに読み取られる文字はスキーマの固定部分に加えられる。その際、例えば固定部分は隣接し合う文字のブロックを形成するとか、メッセージの始め又は終わりにあるといった追加の規準を設けることができる。ある位置において読み取られる文字が様々に変わる場合、それにより張られる部分範囲はスキーマの可変文字とみなされる。この部分範囲は、ある特別な部分範囲を全体として覆うために、場合によっては更に拡大することができる。例えば、ある位置で文字1、3、6が読み取られたとすると、部分範囲は、実施形態に応じて、正確にそれらの文字{1, 3, 6}から成るものでもよいし、読み取られた最小及び最大の文字である1と6の間で読み取られなかった文字2、4、5も含めた全区間{1, 2, 3, 4, 5, 6}から成るものでもよいし、完全な数字のクラスから成るものでもよい。ある位置においてほぼ任意のランダムな文字が読み取られた場合、ここではスキーマのために固定部分も有用な可変部分も学習することはできない。スキーマ内に自由な箇所、つまり事実上制限のない可変文字を残すか、或いは学習を中断する。更に、読み取られたメッセージのうちこの状況を招いた異常値を取り除いた部分集合を見つけ出すことを試みることができる。その異常値を出力することで、この読み取られたメッセージをスキーマ中で考慮しないことが適正かどうかをユーザが調べることが可能となるようにすることができる。

10

【0028】

スキーマをまず空の領域で位置毎に初期化し、それぞれある位置で読み取られた最初の文字を該位置のために保存し、ある位置で該位置に対して最初の文字より後で読み取られた文字が該位置に対してまだ知られていなかった文字であれば、該位置のための部分範囲を前記読み取られた文字が含まれるように拡大する、というようにすることが好ましい。これはスキーマの学習の有利な実装である。ここでも例のごとく、それぞれのスキーマを学習する基になる読み取られたメッセージが同じコードファミリーに属するように仕分けをすることが好ましい。まずはどの位置についても予備知識がない空のスキーマが初期化される。それぞれある位置で読み取られた文字が最初はスキーマの固定文字とみなされる。しかしそれはその後、更に読み取られるメッセージを通じて正しいと証明されなければならない。その際、固定文字は少なくともm回又はn回中m回読み取られたものであるということも要求することもできる。ある位置において別の文字が読み取られたら、それをn回中m回の基準に従って異常値とみなすことができる。しかし好ましくは、それまでこの位置で読み取られた文字により可変文字の部分範囲を張る。ある可変文字の位置において新規の文字が読み取られるたびにその部分範囲を拡大することができる。好ましくは部分範囲を1つのクラス(数字、表音文字等)に限定しておく。そうでなければ、部分範囲を非常に大きくすること、或いは学習のエラーメッセージを出すことが考えられる。なぜなら、この位置にはスキーマ中で有意義に捕らえられる規則性又は体系性が認められないからである。

20

30

【0029】

好ましい発展形態では、光電式コードリーダに、受信光から画像データを生成するための少なくとも1つの受光素子と、本発明に係る光学コード読み取り方法が実装された内部及び/又は外部の制御及び評価ユニットとが設けられる。

40

【0030】

以下、本発明について、更なる特徴及び利点をも考慮しつつ、模範的な実施形態に基づき、添付の図面を参照しながら詳しく説明する。

【図面の簡単な説明】

【0031】

【図1】読み取り対象の光学コードを付した物体を搬送するベルトコンベアの上方に模範的に取り付けられたコードリーダの概略全体図。

【図2】フィルムの下にあり、反射のため読み取りが難しい光学コードの画像例。

【図3】光学コードから読み取られたメッセージを検査及び/又は訂正するためのスキーマ

50

マを学習するための模範的なフローチャート。

【図4】複数の読み取られたメッセージをそれらから学習したスキーマの説明図とともに示した図。

【図5】スキーマの可変部分の部分範囲において取り得る文字（ここではASCIIコード）の模範的な区分。

【図6】光学コードから読み取られたメッセージをスキーマに基づいて検査及び/又は訂正するための模範的なフローチャート。

【発明を実施するための形態】

【0032】

図1はベルトコンベア12の上方に取り付けられた光電式コードリーダ10を示している。ベルトコンベア12は矢印16で示したようにコードリーダ10の検出領域18を通過するように物体14を搬送する。物体14はその外側表面にコード領域20を持っており、これがコードリーダ10により検出されて評価される。このコード領域20は上面に付されているか、少なくとも上から見えるように付されている場合にのみコードリーダ10で認識できる。そこで、図1の描画から逸脱して、例えば側面又は底面に付されたコード22を読み取るために複数のコードリーダ10を異なる方向から取り付けることで、いわゆる全方向からの多重読み取りを可能にしてもよい。読み取りシステムにおける複数のコードリーダ10の配置は実際には読み取りトンネルとして実施することがほとんどである。このようにコードリーダをベルトコンベア付近に固定して用いることは実際に非常によくある。しかし、本発明はコードの読み取り若しくはコードリーダ10そのものに関するものであるから、本例を限定的なものとして解釈してはならない。例えば、コードは手動でスキャンすることもできる。また、プレゼンテーション型の応用ではコード又はコードを付した物体14をコードリーダ10の読み取り領域内にかざすことができる。

【0033】

コードリーダ10は搬送されている物体14及びコード領域20の画像データを受光器24で取得し、そのデータが制御及び評価ユニット26によって画像解析及びデコード法を用いて更に処理される。本発明にとって具体的な撮像法は重要ではないため、コードリーダ10はそれ自体公知である任意の原理により構成することができる。例えばその都度1ラインだけを捕らえる場合、ライン状の画像センサを用いてもよいしスキャン法によってもよく、制御及び評価ユニット26は搬送運動の進行中に得られる複数のラインを結合して画像データにする。マトリクス状の画像センサを用いれば一度の撮影だけでより広い領域を捕らえることができる。ここでもまた複数の撮影画像の結合を搬送方向にもそれと交差する方向にも行うことができる。コードリーダ10の最も重要な機能はデコード、即ち光学コード内に符号化されたメッセージを平文として読み取ることである。メッセージはペイロード文字から成る文字列であり、好ましくは少なくとも1つの検査文字が付されている。検査文字は典型的には末尾にある。コードリーダ10はインターフェイス28を通じて情報（コードから読み取られたメッセージ、画像データ等）を出力する。

【0034】

以下、コードからそれぞれ読み出されたメッセージの検査及び/又は訂正について図2～6を参照しながら説明する。これは制御及び評価ユニット26において行われることが好ましい。もっとも、画像データ又は中間結果をインターフェイス28経由で出力し、デコード、検査及び/又は訂正の少なくとも一部を、制御用コンピュータ、ネットワーク又はクラウド等の外部の上位システムに出すことも同様に考えられる。コード領域20のセグメント化及び発見のための画像データの前処理、並びにデコードそのものは公知であるものとし、説明を行わない。

【0035】

図2は、フィルムの下にあり、反射のため読み取りが難しい光学コードの画像例を示している。ラベル上には実際よくあるように複数のコードがある。コードの読み取りにとって特に重要なのは、後段での物品の仕分けに関する情報を保持している、いわゆる仕分け関連コードである。そばに他のコードが付されている可能性もある。しかし、例えば図2

10

20

30

40

50

の一番上のコードが読み取ることができても仕分け関連コードが読み取れなければ役に立たない。また、コードの反射又は他の破損若しくは妨害がコード領域 20 の全てのコードに該当するという状況や、逆に全てのコードが仕分けに関連している若しくはそれらの正しいデコードが他の所で使用上不可欠であるという状況も当然想定される。

【0036】

図2のように反射又は他の破損により損なわれたコードに対する読み取り率を上げるため、スキーマ中にあるコードに関する予備知識を考慮に入れる。そうすると、読み取られたメッセージをスキーマに基づいて補完することで、読み取りエラー (No Read) をなおも避けたり、誤読 (Mis Read) を検出したりできる。これについて、図2にも示したようなバーコードの例で説明する。本発明はバーコードに特に有用である。なぜなら、他のコード形式ではリードソロモン符号化等の強力な訂正が利用可能だからである。ただし、それにより他の種類のコードへの本発明の応用を排除するものではない。

10

【0037】

図3はスキーマの自動学習のための模範的なフローチャートを示している。代わりに、スキーマを特にグラフィカルユーザインターフェイスを介して入力したり、データ記憶媒体から読み込んだりすることもできる。ユーザインターフェイスは学習したスキーマを表示し、検査し、場合により編集することにも適している。

【0038】

ステップ T1 では多数の読み取られたメッセージが取得される。それらはその後の使用時に出現するような光学コードをデコードした結果である。それらの読み取られたメッセージは稼働の進行中に徐々に及び/又は過去 (例えば以前の読み取り結果を含むログファイル) から取得することができる、

20

【0039】

ステップ T2 では、読み取られたメッセージを評価し、出現するコード形式を求める。これは例えば出現するコード長の統計量による。それにより、どのコードファミリーのためにスキーマを生成すべきかを確認することができる。例えば、後でスキーマに基づいて仕分け関連コードのみを検査若しくは訂正しさえすればよい場合、それに対応するコードファミリーがそのコード長、特定の識別子又は内容によって認識される。スキーマはその都度、読み取られたメッセージのうち該当のコードファミリーに属する部分集合のみを基にして作り上げられる。単一のスキーマだけを、最も頻出するコード又は所与のコードファミリーのためだけに学習することが考えられる。その場合、ステップ T1 では、そのコードファミリーに合った仕分け済みの読み取られたメッセージのみを用意することが好ましい。

30

【0040】

ステップ T3 では、読み取られたメッセージに基づき、あるコードファミリー若しくはスキーマに対して、どの文字がどの位置に現れるかを評価する。それには例えば、単に各位置に対して読み込まれた大量の未整理の文字をその位置に結びつけたり、該位置において読み取られた文字が全て入るような領域をそれぞれ設けたり、読み取られた文字の真の頻度分布を位置毎に形成したりすることができる。

【0041】

40

そしてステップ T4 では、文字毎の分布からスキーマが導き出される。スキーマは普通、固定部分と可変部分を備えている。スキーマの固定部分の固定文字は、各コードファミリーにおいて学習中に読み取られる全ての文字において繰り返されるか、或いは異常値を考慮するために n 回の事例のうち少なくとも過半数である m 回繰り返される。固定部分はスキーマのコードファミリーを識別する手段でもある。それ故、少なくとも1つの追加のスキーマを生成することがあり得る。なぜなら、読み取られたメッセージ中に2つの異なる固定部分が同じ位置にあれば、それは恐らく実際には同じコードファミリーではなく2つのコードファミリーだからである。固定部分は後で稼働中に、読み取られた最新のコードに対する正しいスキーマを見つけ出すために役に立つ。

【0042】

50

スキーマの可変部分の可変文字は、固定文字のように読み取られたメッセージを超えて同一ではなく、取り得る文字の部分範囲内に留まる。それは例えば表音文字又は数字である。ステップ T 3 においてある位置で求められた分布の区間幅があまりに大きくなる場合、その可変文字にはもう予測力がほとんどないからいつかその意味を失う。そうするとそれは自由文字となる。これをスキーマ中で容認してもよいが、これらの読み取られたメッセージの入力データには適切なスキーマがあり得ないというエラーメッセージを結果として出すこともできる（特に複数の自由文字がある場合）。

【 0 0 4 3 】

ステップ T 1 ~ T 4 は上述のように厳密に一度だけ実行しなければならないわけではない。特に段階的又は反復的なやり方が可能であり、その場合、更に別の読み取られたメッセージが次々に評価される。その場合、それに応じて、可能であればステップ T 2 において別のスキーマを新たに開き、ステップ T 3 と T 4 においてそれを補完することで、読み取られたメッセージに適したスキーマにする。このようなスキーマの反復的な適合化は本来の読み取り稼働の間にオンザフライで行うことも考えられる。スキーマの説得力を統計的に実証するため、少なくとも最小数の読み取られたメッセージをスキーマに取り込むことを要求することができる。

【 0 0 4 4 】

図 4 は複数の読み取られたメッセージをそれらから学習したスキーマの説明図と共に示している。これは図 3 に示した学習の特別に有利な実装であり、これを用いて、スキーマの形で予備知識を獲得する処理について更に説明する。

【 0 0 4 5 】

このために図 5 はスキーマの可変文字のための各部分範囲で取り得る文字の模範的な区分を補足的に示している。表にはバーコードに特に関係のある 0 ~ 1 2 7 の A S C I I 値が含まれており、それらが印刷不能文字、特殊文字 # 1、数字、特殊文字 # 2、大文字、特殊文字 # 3 及び小文字という 7 クラスに区分されている。対応する A S C I I 値の範囲は表中の凡例と強調用のバーからすぐに見て取ることができる。当然ながら本発明は特にこれらのクラスに限定されるものではない。しかし、意味論的なクラス分けには意味がある。なぜならそれはユーザにとってより理解しやすく、また、実際のコードファミリーは、可変文字が任意ではなく意味論的な部分範囲に限定されているような体系に従うことが典型的だからである。

【 0 0 4 6 】

図 4 に戻って、同図の左上には。図 3 のステップ T 1 に記載の入力データとして、コード長が 2 1 の 1 0 個の読み取られたメッセージが示されている。読み取られたメッセージは訂正されていないことが好ましい。この単純な例では、読み取られたメッセージが全て同じコードファミリーに属しており、スキーマを 1 つだけ学習すればよいものと定めているため、図 3 のステップ T 2 は考慮されない。

【 0 0 4 7 】

図 4 の右上には、図 3 のステップ T 3 に対応して、コードの位置毎にそれまでに出現して読み取られた文字の A S C I I コードの下限と上限が示されている。これらは文字にとってあり得ない値（例えば [- 1 , - 1] ）で初期化される。その後、それぞれの位置毎に測定された文字で境界が変更される。例えば、1 行目において 1 番目の文字として A S C I I コード 7 4 を持つ「J」が読み取られ、下限がその値 7 4 に設定される。2 行目の 1 番目の位置で「J」が再び現れて読み取られ、上限も値 7 4 に設定される。あるいは、ある位置における文字を最初に読み取ったときにその値を上限に配置したり、両方の境界に配置したりしてもよい。図示したやり方には、少なくとも、値の繰り返しを最初に確認したことが認識できる状態で残るといった利点がある。ここで、繰り返しを計数する別のパラメータを補ったり境界内に符号化したりしてもよい。

【 0 0 4 8 】

後半の可変文字は、文字の最初の読み取り時にはその位置において全く同様に扱われる。そのような位置においてその後読み取られる文字は、既知の境界の間に入っているか

10

20

30

40

50

、或いは下限又は上限を上書きするから、その新たに読み取られた文字はもうその部分範囲内に含まれている。例えば末尾の文字の場合、まず、読み取られた「0」のASCIIコードである48が下限に現れている。2行目では、上限が、読み取られた「6」に対応して、これに該当するASCIIコードである54に設定されている。10個のメッセージを読み取った後では、数字「0」と「9」に対応するASCIIコードを境界とする部分範囲[48, 57]が得られている。なぜならこれらは10個のメッセージ中でこの位置に現れる両極端の値だからである。

【0049】

図4の下部には図3のステップT4により生成されるスキーマが示されている。そのために、ステップT3においてメッセージ若しくはスキーマの位置毎に得られた分布が評価される。その分布が図4の右上部分の最終行に示されている。読み取られたメッセージの前半部分では「JJD01460000」という文字列が常に繰り返されている。これがスキーマの固定部分である。これは、これらの各位置において下限と上限が一致していること、即ち10個の全メッセージにおいてそれぞれ同じ文字が読み取られていることから分かる。

【0050】

メッセージの残りの後半部分は可変部分である。ここではそれぞれ下限と上限が互いに違っている。基本的には、それぞれの可変文字に対して、下限と上限によりそこで得られた区間をそのまま用いることができる。しかし、好ましくはクラスにまで一般化し、上限と下限をそのクラスの上下の境界にそれぞれ設定する。例えば、図4の10個のメッセージ中の12番目の位置、即ち可変部分の最初の位置では、数字の「8」又は「9」しか現れていないから、[56, 57]という部分範囲が定められている。しかし、この元々の部分範囲から逸脱してクラスを開き、スキーマ中ではこの位置において全ての数字を許容するものとする。故にスキーマの部分範囲は[48, 57]となる。

【0051】

ある位置で読み取られた文字がクラスの境界を超えてしまうことが考えられる。これは訓練を中断する理由となり得る。なぜなら、そのようなスキーマは信頼できないはずだからである。代案では複数のクラスを許容する。その場合、そのために、最初に算出された境界をその下又は上に隣接するクラスの境界にずらすことができる。これにより、複数のクラスに亘る繋がった部分範囲ができる。原則的には、図5のASCII表において互いに離れている数字と小文字のように、分離したクラスも考えられる。その場合、既に部分範囲の決定時に、新たに読み取られた文字がそれまでのクラス境界を超えているかどうかを検査する必要があり、超えているときには新たなクラスを含めるために2つの追加の境界を補う。このように複数クラスを持つ可変文字のクラスの1つを単一の文字に置き換えることができる。例えばその場合、この可変文字には数字と特殊文字「+」が許容される。クラスをちょうど必要なだけ大きくすることが有意義である。可変文字は稼働中に出現する文字をカバーすべきであるが、この条件下でできるだけ区分けを精密にするべきである。

【0052】

図4のスキーマは始めに固定部分があり、終わりに可変部分がある。本発明はこれに限定されず、複数の固定部分及び/又は複数の可変部分を任意に配置することができる。原則的には位置及び文字毎にそれが固定文字か可変文字かを定めることができ、この点については何の制限もない。

【0053】

図6は光学コードから読み出されたメッセージをスキーマに基づいて検査及び/又は訂正するための模範的なフローチャートである。予め少なくとも1つのスキーマを決めておくか、図3~5を参照して説明した方法で学習しておく。

【0054】

ステップS1では、コードリーダ10によりコード領域20を含む画像が撮影される。ステップS2では、その画像データを図1を参照して述べたようにデコードで評価することで、コードに含まれているメッセージが読み出される。

【0055】

10

20

30

40

50

ステップ S 3 では適切なスキーマが割り当てられる。これはコード長、コード形式及び / 又はメッセージの一部に基づいて決定される。特に、そのスキーマに合ったコードを識別するために該スキーマの固定部分を一種の指紋として利用することができる。或いは、スキーマが 1 つしかない、又は 1 つのスキーマが予め固定的に決められている。メッセージに合うスキーマがない場合、図 6 に示したその後の手順は適用できないため、ステップ S 4 で中断されるか、跳び超えられる。その場合、読み取られたメッセージは本発明がなかったかのように扱われる。即ち、読み取られたメッセージを出力するか、認識されたエラーをそのまま残すか、或いはスキーマに基づく本発明の訂正又は検査とは別の訂正又は検査を更に行うことができる。

【 0 0 5 6 】

適切なスキーマを用いて今度は、ステップ S 2 において読み取られたメッセージが有効なメッセージかどうかを区別される。そのために例えば検査合計を参照することができる。場合によってはメッセージの少なくとも一部が全く読み取り不能かもしれない。

【 0 0 5 7 】

メッセージが無効の場合、ステップ S 5 においてスキーマに基づく訂正を試みる。そのために、スキーマの固定部分が、読み取られたメッセージ中の対応する位置にある文字と比較される。好ましくは、スキーマ中及び読み取られたメッセージ中において固定文字のうち最小限の割合、例えば少なくとも半分が一致すべきであるとする。当該スキーマに合った固定部分を持つコードが本当に読み取られたことを十分確実にする。可能であれば、このような検査を既にステップ S 3 で行っておけば、それを繰り返さずに済む。

【 0 0 5 8 】

比較の結果が肯定的であれば、スキーマの固定部分をメッセージに転記する。その際、メッセージの固定文字をスキーマの固定文字で上書きし、必要であれば文字を割り込ませることで全体として正しいコード長になるようにする。別の実施形態として、メッセージのコード長が訂正前に既にスキーマの必要コード長と一致することを要求することができ、その場合は文字の割り込みは行わない。

【 0 0 5 9 】

メッセージの固定部分はステップ S 5 による訂正により強制的に正しくなる。ただ、固定部分における最小限の割合の一致では不十分であり、読み取られたコードが実際にはその固定部分を有していない、という可能性がまだ残っている。しかし、場合によってはエラーが固定部分だけにあるのではなく、更に可変部分でも 1 又は複数の文字が間違っ

【 0 0 6 0 】

第 1 の訂正例では、スキーマが図 4 で学習したものであり、コード長が 2 1、位置 1 ~ 1 1 が固定部分「JJD01460000」で、その後ろの位置 1 2 ~ 2 1 が数字から成る可変部分であるものとする。このスキーマを他の全ての訂正例でも参照する。いま、例としてメッセージ「XJD023600009278066464」が読み取られたとする。検査合計は 1 8 で、太字の箇所（1 番目、5 番目及び 6 番目の文字）がスキーマの固定部分と違っている。検査合計に基づく検査は成功しないため、訂正が試みられ、半分を明らかに超えて一致している固定部分が、読み取られたメッセージに転記され、「JJD014600009278066464」となる。今度は検査合計も合っているため、これは有効なメッセージである。

【 0 0 6 1 】

第 2 の訂正例では、読み取られたメッセージが「D014600009278066464」であるものとする。今度のエラーは最初の 2 つの文字が読み取られなかったこと、即ち先頭の「J」がないことである。故にこのメッセージは必要なコード長である 2 1 より 2 文字だけ短い。従って、もし検査合計が偶然に誤って有効なメッセージを示していても大した問題で

10

20

30

40

50

はでない。それは独立した第2の指標である。固定部分の一致は、要求された最小限の割合である50%を依然として超えている。故に、スキーマの固定部分を正しい位置に割り込ませることができる。即ち、固定部分の最初の2つの文字「JJ」を、訂正のために、読み取られたメッセージの前に置くことができる。これにより、読み取られたメッセージのコード長は正しい値である21になり、算出される検査合計も今度は正しくなる。

【0062】

第3の訂正例では、検査合計が読み取れなかったために訂正前のメッセージが無効になっている。それでもまず訂正の方法に何ら変わりはない。ただ、その後の検査ができず、コード長だけがなお検査できるにすぎないから、その結果は非常に信頼性が低い。ここでコード読み取りの反復を要求することが考えられる。いくつかの実施形態ではそうでなくとも各コードが複数回読み取られるから、その場合は、可変部分の位置にある文字をエラーなしで読み出すことができたと十分に確信できるようにするために、複数回のデコード成功に相当する閾値を上げることが考えられる。

10

【0063】

ステップS5の訂正に続いて今度は場合分けが行われる。訂正によっても有効なメッセージが再構成できなかった場合、ステップS6は読み取りエラー（ここではNoRead）のままである。そうではなく訂正が成功した場合、誤読の認識を行うステップS7の処理に進む。ステップS3に続く場合分けの2番目の経路、即ち、メッセージがステップS5による訂正なしで既に有効となった場合も、このステップS7に至る。なお、図6から離れて、メッセージの元々の読み取り又はスキーマに対してより高い信頼性がどうかによって、ステップS3で既に有効とされたメッセージについてもステップS5による訂正を常に実行することも考えられよう。

20

【0064】

ステップS7で今度は更に誤読（MisRead）が認識される。これは、ステップS5による訂正あり又は訂正なしで有効なメッセージが読み取られた場合であって、特にペイロード文字から算出された検査合計と読み取られた検査合計が一致している場合である。つまりこの妥当性確認は冒頭で既に論じたようにエラーが1つだけの場合にのみ信頼できるものであり、エラーが複数あれば偶然に正しい検査合計が算出され得る。

【0065】

最初に任意選択で、これが既にステップS3又はS5において行われていなければ、スキーマの固定部分とその固定部分の位置に対応するメッセージ中の文字との比較に基づいて、スキーマがメッセージに合っていることを確かめる。それには固定部分が最小限の割合（例えば50%）まで一致していなければならない。

30

【0066】

スキーマの可変部分については、読み取られたメッセージの可変文字がその属する位置においてスキーマの部分範囲内にあるかどうかを検査する。例えば、ある位置においてスキーマが数字を要求しているのに、特殊文字が読み取られていたら、そのメッセージは有効とはみなされない。加えて、スキーマのコード長が読み取られたメッセージのコード長と等しいかどうかを検査することができる。ステップS7の基準によって有効でないメッセージがあれば、図6のフローは有効な読み取りなし（この場合はMisRead）としてステップS6で終わる。

40

【0067】

それ以外の場合、メッセージは有効であるとみなされる。デコーダが元々エラーなしで読み取ることができた可能性もあれば、ステップS5でエラーが訂正され、ステップS7で誤読も認識されなかった可能性もある。このメッセージはステップS8で出力されるか、本発明より前から既に通例となっているステップにより更に処理される。

【0068】

以下の誤読の認識のいくつかの例でも図4で学習したスキーマが引き続き用いられる。第1の非常に簡単な例では「JJD014600005183211」というメッセージが読み取られた。これは18文字であり、スキーマのコード長である21より短いから誤読である。

50

【 0 0 6 9 】

第2の例では「JJD01460000MJ18321108」というメッセージが読み取られた。今度はコード長は合っているが、太字の箇所（12番目と13番目の文字）がスキーマにいう数字ではなく大文字の「MJ」と読み取られている。これも同様に誤読である。

【 0 0 7 0 】

第3の例では、ステップS7の処置でも見つけることのできない誤読が生じている。固定文字とは違って可変文字には、複数のエラーがなおも補い合い、読み取った検査合計になる一定の余地がある。その際、検査合計は正しく読み取られている可能性もあれば、それ自身が間違っていて読み取られていて複数のエラーの一部になっている可能性もある。今度は正しいメッセージは「JJD014600009283279219」、読み取られたメッセージは「JJD014600009275279219」であって、太字で示した数字（14番目と15番目）は間違っていて読み出された検査合計を用いて訂正されている。このエラーは認識できない。1つには検査合計が合っており、その機能は訂正を試みたことにより言わば消尽されている。その上、数字が組み込まれているため、スキーマを満たしている。検査合計を用いた訂正により数字以外が組み込まれていたとしたら、それはスキーマを用いればなお見つかったであろう。この第3の例では、読み取られたメッセージが実は誤読（Mis Read）であるにも関わらずステップS8で有効とみなされる。当然ながら、検査合計を用いた訂正を試みていなかった可能性もあり、その場合は読み取りなし（No Read）に留まる。既に冒頭で論じたように、このリスクは検査合計を用いた訂正の場合には常に存在しており、それにより読み取り率が上がる（No Readが減る）が、同時に誤読（Mis Read）の危険性も高まる。本発明に係るスキーマによって後者の危険性は少なくとも更に低減される。

【 0 0 7 1 】

スキーマを、一目で分かり、跡づけて理解できるやり方で描出又は表現すれば有意義である。そうすれば診断や、場合によっては手動での最適化が容易になる他、スキーマに関連するステップのプログラミングをエラーなしで体系的に行うことも容易になる。簡単な実装として、図4の例のようにスキーマの位置に部分範囲の境界を結びつけることができる。これは複数のクラスの境界及び/又は部分範囲中の許容文字のリストに拡張することができる。

【 0 0 7 2 】

これは疑似コードで次のように表現することができる。ここではクラス（Type）のリストが拡張可能であり、特に図5のリストを用いて拡張できる。

【 0 0 7 3 】

```
Schema=CodeCharSchemaの配列
CodeCharSchema=SingleSchemaEntryのリスト
SingleSchemaEntry=
Record
  Type: (SingleCharacter, CharInterval)
  Value: Char; /* 個々の文字、例えば特殊文字 */
  Min, Max: Char; /* これにより区間を既述する */
End;
```

【 0 0 7 4 】

スキーマを既述するための更なる代替物は正規表現である。これを用いれば、スキーマの定義が、多くのユーザに知られている厳密に定義された構造に還元され、しかもそのために設計、編集及び変換用の標準的なコンポーネントが存在する。正規表現として定義されたスキーマの例は「JJD...[a-z][a-z][0-9]...」である。これは「JJD」という固定部分で始まり、それに3つの任意の文字が続く、その後ろに2つの小文字、1つの数字、そして再び3つの任意の文字が続く。ただし、どの正規表現もスキーマになり得るわけではないことに注意されたい。例えば、「^...[0-9]...\$」には行の初めと行の終わりが含まれているが、スキーマにはこれに対応するものはあり得ない。編集機能はこれらの可能性を最

初から全く提供しないか、少なくとも警告により制止すべきであろう。

【 0 0 7 5 】

正規表現の代わりに独自の体系又は言語をスキーマのために設計することが考えられる。そうすればスキーマの必要条件により良く適合させることができる。例えば、大文字を「C」、小文字を「c」、数字を「N」で表し、固定文字は括弧に入れ、自由文字はここでも「.」で表すことができる。そうすると上記の正規表現の例は「[JJD]...ccN...」という、よりコンパクトな書き方になる。

10

20

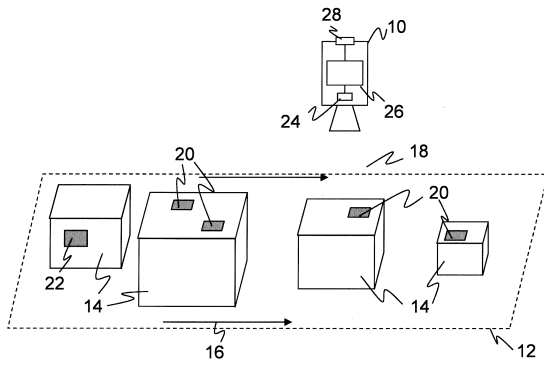
30

40

50

【図面】

【図 1】

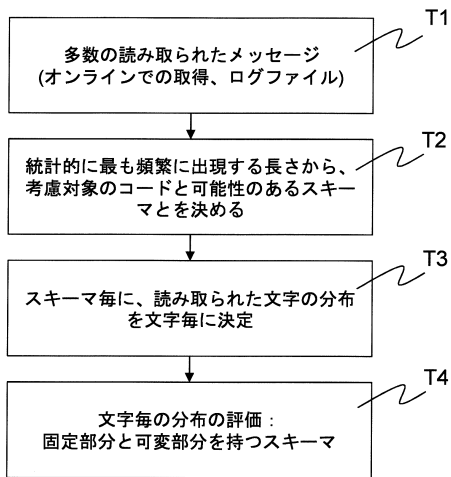


【図 2】



10

【図 3】



【図 4】

結果 ASCII区間 (最小、最大。不明=-1)

JDD014600009191627300 [74,-1][74,-1][68,-1][48,-1][49,-1], ..., [48,-1],[57,-1],..., [48,-1]

JDD014600008744502826 [74,74][74,74][68,68][48,48][49,49], ..., [48,48],[56,57],..., [48,54]

JDD014600009200986454

JDD014600009278066464

JDD014600009175764525

JDD014600008744501278

JDD014600008744501802

JDD014600009283643378

JDD014600009283279219

JDD014600009200968605 [74,74][74,74][68,68][48,48][49,49], ..., [48,48],[56,57],..., [48,57]

N=10の後の評価： 一定部分 = ☆ 0~9の数字文字 = ■

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
☆	☆	☆	☆	☆	☆	☆	☆	☆	☆	☆											
J	J	D	0	1	4	6	0	0	0	0											

20

30

40

50

フロントページの続き

審査官 田中 啓介

- (56)参考文献 国際公開第2013/179436(WO, A1)
特開2019-049967(JP, A)
特開2002-288592(JP, A)
- (58)調査した分野 (Int.Cl., DB名)
G06K7/00-7/14