(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2004-537768
(P2004-537768A)
(43)公表日　平成16年12月16日(2004.12.16)

(51)Int.Cl.[7]		F I			テーマコード（参考）
G06K	9/62	G06K	9/62	650C	5B064
G09G	3/20	G09G	3/20	612U	5C006
G09G	3/36	G09G	3/20	631V	5C080
G09G	5/00	G09G	3/20	680W	5C082
G09G	5/36	G09G	3/36		

審査請求 未請求　予備審査請求 有　（全 95 頁）　最終頁に続く

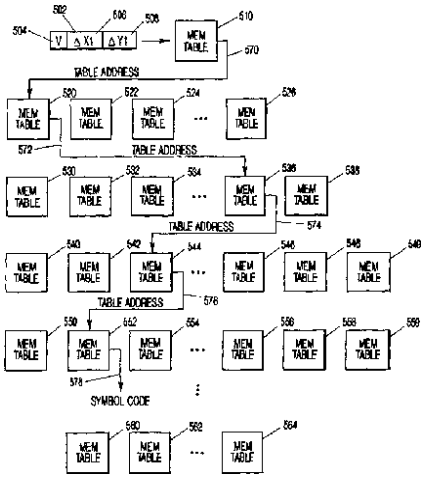
(21)出願番号　特願2001-548373(P2001-548373)
(86)(22)出願日　平成12年12月21日(2000.12.21)
(85)翻訳文提出日　平成14年7月1日(2002.7.1)
(86)国際出願番号　PCT/US2000/034682
(87)国際公開番号　WO2001/048729
(87)国際公開日　平成13年7月5日(2001.7.5)
(31)優先権主張番号　09/474,667
(32)優先日　平成11年12月29日(1999.12.29)
(33)優先権主張国　米国(US)
(31)優先権主張番号　09/741,391
(32)優先日　平成12年12月20日(2000.12.20)
(33)優先権主張国　米国(US)

(71)出願人　500575824
ハネウェル・インターナショナル・インコーポレーテッド
アメリカ合衆国ニュージャージー州０７９６２，モーリスタウン，コロンビア・ロード　１０１
(74)代理人　100089705
弁理士　社本　一夫
(74)代理人　100076691
弁理士　増井　忠弐
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男

最終頁に続く

(54)【発明の名称】カリグラフィック表示のためのシンボル認識および再生のためのアプリケーションによるパターン認識のためのシステム、方法および装置

(57)【要約】
本発明は、テーブル・ルックアップ・オペレーションのシーケンスを組み込むステート・マシンを使用するパターン認識のための技術を提供する。入力パラメータのシーケンスは、アプリケーションに特有のアルゴリズムに従って導出され、これらのオペレーションのための対応するメモリ・アドレスのシーケンスを生成する。メモリ・テーブルは、入力シーケンスに対応する階層構造において編成される。テーブル・データは、特定の入力パターンのライブラリを認識するように設計される。入力シーケンスが入力特有のパスを、メモリ・テーブルを通じてトレースし、これはライブラリにおけるパターンの１つが認識される（５７８）まで、あるいは入力シーケンスがライブラリ・パターンのいずれかと矛盾すると決定されるまで行われる。各ライブラリ・パターンについて、テーブル・データが、アプリケーションに特有である入力値における変化に対応するように設計される（たとえば、雑音および／または許容差による変化）。テーブル・データを、解析、シミュレーション、学習またはこれらの方法の組み合わせによって導出することができる



【特許請求の範囲】
【請求項１】
パターンを認識するための方法であって、
メモリ・アドレスのシーケンスを導出するステップであって、前記メモリ・アドレスは、少なくとも１つのメンバ・パターンを有する少なくとも１つの既知のパターン・セットに比較されるべきである入力を反映する値を含むことを特徴とする、ステップと、
前記メモリ・アドレスのシーケンスを、対応するメモリ・テーブル・ルックアップ・オペレーションのシーケンスに適用して結果を生じるステップと、
前記メモリ・テーブル・ルックアップ・オペレーションの前記結果に従って、前記入力が前記少なくとも１つの既知のパターン・セットのいずれかのメンバ・パターンへの推定合致であるとして認識されるかどうかを決定するステップと、
を含む方法。
【請求項２】
前記入力が、前記少なくとも１つの既知のパターン・セットのいずれかのメンバ・パターンへの推定合致であるとき、複数のメモリ・テーブルに、予想されたメモリ・アドレス値のシーケンスを反映するデータをロードするステップをさらに含み、少なくとも一部の前記複数のメモリ・テーブルが、前記メモリ・テーブル・ルックアップ・オペレーションのシーケンスにおいて使用されることを特徴とする、請求項１に記載の方法。
【請求項３】
前記メモリ・アドレスのシーケンスを適用する前記ステップが、前記メモリ・アドレスのシーケンスを、前記複数のメモリ・テーブルからなる、かつ少なくとも１つのメモリ・テーブルを各レベルで有する多数のレベルを有するメモリ・テーブルの構造に適用するステップを含み、前記メモリ・テーブル・ルックアップ・オペレーションのシーケンスのために使用された対応するメモリ・テーブルのシーケンスが、前記メモリ・テーブルの構造の少なくとも一部の前記複数のレベルを通じて進行することを特徴とする、請求項２に記載の方法。
【請求項４】
前記メモリ・テーブル・ルックアップ・オペレーションの前記結果に従って、前記入力が前記少なくとも１つの既知のパターン・セットのうち少なくとも１つのメンバへの合致である確率を反映する少なくとも１つの値を確かめるステップをさらに含み、複数のメモリ・テーブルにデータをロードする前記ステップが、前記入力が前記少なくとも１つの既知のパターン・セットのうち少なくとも１つのメンバへの合致である前記確率を反映する結果を提供するためのデータのロードを含むことを特徴とする、請求項３に記載の方法。
【請求項５】
少なくとも１つのメンバ・パターンを有する前記少なくとも１つの既知のパターン・セットが、単一のパターンを含むことを特徴とする、請求項３に記載の方法。
【請求項６】
少なくとも１つのメンバ・パターンを有する前記少なくとも１つの既知のパターン・セットが多数のパターンを含み、前記メモリ・アドレスのシーケンスを適用する前記ステップが、前記メモリ・テーブルのシーケンスを含む前記特定のメモリ・テーブルを、少なくとも部分的には前記メモリ・アドレスの前記値から、前記テーブル・ルックアップ・オペレーションの前記結果によって選択することを含むことを特徴とする、請求項３に記載の方法。
【請求項７】
前記テーブル・ルックアップ・オペレーションの前記結果に従って、前記入力が前記少なくとも１つの既知のパターン・セットのいずれかのメンバ・パターンへの推定合致であるとして認識される場合についての前記入力への推定合致である、前記少なくとも１つの既知のパターン・セットのうち少なくとも１つの特定のメンバを識別するステップをさらに含む、請求項６に記載の方法。
【請求項８】

前記メモリ・テーブル・ルックアップ・オペレーションの前記結果に従って、前記入力が前記少なくとも１つの既知のパターン・セットのうち少なくとも１つのメンバへの合致である確率を反映する少なくとも１つの値を確かめるステップをさらに含み、複数のメモリ・テーブルにデータをロードする前記ステップが、前記入力が前記少なくとも１つの既知のパターン・セットのうち少なくとも１つのメンバへの合致である前記確率を反映する結果を提供するためのデータの前記ロードを含むことを特徴とする、請求項７に記載の方法。

【請求項９】

パターン認識装置であって、

メモリ・アドレスのシーケンスを生成するための変換システムであって、前記メモリ・アドレスは、少なくとも１つのメンバ・パターンを含む少なくとも１つの既知のパターン・セットに比較されるべきである入力を反映する値を含むことを特徴とする、変換システムと、

少なくとも１つの物理記憶装置内に存在する複数のメモリ・テーブルと、

前記メモリ・アドレスのシーケンスを、前記複数のメモリ・テーブルからの対応するメモリ・テーブルのシーケンスのアドレス入力に適用して、結果を生じるためにメモリ・テーブル・ルックアップ・オペレーションのシーケンスを実行するためのアドレス指定回路と、

前記メモリ・テーブル・ルックアップ・オペレーションのシーケンスからの前記結果に従って、前記入力が前記少なくとも１つの既知のパターン・セットのいずれかのメンバ・パターンへの推定合致であるとして認識されるかどうかを決定するための出力回路と、

を含む装置。

【請求項１０】

少なくとも１つのセットのデータ値が前記複数のメモリ・テーブルに格納され、前記少なくとも１つのセットのデータ値が、前記少なくとも１つの既知のパターン・セットのいずれかのメンバ・パターンへの推定合致である前記入力についての、予想されたメモリ・アドレス値のシーケンスを反映することを特徴とする、請求項９に記載の装置。

【請求項１１】

前記複数のメモリ・テーブルは、多数のレベルを有するメモリ・テーブルの構造を含み、前記複数のメモリ・テーブルのうち少なくとも１つを前記多数の各レベルで有し、前記メモリ・テーブルのシーケンスが、前記メモリ・テーブルの構造の少なくとも一部の前記複数のレベルを通じて進行することを特徴とする、請求項１０に記載の装置。

【請求項１２】

データを前記複数のメモリ・テーブルに格納するための格納回路をさらに含み、前記データが前記少なくとも１つのセットのデータ値を含むことを特徴とする、請求項１１に記載の装置。

【請求項１３】

前記少なくとも１つのセットのデータ値のうち少なくとも１つが、前記変換システムによって、少なくとも１つの既知のパターン・セットに合致する入力から生成される前記メモリ・アドレス値のシーケンスから導出されることを特徴とする、請求項１１に記載の装置。

【請求項１４】

前記メモリ・テーブル・ルックアップ・オペレーションのシーケンスからの前記結果がさらに、前記入力が前記少なくとも１つの既知のパターン・セットのうち少なくとも１つのメンバへの合致である確率を定量化する少なくとも１つの出力値を含むことを特徴とする、請求項１１に記載の装置。

【請求項１５】

メモリ・テーブル・ルックアップ・オペレーションを、前記メモリ・テーブルの構造の前記各多数のレベルで同時に実行するためのルックアップ回路と、

前記メモリ・アドレスのシーケンスを進めるための少なくとも１つのコントロール・クロ

ック信号と、
前記少なくとも１つのコントロール・クロック信号に応答して、新しいテーブル・ルックアップ・オペレーションのシーケンスを開始するためのシーケンシング回路であって、前記メモリ・アドレスのシーケンスにおける各位置が、前記メモリ・アドレスのシーケンスを前記少なくとも１つの既知のパターン・セットと比較するための、可能な開始位置として使用されることを特徴とする、請求項１１に記載の装置。
【請求項１６】
前記パターン認識装置が少なくとも１つの集積回路を含むことを特徴とする、請求項１１に記載の装置。
【請求項１７】
少なくとも１つのメンバ・パターンを有する前記少なくとも１つの既知のパターン・セットが、単一のパターンを含むことを特徴とする、請求項１１に記載の装置。
【請求項１８】
少なくとも１つのメンバ・パターンを含む前記少なくとも１つの既知のパターン・セットが多数のパターンを含み、前記メモリ・テーブル・ルックアップ・オペレーションのシーケンスからの前記結果が、前記メモリ・アドレスの前記値を反映し、これが、前記メモリ・テーブルのシーケンスを含む特定のメモリ・テーブルの決定において使用されることを特徴とする、請求項１１に記載の装置。
【請求項１９】
前記入力が認識された場合、前記入力への推定合致である、前記少なくとも１つの既知のパターン・セットのうち少なくとも１つの特定のメンバが、前記メモリ・テーブル・ルックアップ・オペレーションの前記結果によって識別されることを特徴とする、請求項１８に記載の装置。
【請求項２０】
メモリ・アドレスのシーケンスを生成するための前記変換システムがさらに翻訳システムを含むことを特徴とする、請求項１９に記載の装置。
【請求項２１】
メモリ・アドレスのシーケンスを生成するための前記変換システムがさらにスケーリング・システムを含むことを特徴とする、請求項１９に記載の装置。
【請求項２２】
メモリ・アドレスのシーケンスを生成するための前記変換システムがさらにベクトル回転システムを含むことを特徴とする、請求項１９に記載の装置。
【請求項２３】
メモリ・アドレスのシーケンスを生成するための前記変換システムがさらに、少なくとも１つの離散高速フーリエ変換を含むことを特徴とする、請求項１９に記載の装置。
【請求項２４】
データを前記複数のメモリ・テーブルに格納するための格納回路をさらに含み、前記データが前記少なくとも１つのセットのデータ値を含むことを特徴とする、請求項１９に記載の装置。
【請求項２５】
前記少なくとも１つのセットのデータ値のうち少なくとも１つが、前記変換システムによって、少なくとも１つの既知のパターン・セットに合致する入力から生成される前記メモリ・アドレス値のシーケンスから導出されることを特徴とする、請求項１９に記載の装置。
【請求項２６】
前記メモリ・テーブル・ルックアップ・オペレーションのシーケンスからの前記結果が、さらに、前記入力が前記少なくとも１つの既知のパターン・セットのうち少なくとも１つのメンバへの合致である確率を定量化する少なくとも１つの出力値を含むことを特徴とする、請求項１９に記載の装置。
【請求項２７】

メモリ・テーブル・ルックアップ・オペレーションを、前記メモリ・テーブルの構造の前記各多数のレベルで同時に実行するためのルックアップ回路と、
前記メモリ・アドレスのシーケンスを進めるための少なくとも１つのコントロール・クロック信号と、
前記少なくとも１つのコントロール・クロック信号に応答して、新しいテーブル・ルックアップ・オペレーションのシーケンスを開始するためのシーケンシング回路であって、前記メモリ・アドレスのシーケンスにおける各位置が、前記メモリ・アドレスのシーケンスを前記少なくとも１つの既知のパターン・セットと比較するための、開始位置として使用されることを特徴とする、シーケンス回路と、
を含む、請求項１９に記載の装置。

【請求項２８】
前記パターン認識装置が少なくとも１つの集積回路を含むことを特徴とする、請求項１９に記載の装置。

【請求項２９】
表示システム装置であって、
少なくとも１つの表示ピクセルの２次元アレイを含む表示デバイスと、
特定の表示イメージを反映するピクセル値を格納する、少なくとも１つの２次元イメージ・メモリと、
前記表示デバイスを、前記少なくとも１つの２次元イメージ・メモリに格納された前記ピクセル値からリフレッシュする回路と、
少なくとも１つの入力信号からなる信号入力セットであって、前記信号入力セットのための入力値の経時的な変化が、前記表示デバイス上に表示するためのシンボルを反映することを特徴とする、信号入力セットと、
前記信号入力セットからの第１のシンボル・セットの前記シンボルを認識し、さらに前記第１のシンボル・セットの認識されたシンボルを識別する出力を含む、パターン認識装置と、
第２のシンボル・セットの前記シンボルのうちいずれかを前記少なくとも１つの２次元イメージ・メモリに、独立してレンダリングするレンダリング回路と、
前記パターン認識装置からの前記出力に応答して、前記レンダリング回路による前記第２のシンボル・セットからの前記シンボルのレンダリングを命令する、インターフェイシング回路と、
を含む装置。

【請求項３０】
前記信号入力セットが、アナログＸ偏向入力、アナログＹ偏向入力、および少なくとも１つのビデオ入力信号を含むことを特徴とする、請求項２９に記載の装置。

【請求項３１】
前記アナログＸ偏向入力のタイムサンプリングされた値を、Ｘデジタル値のシーケンスに変換する第１の回路と、
前記アナログＹ偏向入力のタイムサンプリングされた値を、Ｙデジタル値のシーケンスに変換する第２の回路と、
前記少なくとも１つのビデオ入力信号および前記Ｘデジタル値のシーケンスおよび前記Ｙデジタル値のシーケンスを、アドレス値のシーケンスに変換し、前記アドレス値のシーケンスを入力として前記パターン認識装置に提供する、第３の回路と、
を含む、請求項３０に記載の装置。

【請求項３２】
前記信号入力セットがラスタ・ビデオを含むことを特徴とする、請求項２９に記載の装置。

【請求項３３】
前記第１のシンボル・セットおよび前記第２のシンボル・セットが同じであることを特徴とする、請求項３０に記載の装置。

【請求項３４】
前記信号入力セットの前記入力値の前記経時的な変化が、前記第１のシンボル・セットのシンボルであるとして認識されないシンボルを反映するとき、前記少なくとも１つのビデオ入力信号を反映するピクセル値を少なくとも１つの２次元イメージ・メモリに、前記アナログＸ偏向入力および前記アナログＹ偏向入力によって決定された位置で書き込む、直接格納手段をさらに含む、請求項３０に記載の装置。
【請求項３５】
前記パターン認識装置がさらに、
少なくとも１つの物理記憶装置内に存在する複数のメモリ・テーブルと、
前記アドレス値のシーケンスを、前記複数のメモリ・テーブルからの対応するメモリ・テーブルのシーケンスのアドレス入力に適用して、前記第１のシンボル・セットのうち認識されたシンボルを識別する前記出力を含む結果を生じるために、メモリ・テーブル・ルックアップ・オペレーションのシーケンスを実行するためのアドレス指定回路と、
をさらに含む、請求項３０に記載の装置。
【請求項３６】
データ値のセットが前記複数のメモリ・テーブルに格納され、前記データ値のセットが、前記信号入力セットについての前記入力値が前記第１のシンボル・セットのいずれかのシンボルへの推定合致である場合についての、前記予想されたアドレス値のシーケンスを反映することを特徴とする、請求項３５に記載の装置。
【請求項３７】
前記複数のメモリ・テーブルは、多数のレベルの構造に編成され、前記複数のメモリ・テーブルのうち少なくとも１つを前記多数の各レベルで有し、前記メモリ・テーブルのシーケンスが、前記構造の前記多数のレベルを通じて進行することを特徴とする、請求項３６に記載の装置。
【請求項３８】
少なくとも１つの表示ピクセルの２次元アレイを含む表示デバイスと、
特定の表示イメージを反映するピクセル・データを格納するための少なくとも１つの２次元アレイをそれぞれ含む、少なくとも３つのイメージ・メモリと、
前記表示デバイスを、前記少なくとも３つのイメージ・メモリのうち１つに格納された前記ピクセル・データからリフレッシュするためのリフレッシュ回路と、
前記少なくとも３つのイメージ・メモリのうち１つに格納された前記ピクセル・データをクリアするためのリセット回路と、
更新されたピクセル・データを前記少なくとも３つのイメージ・メモリのうち１つに書き込むための更新回路であって、前記更新されたピクセル・データは、更新された表示イメージを反映することを特徴とする、更新回路と、
前記少なくとも３つのイメージ・メモリの機能を周期的に交換するためのコントロール回路であって、直前に更新されたイメージ・メモリが使用されて前記表示デバイスがリフレッシュされ、前記表示デバイスをリフレッシュするために先に使用されたイメージ・メモリがクリアされ、更新されたピクセル・データが、クリアされるイメージ・メモリに書き込まれることを特徴とする、コントロール回路と、
を含む表示システム装置。
【発明の詳細な説明】
【０００１】
発明の背景
発明の分野
本発明は一般にパターン認識の分野に関する。より詳細には、本発明は、リアルタイム・アプリケーションにおいて使用可能なパターンまたはシンボル認識のためのシステム、方法および装置に関する。
背景技術
パターン認識には、いくつかのアプリケーションおよび手法がある。たとえば、ニューラ

ル・ネットワークを、文字認識に使用することができる。しかし、ニューラル・ネットワーク用の大量のオフザシェルフ半導体デバイスは使用可能ではない。したがって、ニューラル・ネットワークを使用する実施は相対的に高価になる可能性があり、これらにより使用されるパーツが陳腐化するようになる可能性があるという固有のリスクが生じる。また、通常、ニューラル・ネットワーク・デバイスのコストを下げるために行われる折り合いのため、これらがいくつかのリアルタイム・アプリケーションのためのパフォーマンス要件を満たさない可能性もある。同様に、デジタル信号プロセッサ（ＤＳＰ）もパターン認識に使用することができる（たとえば、時間または周波数領域において実行された相関オペレーションによる）。しかし、ＤＳＰも高価になる可能性があり、いくつかのリアルタイム・アプリケーションによって必要とされるパフォーマンス・レベルを達成しない可能性がある。したがって、高パフォーマンスを相対的に低コストで、パーツが陳腐化する著しいリスクがなく達成することができる、パターン認識の代替手法を有することが有用となる。

【０００２】

多数の表示システムは陰極線管（ＣＲＴ）を使用するが、新しいＣＲＴは必ずしも、修理あるいはこれらの表示システムと置き換えることができるわけではない。したがって、これらのディスプレイを、新たに設計された表示システムで置き換えるための要求があり、これは液晶ディスプレイ（ＬＣＤ）技術など、フラットパネル表示技術を利用するものである。しかし、既存のＣＲＴベースのディスプレイへの電気的インターフェイスはしばしば、ＬＣＤベースのディスプレイの要件との互換性がなく、これは、これら２つの技術の間の違いのためである。これら２つの技術の主な非互換性は、各表示タイプの個々のピクセルが照度について選択される方法である。ＬＣＤ技術が、離散的に個々のピクセルを選択するためのメカニズムを使用するのに対して、ＣＲＴ技術はアナログ電圧信号を使用して１つまたは複数の電子ビームを表示面上で位置付ける。

【０００３】

エビオニクス表示など、リアルタイム・アプリケーションのための表示処理システムは一般に、ハードウェア・ベクトル生成（すなわち、線画）機能を組み込んでいる。このハードウェアは、Ｘ（水平）およびＹ（垂直）表示画面位置を選択するためのデジタル出力、および、選択された強度および／または色を指定するためのデジタルビデオを提供する。分離したビデオ・オン／オフ出力も提供することができる。カリグラフィック／ストローク・タイプのＣＲＴディスプレイでは、Ｘ、Ｙおよびビデオ信号がアナログに変換されて、ＣＲＴ電子ビームの位置および強度が直接コントロールされる。次いで、シンボル・ジェネレータ・ハードウェアが、そのコマンドを約６０ＨＺのレートで実行することによって、表示を直接リフレッシュする。動的な表示フォーマットでは、コマンド・ライン／プログラムが、シンボロジのスムーズな動きを達成するために十分高いレートで更新されるべきである。これには、２０または３０ＨＺの更新レートが必要となる可能性がある。

【０００４】

ＬＣＤ技術では、あるいはラスタ・タイプのＣＲＴベースのディスプレイでは、表示画面が２次元フレーム・バッファ・メモリ、すなわち「イメージ・メモリ」からリフレッシュされる。イメージ・メモリは、表示画面のピクセルの厳密な２Ｄマッピングにすることができ、これにより各メモリ位置が、その対応するピクセルの色／強度を決定するコードを保持する。次いで、シンボル・ジェネレータのデジタルＸおよびＹ出力が使用されて、２Ｄイメージ・メモリがアドレス指定される。ベクトルは、直接表示画面上ではなくイメージ・メモリに書き込まれ／描かれる。リアルタイム・アプリケーションでは、シンボル・ジェネレータが使用されて、あるイメージ・メモリにおける表示フォーマットが更新され、以前の更新を含む別のイメージ・メモリが同時に使用されて表示画面がリフレッシュされる。いくつかのＣＲＴベースのディスプレイは、ストロークとラスタの間でハイブリッド・モードにおいて交替するように設計され、これはストローク・シンボロジをラスタ・ビデオ上にオーバーレイするためである。

【０００５】

エビオニクス表示システムはしばしば、表示機能およびシンボル・ジェネレータ／プロセッサ機能のための、別々の物理ユニットを使用する。ストロークおよびハイブリッドＣＲＴディスプレイはアナログ・インターフェイスを、シンボル・ジェネレータと表示ユニットの間で利用することができる。このようなＣＲＴベースのディスプレイが新しいＬＣＤベースのユニットで置き換えられた場合、シンボル・ジェネレータを置き換えるためにもひどく高いコストがかかる可能性がある。次いで、新しいＬＣＤベースの表示ユニットを、既存のアナログ・インターフェイスにより機能するように設計するべきである。しかし、ＬＣＤ技術は離散的タイプのピクセル・アドレス指定メカニズムを使用するので、これは表示されたシンボロジの品質を落とす傾向がある。

【０００６】

ストローク・タイプのアナログ入力を有するＬＣＤベースの表示ユニットの設計のための従来の手法は、ＣＲＴベースのディスプレイのオペレーションを本質的に真似るか、あるいはシミュレートすることである。アナログのＸおよびＹ偏向入力がデジタルに変換され、これを使用して２Ｄイメージ・メモリがアドレス指定され、ここにデジタル／デジタル化されたビデオが書き込まれる。したがって、ストローク入力が使用されて、シンボロジが直接ＣＲＴ表示画面上ではなく、イメージ・メモリに書き込まれる。次いで、このイメージ・メモリが続いてスキャンされ、読み取られて、ＬＣＤ画面をリフレッシュするためのビデオが提供される。デジタル化入力が、あるイメージ・メモリに書き込むために使用されるが、第２のメモリ（以前の更新を格納するもの）が、表示をリフレッシュするために使用される。この設計手法では、アナログ・インターフェイスが、表示されたイメージへの誤り／異常に寄与する。これらは、１）アナログＸおよびＹ信号上で、外部および内部ソースから誘導された電磁干渉（ＥＭＩ）、２）アナログ回路内で生成された広帯域の熱雑音、３）サンプリング・オペレーションにおける位相雑音、および４）アナログ－デジタル（Ａ－Ｄ）変換からの量子化雑音の結果生じる。

【０００７】

これらのすべての雑音源は、表示されたシンボロジの品質を劣化させる傾向がある。ＥＭＩの重大度は、個々の応用例によって決まる。低域プレフィルタがサンプリングＡ－Ｄ変換オペレーションの前に使用されて、高周波ＥＭＩによる関心のある帯域幅へのエイリアシングが防止されることが好ましいが、より低周波のＥＭＩは、表示されたシンボロジの品質を劣化させる。熱雑音は、アナログ偏向入力のデジタル化されたサンプルにおいて、小さいランダムなずれを引き起こし、これはガウス分布をノルムの周囲に示す。熱雑音の影響は、入力をオーバーサンプリングし（すなわち、ダウンストリーム・オペレーションによって要求されるよりも高いレートでサンプリングする）、かつ次いで低域デジタル・フィルタを使用することによって、最小限にすることができる。位相雑音は、サンプリング時間におけるジッタの結果生じ、これはＡ－Ｄコンバータ回路内で、またサンプリング・クロックにおけるジッタからも生じる可能性がある。しかしこの応用例では、より著しい誤りが、シンボル・ジェネレータにおけるクロックと相対的に、サンプリング・クロックの一貫性のない位相から生じる可能性がある（すなわち、表示フォーマットの異なる更新における）。量子化雑音の結果生じる表示の異常は、高解像度のＡ－Ｄコンバータを使用することによって最小限にすることができる（たとえば、１２ビット・コンバータは、ディスプレイのピクセル解像度より著しく高い解像度を提供することができる）。

【０００８】

表示の異常を、静的または動的として類別することができる。あらゆる表示更新において同じである雑音またはひずみは、静的として類別される。その位置は、表示画面または表示画面上のシンボルに相対的に固定される（すなわち、シンボル自体が動いている可能性がある）。静的異常の一実施例は、個々のシンボルに対するひずみであり、これは偏向入力の攻撃的な低域フィルタリングの結果生じる（たとえば、ひどく丸みのある角を有するシンボルという結果となる）。動的異常は、画面全体で、かつ／またはシンボロジを通じて移動する雑音またはひずみである可能性がある。これも特定の画面または特定のシンボル位置に位置付けることができ、周期的あるいはランダムに示すことができる。人は動き

／変化の検出に熟達しているので、動的異常は静的異常より好ましくない可能性がある。低域フィルタリングの量が増すと、ＥＭＩの結果生じる動的異常を低減することができる。しかしこれにより、シンボロジについての静的ひずみのレベルが増大する。

【０００９】

アナログ入力を有するＬＣＤベースのカリグラフィック表示のためのこの従来の設計手法では、好ましくない表示異常を生じる結果となるので、この問題を有していないある他の設計手法を使用することが有益となる。

発明の簡単な概要

以下の本発明の概要は、本発明に固有の革新的な特徴のいくつかを理解しやすくするために提供し、完全な記載となるように意図するものではない。本発明には多数の応用例があるが、この概要はカリグラフィック表示の応用例に焦点を置く。本発明の様々な態様の完全な理解は、明細書全体、特許請求の範囲、図面および要約書を全体として取り扱うことによって得ることができる。

【００１０】

本発明は、半導体記憶装置を使用して実施することができる、パターン認識ソリューションを提供する。メモリ・チップは、非常に低コストで入手可能である、非常に大量の商品パーツである。さらに、メモリ・チップの密度（すなわち、単一の集積回路におけるビット数）は、半導体技術における進歩により劇的に増大してきており、場合によってはこの増大が続くであろう。したがって、本発明は、非常に高レベルのパフォーマンスを相対的に低コストで達成する、パターン認識問題のソリューションを提供する。

【００１１】

本発明は、表示されたシンボロジの品質を向上させる、ＬＣＤベースのカリグラフィック表示ユニットの設計の代替手法を提供する。この代替手法では、アナログ偏向入力のデジタル化されたサンプルが、ビデオ入力と共に、表示されるべきである個々のシンボルを認識するために監視される。次いで、これらのシンボルを表示ユニット内で、完全にデジタル手段を使用して再生することができ、これにより、表示されたシンボロジにおける、アナログ・インターフェイスでの雑音および低域フィルタリングからのいかなる劣化も回避することができる。この手法は、表示ユニットへのアナログ偏向およびビデオ入力からの個々の表示シンボルの信頼性のある認識のためのメカニズムを、雑音がこれらの入力上に存在するときでさえ、必要とする。シンボル認識機能はリアルタイムで、表示ユニットへの入力により動作するので、本発明はソフトウェアを介してではなく、この応用例用のハードウェアを介して実施されることが好ましい。リアルタイム・パフォーマンスのための必要性でも、表示入力を同時のベースで、シンボル・ライブラリにおけるシンボルのいずれか１つについての予想された入力と比較するソリューションが必要であるようになる（すなわち、一度に１つのライブラリ・シンボルとの順次の比較ではない）。本発明は、このリアルタイム・アプリケーションの厳しい要件を満たすことができる、シンボル／パターン認識問題のための低コストのソリューションである。

【００１２】

本発明の一実施形態は、パターン認識機能を、テーブル・ルックアップ・オペレーションのシーケンスを使用するステート・マシンによって実施し、これらは入力のシーケンスにおけるパターンを認識するように設計される。認識されるパターン（すなわち、パターンのライブラリ）は、テーブル・ルックアップ・オペレーションによって使用されるメモリ・テーブルにおいて格納されたデータの導出において、使用される。テーブルは階層またはツリー状の構造において編成され、これはこの構造の各レベルについて一意のセットのメモリ・テーブルを有する。このパターン認識メカニズムへの入力は、メモリ・アドレスのタイムシーケンスを含み、これを、１つまたは複数のアナログ入力信号の、タイムサンプリングかつデジタル化された値から導出することができる。これは、アナログＸおよびＹ入力を有する、ＬＣＤベースのカリグラフィック表示ユニットの応用例の場合となる。他の応用例は、このメモリ・アドレスのシーケンスを異なる手段によって導出することができる。

【００１３】
ＬＣＤベースのカリグラフィック表示では、ＸおよびＹ入力値が通常、各表示シンボルのレンダリング前のある期間に渡って一定である（すなわち、雑音による小さい変化を除く）。これは、古いＣＲＴベースの表示ユニットによって、それらのアナログ偏向増幅器の安定化を可能にするために必要とされた。この状態を検出し、新しい表示シンボルがレンダリングされる前にシンボル認識メカニズムをリセットするために使用することができる。加えて、デジタル化されたＸおよびＹ入力を、この間隔中にフィルタリング／平均化することができる。これらのフィルタリング値は、保存され、レンダリングされるシンボルのための正確な表示画面位置を提供する。
【００１４】
デジタル化されたＸ、Ｙおよびビデオ入力サンプルのすべてが、シンボル認識メカニズムのためのメモリ・アドレスのタイムシーケンスの生成において使用される必要があるのではない。たとえば、これらのアドレスを生成するための１つのアルゴリズムは、ビデオ入力が状態を変更するすべての入力サンプル、および、ビデオ入力におけるこれらの変更の間の間隔において選択された数の入力サンプルを使用する。これらの各サンプルについて、次いで関連付けられたメモリ・アドレスを、１）サンプルの直前のビデオ状態を表す値、２）サンプリングされたＸ入力の相対的値、および３）サンプリングされたＹ入力の相対的値の連結によって導出することができる。シンボルのレンダリングの前に、ＸおよびＹ表示画面位置が保存されることを想起されたい。次いで、各Ｘ入力サンプルについての相対的Ｘ位置を、入力サンプルのＸ値から保存されたＸシンボル位置を差し引くことによって導出することができる。相対的Ｙ値を、類似の方法で導出することができる。したがって、次いで、選択された入力サンプルが、対応するメモリ・アドレスのタイムシーケンスを生成するために使用される。次いで、このメモリ・アドレスのシーケンスが、シンボル認識オペレーションへの入力として使用される。
【００１５】
シーケンスにおける第１のメモリ・アドレスが、メモリ・テーブルの階層における第１のレベルに存在する単一のメモリ・テーブルをアドレス指定するために使用される。このアドレスの値、およびよってそれが表すサンプルのための入力の値が、ライブラリにおける１つまたは複数のシンボルのレンダリングと無矛盾である場合、メモリ・テーブルのこのアドレスからの出力が、特定のレベル２テーブルをポイントし、これにより、無矛盾のシンボルの可能な認識が可能となる。次いで、シーケンスにおける次のメモリ・アドレスが、階層における第２のレベルでのこの指定されたテーブルをアドレス指定するために使用される。このプロセスが進行するにつれて、入力サンプルのシーケンスは、それらの関連付けられたメモリ・アドレスにより、メモリ・テーブルの構造を通じてパスを向け、これはシンボルが認識されるまで、あるいは、入力シーケンスがライブラリ・シンボルのいずれかと矛盾することが決定されるまで行われる。
【００１６】
３つのタイプの出力が、メモリ・テーブルについて可能である。完了された入力シーケンスが、曖昧でなく、ライブラリにおけるシンボルの１つに相関するものとして認識されたとき、そのシンボルのためのコードがメモリから出力される。不完全な入力シーケンスが、シーケンスにおけるそのポイントまで、１つまたは複数のライブラリ・シンボルに無矛盾であるとき、メモリ出力が、階層における次のレベルでの特定のテーブルをポイントし、これが入力シーケンスの次のサンプルをこの可能性に対してテストする。所与の入力サンプルが、入力シーケンスにおけるいずれかのポイントで、入力シーケンスがいずれかのライブラリ・シンボルに無矛盾でないことを確立したとき、メモリ・テーブルの出力がこの状態を指示する。入力シーケンスが、いずれかのライブラリ・シンボルと矛盾するように決定されたとき、デジタル化された入力サンプルを、シンボルをイメージ・メモリに直接レンダリングするために使用することができる（すなわち、シンボル再生メカニズムを効果的にバイパスする）。しかし、動的表示異常を生じる可能性をなくすため、シンボル認識メカニズムは、すべてのライブラリ・シンボルを（すなわち、雑音が入力上に存在す

るときでさえ）信頼性を有して、かつ無矛盾に認識する必要がある。これには、テーブルが、最悪の場合の雑音状態下で遭遇する可能性のある入力サンプルの値における変化を可能にするように、設計されることが必要である。

【００１７】

本発明の新しい特徴は、当業者には、以下の本発明の詳細な説明の考察において明らかになり、あるいは本発明の実施によって知ることができる。しかし、本発明の詳細な説明および提示した特定の実施例は、本発明のある実施形態を指示するが、例示のためにのみ提供されるものであり、これは、本発明の精神および範囲内の様々な変更および修正が、以下に続く本発明の詳細な説明および特許請求の範囲から、当業者に明らかになるからであることを理解されたい。

【００１８】

添付の図面では、類似の参照番号はいくつかの図を通じて等しいかあるいは機能的に類似の要素を指し、これらは本明細書に組み込まれ、その一部を形成し、さらに本発明を例示し、本発明の詳細な説明と共に、本発明の原理を説明するための機能を果たす。

発明の詳細な説明

本発明は、パターン認識のためのシステム、方法および装置を提供する。本発明をいくつかの応用例において使用することができるが、以下の詳細な説明は、アナログ偏向入力を有するＬＣＤベースのカリグラフィック（ｃａｌｉｇｒａｐｈｉｃ）表示ユニットに関する。アナログ偏向入力をビデオ入力と共に監視することによって、表示されるべきシンボロジを認識することができ、次いで表示ユニット内に、完全なデジタル手段によって再生することができる。これにより、アナログ入力での雑音および低域フィルタリングからのシンボロジにおける異常およびひずみがなくなる。本発明を、この特定の応用例における使用に関して記載するが、本発明を、パターンまたはシンボル認識を必要とする他の応用例における使用のために適合できることを理解されたい。

【００１９】

カリグラフィック表示の応用例では、デジタル化されたＸおよびＹ偏向値がビデオと共に使用されて、レンダリングされた各シンボル上のポイントのシーケンスが選択される。次いで、このポイントのシーケンスが、表示ユニットに格納されるライブラリにおけるすべてのシンボルについての予想されたポイントのシーケンスに対して比較される。入力シーケンスが、ライブラリにおけるこれらのシンボルの１つのレンダリングであるとして認識された場合、デジタル化された入力サンプルのストリームを廃棄することができ、シンボルを、表示ユニット内に格納されたベクトルのリストからレンダリングすることができる。入力シーケンスが、内部に格納されたライブラリにおけるシンボルの１つであるとして認識されなかった場合、シンボルを直接イメージ・メモリに、デジタル化された入力サンプルからレンダリングすることができる。これには、デジタル化された入力サンプル（偏向およびビデオ）がＦＩＦＯバッファに保存されることが必要であり、これはシンボル認識オペレーションのための時間を考慮するためである。シンボル・ライブラリにあるものとして認識されるシンボルは、表示ユニット内でデジタル・ハードウェアにより完全に再レンダリングされる。したがって、これらの表示されたシンボルの品質は、アナログ・インターフェイスでの雑音によって、あるいは低域フィルタリングによって劣化されない。

【００２０】

図１は、カリグラフィック表示の応用例のための本発明の一実施形態を示す。この表示ユニットは、フラットパネルＬＣＤ３４４を組み込むが、水平および垂直表示画面位置を選択するためにアナログ偏向入力を使用するＣＲＴベースの表示ユニットに置き換えられるように設計される。水平またはＸ偏向の入力信号２１８から高周波雑音が、直列レジスタ３０２および並列キャパシタ３０４からなる低域フィルタによって除去される。直列レジスタ３０６および並列キャパシタ３０８を有する類似の低域フィルタは、垂直またはＹ偏向の入力信号２１６上の高周波雑音を除去する。フィルタリングされた入力がサンプリングされ、周期的に偏向インターフェイス３１０でデジタル値に変換される。高サンプリング・レート（すなわち、オーバーサンプリング）が使用され、次いでサンプリングされた

値のストリームが低域デジタル・フィルタ３１２によって処理される。ビデオ入力２２０は多数の信号からなることができ、アナログ、デジタル、またはアナログおよびデジタルの混合にすることができる。ビデオ・インターフェイス３１６が、ビデオ入力２２０の「オフからオン」および「オンからオフ」への移行を検出する。これらの移行が使用されて、表示ユニット内の様々なオペレーションが同期化かつコントロールされる。ビデオ・インターフェイス３１６はまた、ビデオおよび偏向入力のタイミングにおけるいかなる格差も補償する。このような格差は、ＣＲＴベースのディスプレイに合わせて設計されたインターフェイスについて存在する可能性がある。ＣＲＴベースのディスプレイの偏向増幅器のための帯域幅は通常、ビデオ回路の帯域幅よりはるかに低い。したがって、表示画面で偏向を同期化させるために入力でのビデオの、ＣＲＴベースのディスプレイへの移行はしばしば遅延される。この場合、ビデオ・インターフェイス３１６がビデオ・タイミングを調節して、効果的にこの補償を除去する。

【００２１】

ストローク・タイプのＣＲＴベースのディスプレイの場合のように、シンボロジを直接表示画面上に書き込むのではなく、図１に示すディスプレイは、シンボロジをイメージ・メモリ３３６に書き込む。イメージ・メモリは２次元アレイとして編成され、これがしばしば１対１毎に、表示画面上のピクセルのアレイに対応する。したがって、イメージ・メモリへの入力アドレスは、水平またはＸの値および垂直またはＹの値からなる。いくつかの実現において、次いで、メモリにおける各位置が、表示画面の単一の対応するピクセルについてのビデオ情報を格納する。このビデオ情報は、ピクセル強度およびピクセル色を含むことができる。３つのイメージ・メモリ３３６、３３８、３４０の合計が使用される。シンボロジがイメージ・メモリのうちの１つ３３６に書き込まれている間、ＬＣＤ３４４が、先の表示リフレッシュ・サイクル上で書き込まれたシンボロジを保持する別のイメージ・メモリ３４０におけるビデオ・データによってリフレッシュされる。第３のイメージ・メモリ３３８から同時にシンボロジをクリアして、次のリフレッシュ・サイクル上で更新されたシンボロジを書き込むためにこれを使用できるようにすることができる。次いで、イメージ・メモリの機能が、各表示リフレッシュ・サイクルの最初に、循環ベースで交換される。

【００２２】

表示システムは従来２つのイメージ・メモリしか使用しておらず、それによりいかなる所与の時点でも、第２のイメージ・メモリが新しいシンボロジにより更新されている間に、一方のイメージ・メモリが表示のリフレッシュに使用される。更新オペレーションが完了した後、これらの２つのイメージ・メモリの機能が交換される。いくつかのシステムは、すべてではないが、この交換を表示リフレッシュ・オペレーションの完了に同期化させる。交換の後に続いて、更新オペレーションが最初にイメージ・メモリから、先に更新されたシンボロジ／イメージをクリアし、次いで、新しいシンボロジをイメージ・メモリに書き込む。このクリア・オペレーションが、書き込みオペレーションに続いて順次に実行される（すなわち、同時ではない）ので、これによりシステム・パフォーマンスが制限される。最低待ち時間（すなわち、更新されたシンボロジを表示するための時間遅延）および最大更新レートが、これら２つのオペレーションを順次に実行する要件によって制約される。

【００２３】

３つのイメージ・メモリを使用することによって、３つのイメージメモリ・オペレーション（すなわち、表示のリフレッシュ、イメージ・メモリから先のシンボロジをクリアすること、および新しいシンボロジを書き込むこと）をすべて同時に実行することができる。これにより、表示システムが、より低い待ち時間およびより高い更新レートを達成することができる。３つのイメージ・メモリを利用してパフォーマンスを向上させるこの技術は、ＬＣＤベースのカリグラフィック表示の応用例に限定されず、これを、コンピュータによりイメージ・メモリから生成されたシンボロジ／グラフィックスを有するディスプレイをリフレッシュするいかなる表示システムにおいても、使用できることを理解されたい。

本発明のこの特徴は、このようなすべての表示システムの応用例を包含することを意図する。

【００２４】

図１の表示システムでは、２つの異なるメカニズムが、シンボロジをイメージ・メモリに書き込むために使用される。これらのうち１つにおいて、アドレス値をイメージ・メモリ３３６に供給するためにサンプリングかつデジタル化されたＸおよびＹ入力値がＦＩＦＯパイプライン３２８およびセレクタ３２４を通過するようにされる。ビデオ・データ入力をイメージ・メモリ３３６に供給するためにサンプリングされたビデオ入力がＦＩＦＯパイプライン・バッファ３２６およびセレクタ３３０を通過するようにされる。ビデオ・データ入力がオンである（すなわち、オフ／ブランク状態でない）とき、これがイメージ・メモリ３３６へ、選択されたＸ／Ｙ位置で書き込まれる。この手法は、ＣＲＴベースのディスプレイのオペレーションをエミュレートするが、シンボロジは表示画面上に直接ではなく、イメージ・メモリに書き込まれる。しかし、表示されたシンボロジの品質は、アナログ偏向入力２１８および２１６上の雑音の結果として、この手法により損なわれる可能性がある。

【００２５】

シンボロジをイメージ・メモリに書き込むための第２のメカニズムにより、アナログ入力上の雑音の結果生じる表示品質の劣化がなくなる。イメージ・メモリに直接書き込むのではなく、サンプリングされた入力のデータ・ストリームが監視されて、表示されるべき特定のシンボルが検出される。次いで、検出されたシンボルが、表示ユニット内のデジタル手段によって再生される。したがって、これらが、アナログ・インターフェイスの結果生じる可能性のある異常なしに、レンダリングされる。

【００２６】

デジタル化されたＸおよびＹ値のストリーム、および対応するビデオ・データのストリームが最初にポイント選択メカニズム３１４に入力される。データ・ストリームにおけるあらゆるサンプル（すなわち、シンボル上のあらゆるポイント）が、シンボルを識別するために処理される必要はない。したがって、ポイント選択メカニズム３１４が使用されて、シンボル検出メカニズム３１８へ入力されるデータの量が低減される。ポイント選択アルゴリズムが、データ・ストリームにおいて、ビデオにおける移行が起こる（たとえば、ビデオがオフからオンへ、あるいはオンからオフへ切り替わる）サンプル／ポイントのすべてを選択する。相対的に長い間隔がビデオ移行の間に生じるとき、追加のポイントが選択される（たとえば、アルゴリズムが、ビデオ移行の間の間隔中にデータ・ストリームにおける４サンプル／ポイント毎に選択することができる）。選択されるポイントの数を十分高くして、シンボル・ライブラリにおけるすべてのシンボルの曖昧な検出が保証されるようにするべきである。

【００２７】

特定のライブラリ・シンボルが検出されるとき、ベクトル生成メカニズム３２０が使用されて、シンボルが、内部に格納されたライブラリにおける情報から再生される。これはベクトル・コマンドのシーケンスをレンダ・メカニズム３２２に出力する。次いで、レンダ・メカニズム３２２がこれらの各ベクトルをイメージ・メモリ３３６にレンダリングする。これはＸ／Ｙアドレスのシーケンスを、セレクタ３２４を介して、イメージ・メモリ３３６のアドレス入力に出力し、これはビデオ・データを、セレクタ３３０を介して、イメージ・メモリ３３６のデータ入力に出力する。ビデオ・データがイメージ・メモリへ、アドレス指定された各位置で、ベクトルのパスに沿って書き込まれる。所与のシンボルを生成するためのベクトル・コマンドが、ベクトルのエンドポイントの位置を表示上のシンボルの位置に相対的に指定する。各ベクトルのパスに沿ったポイントも、最初に相対的な位置として生成される。したがって、イメージ・メモリへの絶対的なＸおよびＹアドレスを生成するためにレンダリング・メカニズムがシンボルの画面位置をこれらの相対的な値に追加する。

【００２８】

表示されるべきシンボルが、ライブラリにおけるシンボルのうち１つであるとして認識されなかったとき、Ｘ、Ｙのデータ・ストリーム、およびビデオ・サンプルが使用されて、シンボルが直接イメージ・メモリ３３６へ、ＦＩＦＯパイプライン・バッファ３２６および３２８を介してレンダリングされる。ＦＩＦＯバッファ３２６および３２８は、シンボルを識別するか、あるいはシンボルがライブラリにないことを決定するために必要とされる間隔中に、データ・サンプルのストリームを保存するために必要とされる。

【００２９】

図１のブロック図に明示的に示さないが、ＬＣＤ技術に基づいたディスプレイは一般に、エイリアス除去機能の実施を必要とする。表示されたイメージは実際には、所望のイメージの空間的にサンプリングされた表現であり、表示されたイメージの品質は、シンボロジの空間解像度／周波数とディスプレイのピクセル解像度（すなわち、空間サンプリング周波数）の間のエイリアシング効果によって劣化される可能性がある。エイリアシングはストロークＣＲＴディスプレイでは問題ではなく、これは、ピクセル・アドレス指定のアナログ偏向メカニズムに固有である低域空間フィルタリングのためである。しかし、ＬＣＤベースのディスプレイは一般にエイリアス除去回路を組み込み、これは高品質のイメージを提供するためである。このために、ＸおよびＹ入力値のアナログ－デジタル変換は、ＬＣＤ画面のピクセル解像度よりも高い解像度を使用する。したがって、図１に示したデジタル・フィルタ３１２からの各ＸおよびＹ値は、整数および小数ビットを有する。これらの値の整数部が、イメージ・メモリ３３６をアドレス指定するために使用され、少数ビットは、エイリアシング除去のために使用されるサブピクセル・アドレス指定情報を表す。エイリアシング除去のための１つの手法は、ＸおよびＹサブピクセル／小数アドレス指定情報をイメージ・メモリ３３６に、ビデオ・データと共に書き込むことである。次いで、エイリアシング除去アルゴリズムを表示のリフレッシュのために生成されたビデオ上で実施するために、イメージ・メモリがスキャン／読み取られるとき、この情報を処理することができる。第２の手法は、エイリアシング除去機能を、イメージ・メモリの入力側で実施することである。この手法には、イメージ・メモリ３３６における追加の位置への書き込みオペレーションが必要である。整数ＸおよびＹの値によってトレースされるものに隣接あるいは近い位置に、より低い強度のビデオが書き込まれる。

【００３０】

図１に示す本発明の実施形態が、本発明を記載するための手段でしかなく、これは本発明のこの特定の応用例の単一の実施例のみを表すことを理解されたい。いくつかの設計は、たとえば、検出されたシンボルの再レンダリングにおける弧ならびにベクトルのための能力を含むことができる。また、いくつかの表示ユニットは、コンピュータにより生成されたシンボロジをカメラ・ビデオ上でオーバーレイできるように設計される。

【００３１】

図１に示す本発明の実施形態により、シンボルを、非常に高い品質を有する、内部で格納されたライブラリからレンダリングすることができる。しかし、表示画面上のこれらのシンボルの位置が、多数の表示の更新に渡って安定することが等しく重要である。不安定なシンボル位置では、動的な表示異常が生じる。幸いにも、ストローク表示のオペレーションは、正確にシンボル位置を決定することができる空間フィルタリング・オペレーションの実施と互換性がある。シンボル・ジェネレータがＸおよびＹ偏向を新しいシンボルの初期／開始の位置に再位置付けするとき、これはシンボルのレンダリング前に休止し、ＣＲＴベースのストローク表示ユニットの偏向増幅器が応答して正しい出力電圧を安定させるために十分な時間を有するようにする。この安定したＸおよびＹ入力値の時間間隔（すなわち、雑音レベル内の安定）を検出することができ、入力サンプルをこの期間中にフィルタリングして、多数の更新に渡って安定するＸおよびＹ位置の値を提供することができる。いくつかのストローク表示システムでは、英数字シンボルのストリングを単一の休止のみ（すなわち、ストリングにおける最初のシンボルの前のみの休止）によりレンダリングすることができる。この場合、表示ユニットを、追加のシンボルをストリングにおける第１のシンボルの位置に相対的に位置付けるように設計することができ、それにより、スト

リングにおけるすべてのシンボルについて安定した画面位置を提供する。シンボルのレンダリング前のこの安定したＸおよびＹ偏向入力の期間の検出を使用して、シンボル認識メカニズムをリセットすることもでき、これはステート・マシンとして動作する。

【００３２】

動的異常の発生をなくすために、シンボル認識メカニズムが、最悪の場合の雑音状態下で、ライブラリにおけるいかなるシンボルも無矛盾に認識するべきである。加えて、これはいくつかの他の入力シーケンスを、ライブラリにおけるシンボルの１つと間違えるべきではない。すなわち、いかなる入力シーケンス（ライブラリ・シンボルまたは非ライブラリ・シンボル）も、最悪の場合の雑音状態下でさえ、ライブラリにおけるシンボルの１つの検出／認識のためのしきい値レベルに到達するべきではない。シンボル認識機能はリアルタイムで、デジタル化された入力サンプルのストリームにより動作することが必要とされるので、この機能をソフトウェアではなくハードウェアで実施することが必要である可能性がある。リアルタイム・パフォーマンスの必要性でも、入力ポイントのシーケンスを、同時のベースで、ライブラリにおけるすべてのシンボルについて予想されたシーケンスと比較する（すなわち、一度に１つのライブラリ・シンボルとの順次の比較ではない）ソリューションを必要とするようである。本発明は、上のすべての要件と互換性のあるパターン認識の方法である。

【００３３】

パターン／シンボル認識メカニズムを、英数字シンボル「Ｎ」を検出するためのプロセスを示す一実施例によって説明する。このプロセスの最初のステップは、入力データ・ストリームから、レンダリングされているシンボル上の選択されたポイントに対応するサンプルを選択することである。これらが使用されて、シンボル認識メカニズムへの入力として使用されるメモリ・アドレスのシーケンスが生成される。図２は、２つの英数字シンボルの、各シンボルの標準の向きについて、かつ各シンボルの回転された向きについてのレンダリングを示す。図２Ａは、標準の正しい向きにおけるシンボル「Ｎ」４００のレンダリングを示し、図２Ｂは、標準の正しい向きにおけるシンボル「Ｌ」４２０のレンダリングを示す。図２はまた、２つの座標系の重ね合わせも示す。直交座標系は、水平軸Ｘおよび垂直軸Ｙによって示される。半径座標ｒおよび角座標θは、重ねあわされた極座標系を示す。

【００３４】

図２Ａの記号「Ｎ」４００は５つのベクトルによって描画され、また図２Ａ中の参照符号は、これらベクトルの終点を示している。記号「Ｎ」４００が描画される前、図２Ａの点４０２におけるディスプレイへのＸおよびＹの偏向の入力は、ノイズを除いては安定している。先に言及したように、この安定した状態は、フィルタリング後の点４０２のＸおよびＹの値を得るために検出する。記号の認識機構もリセットされる。この場合には点４０２である、記号を描画する際の初期／開始点が記号の位置を定義する。この点は、記号を回転させて描画する際には回転の中心としても使用される。

【００３５】

記号「Ｎ」４００は、まず図２Ａの点４０２から点４０４に空のベクトルを描画することによって描画される。「空のベクトル」はビデオをオフ状態にして描く。このベクトルは、ビデオがオフであることを表すために図２Ａでは点線で示す。ビデオをオンに切り替える３つの連続したベクトルが、この最初の空のベクトルの後に続く。ベクトルは、点４０４から点４０６に、点４０６から点４０８に、そして点４０８から点４１０へと描画される。最後に、もう１つの空のベクトルを点４１０から点４１２に描画する。この後者のベクトルは、英数文字ストリングの描画との互換性のために含まれ、これは、ストリングの次の記号の位置への偏向の位置を決定する。図２Ｂは、記号「Ｌ」４２０の描画を示している。この記号は点４２２に位置し、まず点４２２から点４２４に空のベクトルを描画することによって描画される。これに続き、ビデオがオンの状態で２つのベクトルを描画する。点４２４から点４２６に描画されたベクトルの後に、点４２６から点４２８へのベクトルが続く。この後に点４２８から４３０への空のベクトルが続き、これはストリング中

の次の記号への偏向の位置を決定する。

【００３６】

記号認識機能への入力を生成するために、描画された記号上の点を選択するのには、様々なアルゴリズムを用いることができる。効果的な技術は、まずビデオ遷移が起こる（すなわち、オフからオンへの遷移、またはオンからオフへの遷移）入力ストリームですべてのサンプルを選択することである。この結果、図２Ａの描画された「Ｎ」４００の点４０４と４１０が選択されることになる。記号の描画後に中断が発生した場合は、サンプリングしたＸおよびＹの偏向のこの安定した期間の検出を使用して、記号の終点（すなわち図２Ａの点４１２）を選択することができる。次いで、最初に選択したこれらの点の間に追加の点を選択する。単純で効果的なアルゴリズムは、周期的に追加の点を選択するものである。例えば、ビデオ遷移に続いて４つごとにサンプルを選択することができる。しかし、別のアルゴリズムが可能である。１つの可能性は、連続したベクトル間の頂点を選択するものである。この結果、図２Ａの点４０６と点４０８が選択される。頂点の選択と周期的な点の選択の組み合わせを使用するアルゴリズムも考案することができる。

【００３７】

図２Ａのベクトル間の頂点は、サンプリングしたＸおよびＹの入力値の変化を時間にわたって監視することによって（すなわち、ΔＸ／ΔｔとΔＹ／Δｔの値を監視することによって）検出することができる。これらの値は基本的に、ノイズと、アナログ－デジタル変換の際の制限された解像度を除いては、所与のベクトルを描画する間一定である。しかし、これらの「速度値」は、描画されたベクトルの方向を反映する。これらの値の１つまたは両方の変化を使用して、２つのベクトル間の頂点／隅の位置を検出することができ、したがって描画した記号上の追加点を選択することができる。

【００３８】

記号「Ｎ」４００の各頂点は、実質的に異なる方向を有するベクトルの対によって形成される。これにより、これら頂点の検出が容易になる。ほぼ同じ方向を有する２つのベクトルの頂点の検出はこれに比べると難しく、同じ記号の複数の描画における点の選択の整合性が低くなりかねない。特にノイズがある場合にはこのことが当てはまる。したがって、本発明の実際的な実装では、周期的な追加点の選択の手法が好ましいと思われる。ただし、平易さを考えて、この例では頂点選択のアルゴリズムを使用する。このアルゴリズムを用いると、図２Ａの描画された「Ｎ」４００で、点４０４、４０６、４０８、４１０、および４１２が選択されることになる。

【００３９】

記号認識プロセスの次のステップは、メモリ・アドレスのシーケンスを生成することである。選択されたそれぞれの点から、単一のメモリ・アドレスを得る。次いで記号認識プロセスは、一連のテーブル・ルックアップ・オペレーションのために、このアドレス入力のシーケンスを使用する。この動作のためのメモリ・テーブルは階層状のレベルに構成され、この構造の各レベルには１つまたは複数のテーブルがあり、また入力シーケンスの各メモリ・アドレスに対して別個のレベルがある。入力シーケンスの最初のアドレスは、この構造の第１レベルにあるメモリ・テーブルのアドレス指定に使用される。描画された記号で最初に選択される点の位置がライブラリの１つまたは複数の記号と整合する場合、アドレス指定されたメモリ・テーブルの出力は、この可能性を見込んだ構造のレベル２にある特定のテーブルをポイントすることになる。次いで、入力シーケンスの２番目のアドレスを使用して、構造のレベル２にあるこの指定されたテーブルをアドレス指定する。このプロセスは、メモリ・テーブルの階層中の経路を指示する、対応するメモリ・アドレスを使用して、描画された記号で選択される点とともに進行する。

【００４０】

メモリ・テーブルは、少なくとも３タイプの出力が可能である。点／アドレスのシーケンスが、ライブラリ中の１つまたは複数の記号と整合する限り、各メモリ・テーブルは、階層中の次のレベルにある特定のテーブルをポイントする「テーブル・アドレス」を出力する。そして、描画された記号上の対応する点をテストするために、入力シーケンス中の次

のアドレスを使用してその指定されたテーブルをアドレス指定する。その点が、可能なライブラリ記号のどれとも整合性がない場合、メモリ・テーブルの出力は、描画された記号を認識できなかったことを知らせる。しかし、メモリ・アドレスの完全なシーケンスがライブラリ記号の１つと整合する場合、最後のメモリ・テーブルからの出力は、その描画記号のコードになる。その後、表示装置によって記号を再描画することができる。

【００４１】

描画された記号上で選択された各点について、それに対応するメモリ・アドレスは、その点の相対的なＸ値、その点の相対的なＹ値、および単一のビデオ・ビットからなる。図２Ａの記号「Ｎ」４００の場合、所与の点の相対的なＸ値は、その点のＸ値から点４０２のＸ値を減算することによって得る。所与の点の相対的Ｙ値は、その点のＹ値から、点４０２のＹ値を減算することによって得る。メモリ・アドレスの単一のビデオ・ビットは、所与の点をサンプリングする時間の前にビデオが「オン」（すなわち非ブランク状態）であるか否かを示す。描画された記号の完全なビデオ情報（すなわち、描画された記号の輝度および／または色）は、その記号を正確に再描画するのに必要であるが、記号を認識する際に必要なのはビデオの「オン－オフ」状態だけである。

【００４２】

図２Ａの記号「Ｎ」４００の記号認識プロセスを、図３に図解する。図３のアドレス５０２は、図２Ａの点４０４から得る。アドレス５０２は、点４０４の相対的なＸ位置ΔＸ１　５０６と、相対的なＹ位置ΔＹ１　５０８の両者、および単一のビデオ・ビットＶ５０４からなる。ビデオ・ビットＶ５０４の状態は、サンプリングされた点４０４の前にビデオがブランクであったことを表す。アドレス５０２は、記号認識プロセスの最初のテーブル・ルックアップ・オペレーションに使用される。このアドレスは、図３のメモリ・テーブル５１０からの出力を決定する。図３のメモリ・テーブルの構造の第１レベルは、この１つのメモリ・テーブル５１０からなる。この構造のレベル２は、テーブル５２０、５２２、５２４、５２６と、恐らくは、図３には明示していない追加のテーブルからなる。レベル３は、テーブル５３０、５３２、５３４、５３６、５３８と、恐らくは、明示していない追加のテーブルからなる。同様に、レベル４も、テーブル５４０、５４２、５４４、５４６、５４８、５４９と、恐らくは追加のテーブルからなる。レベル５は、テーブル５５０、５５２、５５４、５５６、５５８、５５９と、恐らくは追加のテーブルからなる。「Ｎ番目の」レベルと表せるこの構造の最後のレベルは、テーブル５６０、５６２、５６４と、恐らくは追加のテーブルからなる。このメモリ・テーブルの構造は、図３には明示していないが、レベル５と最後のレベルの間に存在するレベルをさらに含むことができる。

【００４３】

図２Ａの記号「Ｎ」４００の記号認識動作では、図２Ａの点４０４、４０６、４０８、４１０、および４１２から連続的に得られる５つのアドレスのシーケンスを用いる。これらのアドレスはそれぞれ、図３のメモリ構造の対応するレベルにおけるテーブル・ルックアップ・オペレーションで使用される。図３には、このシーケンスの最初のアドレス（すなわち点４０４から得たアドレス５０２）だけを明示している。（点４０２に相対的な）点４０４の位置と、点４０４の前のブランクのビデオ状態はどちらも、複数の可能な記号の描画との整合性がある。この可能な記号には、記号Ａ、Ｂ、Ｄ、Ｆ、Ｈ、Ｋ、Ｍ、Ｎ、Ｐ、Ｒ、およびＸがいずれも含まれる。したがって、メモリ・テーブル５１０は、入力されたアドレス５０２に応答して、これらの記号の可能性をすべて見込むように設計されたレベル２のテーブル５２０へのポインタ５７０を出力するように設計する。次いで、点４０６から得たシーケンス中の次のアドレスを、テーブル５２０によるテーブル・ルックアップ・オペレーションに使用する。点４０６の位置は、記号「Ａ」または「Ｘ」の描画との整合性はない。したがって、テーブル５２０の出力５７２はレベル３のテーブル５３６をポイントし、このテーブルは、点４０４および点４０６の両方となお整合性がある記号だけを見込むように設計されている。この時点で整合性のある記号には、記号Ｂ、Ｄ、Ｆ、Ｈ、Ｋ、Ｍ、Ｎ、ＰおよびＲが含まれる。次いで、点４０８から得たアドレスを、メモリ

・テーブル５３６に対するテーブル・ルックアップ・オペレーションに使用する。しかし、点４０８は、記号Ｂ、Ｄ、Ｆ、Ｈ、Ｋ、Ｍ、Ｐ、およびＲのいずれの描画とも整合しない。記号「Ｎ」のみが可能性として残る。したがって、テーブル５３６の出力５７４はレベル４のテーブル５４４をポイントし、このテーブルはこの最後の可能性に照らして点４１０をテストするように設計されている。点４１０は、実際に記号「Ｎ」と整合性がある。したがって、点４１０から得たアドレスに応答して、テーブル５４４の出力５７６はレベル５のテーブル５５２をポイントし、このテーブルは、記号「Ｎ」との整合性について点４１２をテストするように設計されている。これが該当するので、テーブル５５２は記号「Ｎ」のコード５７８を出力して、その記号が認識されたことを示す。点４１２の位置が記号「Ｎ」４００の描画と整合しない場合、テーブル５５２は、描画された記号がライブラリにあるどの記号とも整合しないことを示すコードを出力する。

【００４４】

メモリ・テーブルの内容は、記号ライブラリにある記号をいずれも認識するように設計する。一部の応用例では、ライブラリの各記号についてのメモリ・アドレスのシーケンスは、メモリ・テーブル構造中で単一の固有の経路をたどることができる。しかし、これが常にあてはまるわけではない。次のような例を考えられたい。一部の特定の応用例では、描画された記号で選択されたある点の位置が、例えば記号「Ｓ１」と記号「Ｓ２」など、２つの異なるライブラリ記号と整合する可能性がある。ただし、その点が、おそらくはノイズが原因でわずかに異なる位置にある場合、その点は記号「Ｓ１」とは整合するが、記号「Ｓ２」とは整合しないことが考えられる。この場合は、構造の次のレベルが、記号「Ｓ１」の可能な認識を見込んだ２つのメモリ・テーブルを含んでいることになる。２つのテーブルの１つは、「Ｓ１」と「Ｓ２」の両方を見込み、もう一方のテーブルは「Ｓ１」だけを見込む。そしてこの応用例では、少なくとも２つのパスがメモリ・テーブルの構造中に存在し、その結果記号「Ｓ１」が認識される。

【００４５】

ここに記載するパターン認識の方法は、ニューラル・ネットワークの手法と似た部分がある。例えば、この２つの手法はどちらも、入力のシーケンスをライブラリにあるすべてのパターンと同時に比較する（すなわち、一度に１つのライブラリ・パターンと逐次比較するのに対して）。ただし、ハードウェアおよびソフトウェアどちらの実装においても、本発明は、より高レベルのパフォーマンスを達成することができる。ハードウェア実装の場合、本発明は一般に、ニューラル・ネットワークの手法に比べてコストの低い解決法を提供する。また、本発明では、標準的な市販のメモリ・デバイスを利用することができるので、ニューラル・ネットワークの手法に伴う、将来の部品の旧型化というリスクも回避される。ニューラル・ネットワークの手法と同様に、本発明は、基本的な設計を専用アプリケーションへと適合する学習法を取り入れることもできる。

【００４６】

本発明のメモリ・テーブル中のデータは、分析、シミュレーション、学習法、あるいはこれらの技術の組み合わせによって得ることができる。学習法の場合、データは、ライブラリ記号の実際の描画から集めることができる（例えば、図１のブロック図の点選択機能３１４の出力から）。このデータを、最終製品の実際の現実世界の環境で収集すると、その環境の実際のノイズ状態を反映するものになる。そして、コンピュータ・プログラムを使用した自動化プロセスでそれを分析し、描画された各記号で選択された点の名目値と、実際のノイズ環境におけるこれらの値の標準的な偏向の両方を求めることができる。テーブル・データの生成には、標準的な統計的方法を用いることができる。例えば、記号認識機能における特定エラーの確率についてこのデータを生成することができる。このテーブル・データを得る手法は、ニューラル・ネットワーク特有のあくまでも経験に基づく学習手法を上回る大きな利点を提供する。ニューラル・ネットワークの解決法では、ノイズ・マージンとエラーの確率は経験的にしか知ることができず、またその確度もはるかに低い。

【００４７】

本発明の特定の応用例で必要とされるメモリ量は、容易には求められない。例えば、所与

の構造レベルに必要とされるメモリ・テーブルの数は応用例に依存するが、一般には、ライブラリ中の記号（またはパターン）の数とともに増加する。ここに記載する応用例では、メモリ構造中のレベルの数は、ライブラリにある任意の記号を描画するために選択される点／サンプルの最大数によって決まる。しかし、大半の記号では、選択される点の数はこの最大数よりも少ない。したがって、ライブラリ記号の大半は、構造のレベルをすべて利用することはない。描画された記号で選択する点の数は、あいまいでない記号認識を行うのに十分な数でなければならないが、追加の／冗長な点を選択するとメモリ・サイズが不必要に増大する。このような複雑性はあるものの、本発明の可能なメモリ要件を考察することはやはり有用である。

【００４８】

ライブラリが１２８個の記号を含み、描画された記号で最大３３個の点が選択される設計に必要なメモリを考えてみたい。このメモリ構造はしたがって、３３のレベルを含むことになり、最初のレベルは単一のテーブルからなる。残りの３２レベルについては、「平均的な」テーブル数の妥当な上限は６４、すなわち記号数の半分になると考えられる。多くの記号に類似性があることから、構造の最初の方のレベルが必要とするテーブルは比較的少ない。また、描画される記号の大半では、選択される点の数は最大の選択される点の数よりも少ないので、構造の最後の方のレベルも比較的少ないテーブルを有する。どのレベルでも、テーブルの数は一般に、ライブラリ記号の総数である１２８よりもやや少ない数になる。したがって、３２のレベルそれぞれにある平均６４個のテーブル（すなわち合計２０４８個のテーブル）が、実際のメモリ要件の上限を表すことができる。ΔＸ値およびΔＹ値それぞれを６ビットと想定し、単一のビデオ・ビットを考慮に入れると、１３ビットのアドレスが与えられ、このアドレスはメモリ・テーブル１つにつき８１９２ワードを必要とする。（ΔＸおよびΔＹを６ビットにすると、比較的大きな英数字記号に対応することができる。）

このメモリ・テーブルには、８ビット／１バイトの出力語で十分である可能性がある。この出力中の単一ビットで、残りの７ビットが７ビットの記号コードであることを表すことにより符号化された記号の認識を示すか、あるいは残りの７ビットが７ビットの「テーブルアドレスコード」であることを表すことができる。各レベルを最大で１２７個のテーブルに制限する場合は、次のレベルのテーブルの１つを選択するか、または描画された記号の認識の失敗を表すのには、７ビットのテーブルアドレスコードで十分である。（この全体的なメモリ・アドレス指定方式ではレベル番号も用いる。）この結果、合計１２８Ｍｂｉｔのメモリが必要になる（すなわち、３２レベル・６４のテーブル・８１９２ワード・８ビット＝１２８Ｍｂｉｔ）。現在の技術を用いると、このメモリは単一のＦＬＡＳＨタイプのメモリ・チップを用いて実装することができる。

【００４９】

応用例によっては、大きな全画面のシンボロジを表示することが必要となる。本発明を用いて小さな英数字記号だけを認識し、再描画した場合、大きな記号は再描画せずに直接画像メモリに書き込むことになる。しかし本発明は、全画面のシンボロジを認識するように適合することもできる。これを実用化する手法の１つは、メモリ・アドレスで使用されるΔＸ値およびΔＹ値をより低い解像度に合わせてスケーリングすることである。さもないと、メモリ・テーブルのサイズが著しく大きくなってしまう。応用例によっては、この値を、表示画面のピクセル解像度の１／８または１／１６に縮小する。これと同じスケール係数、またはこれと類似の係数を、描画された記号における点選択の頻度にも適用する。全画面の記号と小さな英数字記号の両方を認識し、再描画するように設計されたシステムでは、記号の認識に２つの独立した回路を使用することができる。この２つの回路は並行して動作することができ、１つの回路を英数字シンボロジの認識に使用し、もう一方の回路を大きなシンボロジの認識に使用する。大きな記号を処理するための代替の手法は、メモリ・アドレスで使用されるΔＸ値およびΔＹ値の計算に使用するアルゴリズムを修正することである。この手法では、描画された記号で最初に選択される点だけを、その記号の開始位置の基準とする。残りの点については、ΔＸとΔＹの値を、そのシーケンス中でそ

の１つ前に選択された点の位置の基準とする。この代替手法は、大きな記号の認識により高い解像度を利用できるという利点を有する。この手法は小さなシンボロジにも使用することができるが、通例大きなシンボロジにはより低い点選択の頻度が用いられるので、小さい記号と大きな記号には独立した記号認識機構を用いるのが有利である。応用例によっては、大きな記号と小さい記号の混合を取り込んだ、特定の表示形式の限定されたセットしか表示しないものもある。このような応用例では、２つの記号認識回路を併せて使用して、描画された記号の混合および／またはシーケンス化によって特定の表示形式を識別することができる。

【００５０】

多くのシステムでは、英数字記号の基本的なセットを複数のサイズで表示することができる。この記号認識機構は、所与の記号の各サイズを、完全に異なる記号として扱うことができる。先の例における合計１２８個の異なる記号は、基本的な記号セットの２つまたは３つの異なるサイズの認識を可能にすることができる。この手法は、複数サイズのシンボロジを扱う方法の中で最も費用効果の高いものであると思われる。ただし、この記号認識機構が、時間分割式に複数のデータ・ストリームを処理するのに十分な速さを有するのであれば、複数の記号サイズを認識する別の手法が可能である。メモリ・テーブルは、メモリ・アドレスの最初のストリームを介して「通常サイズ」の記号を認識するように設計することができる。その後、ΔＸとΔＹの値を代替サイズから「通常」サイズにスケーリングすることにより、各代替記号サイズについて追加のデータ・ストリームを生成することができる。すると、この記号認識機構が、このデータ・ストリームすべてによって時間分割式に使用される。次いで、特定のデータ・ストリームのスケール係数から、認識された記号の実際のサイズを求めることができる。代替の手法は、単一のデータ・ストリームだけから、複数サイズの記号を認識することを可能にする。この手法では、描画された各記号上の特定の点（例えば、最初のビデオ遷移がある点）を固定値にスケーリングし、同じスケール係数をその描画記号で選択されるすべての点に適用する。（この手法を簡潔にするために、スケール係数は、ΔＸまたはΔＹのうち大きい方を固定値にスケーリングすることによって得てもよい。）次いで、描画された点で使用したスケーリング・アルゴリズムと整合する記号固有のスケール係数とともに、各ライブラリ記号をテーブルに記憶する。そして、テーブルのスケール係数と、描画された点に適用されたスケール係数から、認識された記号の実際のサイズを得ることができる。

【００５１】

本発明は、回転された記号の認識にも用いることができる。最適な実時間のパフォーマンスのために、記号認識機構は、入力された点のシーケンスをすべてのライブラリ記号について同時にテストするだけでなく、これらの記号を任意の可能な方向で認識するように設計すべきである。ライブラリ記号の回転させたバージョンをメモリ・テーブルに含めると、メモリ・サイズが著しく増大する。しかし、図３に示す記号認識機構を修正すると、メモリを増加させずに、回転した記号を認識することが可能になる。この修正を加えた手法では、描画された記号の上で選択される点はそれぞれの極座標によって定義される。所与の点のΔＸおよびΔＹの値は、テーブル・ルックアップ・オペレーションによって、それに対応する極座標の値（ｒとθ）に変換することができる。先の例では、ΔＸとΔＹはともに、６ビットの符号付きの値である。これらの値は、ｒについては５ビットの大きさの値に、θについては７ビットの角度値に変換することができる。ΔＸおよびΔＹの値と同様に、これらの極座標値も記号の位置に相対的にすることができる。したがって図２Ａの記号「Ｎ」４００では、この２つの値を点４０２の基準とし、図２Ｂの記号「Ｌ」４２０では、この２つの値を点４２２の基準とする。この結果、記号認識に使用されるメモリ・アドレスは、単一ビットのビデオ値、ｒの値、およびθの値からなる。したがって、ライブラリ記号を反映する、メモリ・テーブルに記憶されたデータは極座標系に一致する。

【００５２】

回転を扱うアルゴリズムでは、「描画されるオフセット角度」の減算によって、メモリ・アドレス・シーケンス中のθ値を、描画された記号上のそれに対応する点のθ値から修正

する必要がある。このオフセット角度により、描画された記号を、記号認識機構に入力される前に効果的に回転させる。描画された記号で最初に選択される点の角度は保存して、描画されるオフセット角度の値に使用する。したがって、その実際の方向に関係なく、描画されたライブラリ記号は常に同じ方向で記号認識機構に提示されることになる（すなわち、「０」に等しい最初に選択された点の角度で）。したがって、メモリ・テーブルは、各ライブラリ記号の１つの方向だけを記憶すればよく、メモリ・テーブル中の所与の記号の方向は、その特定の記号に固有であることができる。（このメモリ・アドレス中の角度ビットは常に「０」なので、構造のレベル１にある単一のメモリ・テーブルは、他のテーブルよりも小さくてよいことに留意されたい。）

ここでは、通常の、すなわち回転されない所与のライブラリ記号の方向は、「０」に等しい回転角度に対応するものと定義する。図２Ａは、位置４０２におけるその通常方向での（すなわち「０」に等しい回転角度による）記号「Ｎ」４００の描画を示している。この記号で最初に選択される点（すなわち点４０４）の角度は、角度θＮで示している。図２Ｃは、位置４４２における方向－θＮでの記号「Ｎ」４４０の描画を示す。これは、メモリ・テーブルに記憶され、記号認識機構への入力時に提示される記号「Ｎ」の方向である。図２Ｂは、回転させない方向で位置４２２に描画した記号「Ｌ」４２０を示す。記号「Ｌ」４２０で最初に選択される点（すなわち点４２４）は角度θＬを有する。図２Ｄは、点４５２に方向－θＬで描画された記号「Ｌ」４５０を示す。これは、メモリ・テーブル中にある記号「Ｌ」の方向であり、記号認識機構の入力に提示される方向である。このように、各ライブラリ記号は、描画された記号で最初に選択される点に角度「０」を与える方向に対応する、その記号に特有の「テーブルオフセット角度」とともにメモリ・テーブルに記憶される。（記号「Ｎ」のテーブルオフセット角度は－θＮに等しく、記号「Ｌ」のテーブルオフセット角度は－θＬに等しい。）記号を認識する際、描画された記号の実際の方向は、描画されるオフセット角度とテーブルオフセット角度の合計によって与えられる。したがって記号はこの方向で再描画することができる。

【００５３】

記号を描画する前のＸとＹの偏向が比較的安定した期間中に、記号の位置を得るために、サンプリングしたＸとＹの値をフィルタリングすることを思い出されたい。これは、複数回の表示の更新にわたって安定した／一貫性のある記号の位置を提供し、それにより動的な表示の異常を防ぐ。回転された記号の動的な異常を回避するには、複数回の更新にわたって安定した正確な回転角度も得る必要がある。上記の回転されたシンボロジを認識する手法は、記号の位置を求めるのに２つの記号点のみに依拠する。これら２点の最初の点、すなわち記号の位置が正確であると分かっている場合でも、第２の点（例えば図２Ａの記号「Ｎ」４００の点４０４）の座標はノイズを受けやすい。したがって、求めた回転角度が、記号認識に適するものであっても、動的な表示の異常を回避するのに十分な安定性を複数回の更新にわたって持たないことがある。ただし、図１の点選択機構３１４へのデジタル化されたＸおよびＹの入力はすべて、ＦＩＦＯパイプライン・バッファ３２８に保存される。これを保存することにより、ある記号がライブラリの記号の１つとして認識されないときに、画像メモリに直接描画することが可能になる。描画された記号のこれらの点の一部を、後処理操作（すなわち記号の認識後の処理）で使用して、描画記号のより正確な方向角度を得ることができる。サンプリング・シーケンス中の記号に固有の点である可能性がある複数の点は、記号の方向角度の複数の計算を提供するのに使用することができる。　記号を回転させずに描画する場合、所与の点の予測される角度を、サンプリングした点の実際の角度から減算する。次いでこの複数の記号の方向のサンプルを平均化して、より正確な方向角度を提供することができる。

【００５４】

表示形式によっては、大きなシンボロジと小さいシンボロジの両方に同じ回転角度を使用するものがある（例えば、一部の実時間のアビオニクスの表示形式がそうである）が、この回転角度の値は、より大きなシンボロジからの方が正確に得ることができる。先に指摘したように、描画されたシンボロジの混合および／またはシーケンスから表示形式を識別

することができる。このような場合、より大きなシンボロジから求めた回転角度の値を、より小さなシンボロジの再描画に適用することができる。

【００５５】

本発明のこの例示的な応用例は、本発明を説明する手段として使用しているのであり、本発明を制限するものでないことは理解されたい。この応用例の範囲内でも、本発明の実装と使用の変形例が可能である。例えば、回転された記号を認識する手法を、複数の記号サイズを認識する手法と併せて、かつ／または大きな全画面のシンボロジを認識する手法と併せて使用することができる。また、認識した記号をそれが最初に描画された際と同じ形で再描画する必要はない。例えば、異なるサイズや異なるスタイルのフォントで描画することができる。また、認識した記号を完全に異なる記号として再描画する、「記号の置き換え」を実装することも可能である。

【００５６】

本発明は、文字がすでに２次元空間に描画されている（例えばラスタ・ビデオ信号から文字が描画されているような）文字認識に見られる旧来の問題のために使用することもできる。このような文字は、単一のビデオ・ビットを用いる２Ｄビットマップを使用して定義することができる。そしてこのビットマップから、記号認識機構へのメモリ・アドレス入力を直接生成することができる（例えばビットマップの最上行をレベル１のアドレスに使用し、第２行をレベル２のアドレスに使用するなど）。比較的大きなビットマップの場合は、必要とされる各メモリ・テーブルのサイズを縮小するために、ビットマップの各行から２つ以上のメモリ・アドレス入力を得ることができる。言うまでもなく、この操作の前に、文字サイズの正規化（すなわちＸおよびＹのスケーリング）と、水平線または垂直線を選択するため、または垂直方向および／または水平方向の対称性を最大にするための回転によって、ビットマップを前処理してもよい。

【００５７】

多くのシステムでは、サンプリングしたデータ・パラメータの連続的な実時間のストリームでパターンを認識する必要がある。データ・サンプルは、パターン認識機構に入力するために、それに対応するメモリ・アドレスのストリームに変換するか、またはそのために直接使用することができる。そしてメモリ・アドレスの連続的なストリームを使用して、ライブラリ・パターンのセットを含む有限シーケンスのセットと、実時間入力のシーケンスとを比較する。メモリ・テーブルの構造に記憶されるデータは、ライブラリ・パターンのセットから得る。（ライブラリ・パターンのセットは、パターン中のサンプル数によって、異なる長さのシーケンスを含むことができる。メモリ・テーブルの構造中のレベル数は、最も長いライブラリ・パターン中のサンプル数に対応する。）このような応用例の多くでは、ライブラリ・パターンの開始部と一致する可能性のある、入力データ・ストリーム中の特定の位置を（事前に）識別することはできない。したがって、この記号認識機構を、リセット操作を介して可能なデータ・ストリーム・パターンの始まりに同期化させることはできない。したがって、入力データ・ストリームからの所与のメモリ・アドレスが、ライブラリ・パターンのシーケンス中の任意の位置（例えば、メモリ・テーブル構造中の最初のレベルに対応する第１の位置を含む）と一致することは可能である。また、メモリ構造中の各レベルは、ライブラリ・パターン中のそれに対応するサンプル位置と関連付けられるので、メモリ構造のあらゆるレベルで、テーブル・ルックアップ・オペレーションによって各メモリ・アドレスの入力をテストする必要がある。本発明のハードウェアの実施形態を用いると、メモリ構造のあらゆるレベルで、各メモリ・アドレス入力について同時にテーブル・ルックアップ・オペレーションを実行することができる。したがって、入力データ・ストリーム中のパターンの開始位置を事前に識別できないことによってパフォーマンスが犠牲となる必要はない。

【００５８】

メモリ構造の各レベルに別個の独立したメモリを有する本発明のハードウェア実装を考えてみたい。この場合は、これらのメモリのすべてに同時にアクセスすることが可能である（例えば制御クロック信号の周期ごとなどに）。この各メモリへのアドレス入力は２つの

部分からなる。１つの部分は、パターン認識回路へのメモリ・アドレス入力のストリームによって提供される（すなわち、現在のメモリ・アドレス入力がすべてのメモリに供給される）。アドレスのこの部分は、選択されたテーブルで特定の項目を選択するのに使用される。各アドレスの第２の部分は、各メモリ内で特定のテーブルを選択するのに使用される。各メモリ／レベルへのテーブル・アドレスは、その１つ前のクロック・サイクルの出力により、構造中で次に高いレベルのメモリから提供される（構造のレベル１が単一のメモリ・テーブルしか持たない可能性があることを除いては）。

【００５９】

次に、ここでサイクル「ｉ」と呼ぶクロック・サイクルで開始するこのパターン認識回路の動作を考えてみたい。このクロック・サイクル中に、各メモリ／レベルにおけるテーブル・ルックアップ・オペレーションのために、実時間の入力パラメータのサンプル「ｉ」に対応するメモリ・アドレス入力が使用される。レベル１のテーブル・ルックアップ・オペレーションを使用して、記憶された各ライブラリ・パターンのシーケンス中の最初のサンプルとの整合性について、入力サンプル「ｉ」をテストする。その入力が、パターン中の最初のサンプルだけに基づいてライブラリ・パターンの１つまたは複数と整合する場合、レベル１のメモリ・テーブルは、この可能性を見込んだレベル２の特定のテーブルを選択するテーブル・アドレスを出力する。次いで次のクロック・サイクルでは、この指定されたレベル２のテーブルにおけるテーブル・ルックアップ・オペレーションに入力アドレス「ｉ＋１」が使用される。（このアドレスは、他のレベルのテーブル・ルックアップ・オペレーションにも使用される。例えば、このアドレスは、レベル１で、各ライブラリ・パターンの最初のサンプルとの可能な一致を求めて、入力サンプル「ｉ＋１」をテストするのに使用される。しかし、ここでは、入力サンプル「ｉ」で始まる可能なパターン一致だけを対象とする。）レベル２のテーブル・ルックアップでは、可能な一致として残っているライブラリ・パターンの各シーケンス中の２番目のサンプルに照らして、入力サンプル「ｉ＋１」をテストする（すなわち、サンプル「ｉ」をシーケンス中の最初のサンプルとする）。このテストで、この２つの入力サンプルのシーケンスに対して、１つまたは複数のライブラリ・パターンが可能な一致として残っていると判定された場合、レベル２のテーブルの出力は、この可能性を見込んだレベル３のテーブルを選択するテーブル・アドレスになる。次のクロック・サイクルで、可能な一致として残っている各ライブラリ・パターンのシーケンス中の３番目のサンプルに照らして、３番目のシーケンス入力サンプルをテストするために、この指定されたレベル３のテーブルにおけるテーブル・ルックアップ・オペレーションに、入力サンプル「ｉ＋２」のメモリ・アドレスが使用される。このプロセスは継続し、クロック・サイクルごとにメモリ・テーブル構造を１ステップずつ下へと進む。所与のクロック・サイクルで、対応するいずれかのメモリ構造レベルにおいて、入力サンプルが残りのライブラリ・パターンと整合しない場合は、それがメモリ・テーブルの出力によって示される。この場合、サンプル「ｉ」で始まる入力サンプルのシーケンスは、ライブラリ中のどのパターンとも一致しないことになる。しかし、プロセスがテーブル構造中を進んでライブラリ・パターンの最後のサンプルまで達し、かつ、対応する入力サンプルが、そのライブラリ・パターンの最後のサンプルと一致することが分かった場合は、メモリ出力は、その入力シーケンス（サンプル「ｉ」で始まる）がそのライブラリ・パターンと一致することを示す。

【００６０】

上記から、一回のクロック・サイクルごとに新しい比較動作が開始され、この比較動作は、新しい入力が、ライブラリ・パターンの１つと一致するシーケンス中の最初のサンプルである可能性に基づくことが理解される。開始すると、この比較動作はそれぞれ、一致が見つかるか、またはそのシーケンスがライブラリ・パターンのどれとも一致しないことが判定されることによって中止（ｒｅｔｉｒｅ）されるまで、その後のクロック・サイクルとともに継続する。特定の応用例では、この基本的な回路の異なる実装が必要とされうることは理解されたい。例えば、可能なパターンの開始部で何らかのパラメータの値を使用する前処理操作によって入力メモリ・アドレスが修正される場合（例えば、スケーリング

や回転によって修正される場合）には、構造の各レベルで異なる前処理の値が必要になる。（これは、所与のクロック・サイクルで、構造の各レベルが入力データ・ストリーム中の異なる開始位置のデータ・パターンをテストすることによる。）したがって、回路中の各レベル／メモリには、入力メモリ・アドレスを前処理するための独立したハードウェアが必要となる。

【００６１】

上記のパターン認識回路、あるいはこの回路の一部は、単一の集積回路デバイスとして実装することもできる。このデバイスは、構造の各レベルのための独立したメモリと、これらのメモリをアドレス指定する手段を含む（すなわち、メモリ構造中のテーブル・アドレスの経路指定を含む）。このようなデバイスは、メモリ・テーブルのサイズ、各レベルのメモリに相対的な利用可能メモリの割り当て、およびレベルの総数に関して、ある程度のメモリ構造の再構成を可能にするように設計することができる。また、複数のこのようなデバイスからシステムを構成するのを可能にするために、拡張に対応するように設計することもできる。

【００６２】

一部の応用例は、メモリ・テーブルにＲＡＭを使用することによって益を得ることができる。これは、新しいライブラリ・パターンのセットを反映した、新しいテーブル・データのダウンロードが可能になるためである。例えば、これにより、ノイズ・マージンを適応型のシステムによって実時間で修正することが可能になる。これは、メモリ・テーブルからの出力に追加的情報を含めることによって助長することができる。例えば、追加の出力ビットを使用して、あるパターンが認識されたのが何らかのノイズ・マージン内であったか、またはその外側であったかを示すことができる。（実装に関して、これは、ライブラリ中のパターン／記号の数を単に２倍にするのに相当すると見なすことができる。）テーブルは、確率の値を出力して、認識される各パターン／記号ごとに信頼度係数を生成することができるように設計してもよい。無論、これらの技術は、非適応型のシステムにとっても価値があるものになりうる（例えば、システムの自己監視や、障害の検出／予測に使用することができる）。一部の適応型システムは、テーブル・データを得るための学習法も利用することができる。また、システムで複数のパターン認識回路を使用する場合、他の回路をオンラインの状態にしたまま、これらの回路の１つに新しいテーブル・データをロードすることができる。

【００６３】

本発明が、一般的なパターン認識の問題に対して非常に幅広い応用性を有することが示され、また本発明は多くの異なる応用例で使用することができる。実時間のパフォーマンスを改善する、かつ／またはコストを低減するために、本発明を、ニューラル・ネットワークに代えて、あるいはＤＳＰ相関技術に代えて使用することができる。入力データの前処理とメモリ・アドレス入力のシーケンスの生成に使用される入力パラメータの数と、方法およびアルゴリズムは一般には応用例に固有である。前処理には、離散高速フーリエ変換（ＤＦＦＴ）などのＤＳＰ操作を含むことができる。実証したように、入力データの適切な前処理（例えば、スケーリング／正規化、および／または座標変換や回転による）を使用することにより、必要なメモリ・テーブルのサイズおよび／または数を最小にすることができる。こうした前処理技術は、３次元空間でのパターン認識や、２次元平面への３次元物体の投影の認識を必要とする応用例でも使用することができる。ある応用例が、相互に独立したパラメータ、あるいは前処理によって独立させることができるパラメータを有する場合には、この独立したパラメータのために独立したパターン認識機構を実装することができる。この機構からの出力は相関させることができる。この手法は、すべてのパラメータ値を単一のメモリ・アドレスに含める手法に比べて、合計のメモリ・サイズを大幅に減らすことができる。

【００６４】

本発明の他の応用例は、記号と文字の認識以外にも、音声認識、脅威の検出および識別、自動化された製造および検査システム、および視覚／画像の処理システム（例えば製造で

使用されるロボットによるピックアンドプレース機など）を含むことができる。当業者には、本発明のこの他多くの応用例と、本発明の多数の他の実施形態形態が明らかになろうことが理解される。頭記の特許請求の範囲は、本発明の真の精神と範囲内に含まれるすべての応用例と実施形態を包含するものと解釈されたい。

【００６５】

ここに示す実施形態および実施例は、本発明とその実際的な応用例を最も明確に説明するために、そして当業者が本発明を作成し、利用することを可能にするために提示する。本発明の範囲は、あらゆる面からの同等形態の完全な認識を提供する、添付の特許請求の範囲によって定義するものとする。

【図面の簡単な説明】

【図１】

本発明による、古いＣＲＴタイプのカリグラフィック表示のための置換システムのブロック図を示す図である。

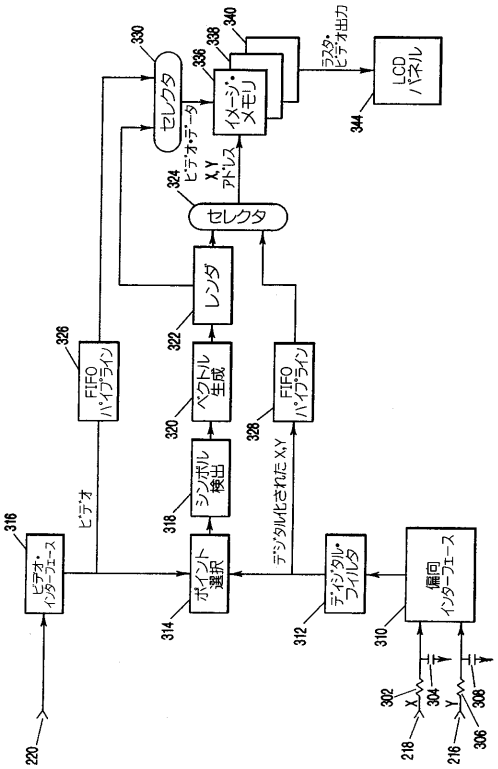
【図２】

本発明による、シンボル「Ｎ」および「Ｌ」が回転なしにレンダリングされる方法、および、これらがメモリ・テーブルにおいて、回転されたシンボロジの認識が可能となるように方向付けられるときの一実施例を示す図である。

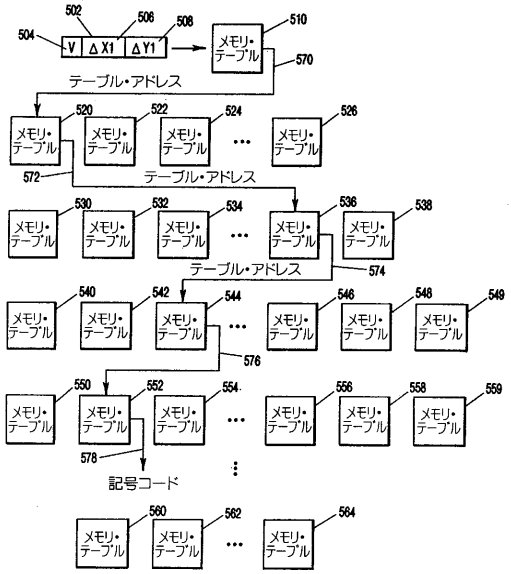
【図３】

本発明によるシンボル認識メカニズムのオペレーションの一実施例を示す図である。

【図１】



【図３】



【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 July 2001 (05.07.2001)

PCT

(10) International Publication Number
WO 01/48729 A2

(51) International Patent Classification⁷: G09G 3/00

(21) International Application Number: PCT/US00/34682

(22) International Filing Date:
21 December 2000 (21.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/474,667 29 December 1999 (29.12.1999) US
Not furnished 20 December 2000 (20.12.2000) US

(71) Applicant *(for all designated States except US)*: **HONEYWELL INTERNATIONAL INC.** [US/US]; 101 Columbia Road, P.O. Box 2245, Morristown, NJ 07960 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **ANDERSON, Bruce** [US/US]; 11600 Academy Road NE, Apartment 1215, Albuquerque, NM 87111 (US).

(74) Agents: **HOIRIIS, David** et al.; Honeywell International Inc., 101 Columbia Road, P.O. Box 2245, Morristown, NJ 07960 (US).

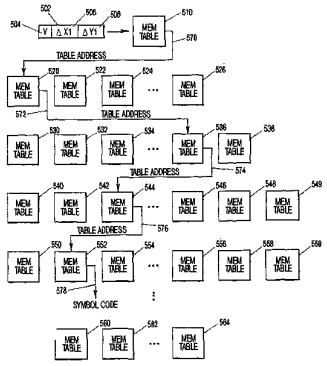
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** SYSTEM, METHOD AND APPARATUS FOR PATTERN RECOGNITION WITH APPLICATION TO SYMBOL RECOGNITION AND REGENERATION FOR A CALIGRAPHIC DISPLAY



(57) **Abstract:** The invention provides a technique for pattern recognition that employs a state machine that incorporates a sequence of table-look-up operations. A sequence of input parameters, derived according to an application-specific algorithm, generates a corresponding sequence of memory addresses for these operations. The memory tables are organized in a hierarchical structure that corresponds to the input sequence. Table data is designed to recognize a specific library of input patterns. An input sequence traces an input-specific path through the memory tables until one of the patterns in the library is recognized (578) or until it is determined that the input sequence is inconsistent with any of the library patterns. For each library pattern, the table data is designed to accommodate the variations in the input values that are specific to the application (e.g., variations due to noise and/or tolerances). Table data can be derived by analysis, simulation, learning or a combination of these methods. The invention can replace neural networks or DSP correlation techniques in real-time applications. It achieves very high performance by comparing the input sequence to all of the patterns in the library simultaneously. The invention can be employed to improve the image quality of a caligraphic display system that uses flat-panel display technology. A method for improving the performance of display systems that employ image memories to refresh a computer generated image is also disclosed.

WO 01/48729 A2

SYSTEM, METHOD AND APPARATUS FOR PATTERN RECOGNITION WITH APPLICATION TO SYMBOL RECOGNITION AND REGENERATION FOR A CALIGRAPHIC DISPLAY

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates generally to the field of pattern recognition. More specifically, the present invention relates to a system, method and apparatus for pattern or symbol recognition usable in real-time applications.

BACKGROUND ART

There are a number of applications for, and approaches to, pattern recognition. For example, neural networks can be used for character recognition. But high-volume off-the-shelf semiconductor devices for neural networks are not available. Therefore, implementations that employ neural networks can be relatively expensive and they carry an inherent risk that the parts employed might become obsolete. Also, because of compromises that are typically made to neural network devices in order to lower their cost, they may not meet the performance requirements for some real-time applications. Similarly, Digital Signal Processors (DSP's) can also be used for pattern recognition (e.g., with correlation operations performed in either the time or the frequency domain). But DSP's can also be expensive and may not achieve the performance level required by some real-time applications. Thus,

it would be useful to have an alternate approach to pattern recognition that can achieve high performance at a relatively low-cost without significant risk of parts obsolescence.

Many display systems employ cathode ray tubes (CRT's), but new CRT's are not always available to repair or replace these display systems. Therefore, a demand exists to replace these displays with newly designed display systems that utilize flat-panel display technology, such as liquid crystal display (LCD) technology. But the electrical interface to existing CRT-based displays is often incompatible with the requirements of LCD-based displays because of the differences between these two technologies. The primary incompatibility of these two technologies is in the way that the individual pixels of each display type are selected for illumination. LCD technology employs a mechanism for discretely selecting individual pixels, whereas CRT technology employs analog voltage signals to position one or more electron beams on the display surface.

Display processing systems for real-time applications, such as for avionics displays, generally incorporate a hardware vector generation (i.e., line drawing) capability. This hardware provides digital outputs for selecting X (horizontal) and Y (vertical) display screen locations, and digital video to specify the selected intensity and/or color. A separate video on/off output may also be provided. For a calligraphic/stroke type of CRT display, the X, Y, and video signals are converted to analog to directly control the position and intensity of the CRT electron beam(s). The symbol generator hardware then

directly refreshes the display by executing its commands at a rate of approximately 60 HZ. For a dynamic display format, the command-list/program should be updated at a rate that is high enough to achieve smooth motion of the symbology. This may require an update rate of 20 or 30 HZ.

For LCD technology—or for a raster type of CRT-based display—the display screen is refreshed from a 2-dimensional frame buffer memory, or "image memory". An image memory can be an exact 2-D mapping of the pixels of the display screen, whereby each memory location holds a code that determines the color/intensity of its corresponding pixel. The digital X and Y outputs of the symbol generator are then used to address the 2-D image memory. Vectors are written/drawn into image memory instead of directly on the display screen. For real-time applications, a symbol generator is used to update the display format in one image memory while another image memory, which contains the previous update, is simultaneously used to refresh the display screen. Some CRT-based displays are designed to alternate between stroke and raster in a hybrid mode in order to overlay stroke symbology onto raster video.

Avionics display systems often employ separate physical units for the display function and the symbol generator/processor function. Stroke and hybrid CRT displays may utilize an analog interface between the symbol generator and the display unit. If such CRT-based displays are replaced with newer LCD-based units, it may be cost prohibitive to also replace the symbol

generator. The new LCD-based display unit should then be designed to function with the existing analog interface. But because LCD technology employs a discrete type of pixel addressing mechanism, this tends to diminish the quality of the displayed symbology.

A conventional approach to the design of a LCD-based display unit with stroke-type analog inputs is to essentially mimic, or simulate, the operation of a CRT-based display. The analog X and Y deflection inputs are converted to digital and used to address a 2-D image memory where the digital/digitized video is written. Thus, the stroke inputs are used to write symbology into an image memory instead of directly onto a CRT display screen. This image memory is then subsequently scanned and read to provide the video for refreshing a LCD screen. While the digitized inputs are used to write into one image memory, a second memory (which stores the prior update) is used to refresh the display. With this design approach, the analog interface contributes errors/anomalies to the displayed image. These result from: 1) induced electromagnetic interference (EMI) on the analog X and Y signals—from external and internal sources; 2) wideband thermal noise generated within the analog circuitry; 3) phase noise in the sampling operation; and 4) quantization noise from the analog to digital (A-to-D) conversion.

All of these noise sources tend to degrade the quality of the displayed symbology. The severity of EMI depends on the individual application. A low-pass pre-filter is preferably employed before the sampling A-to-D

conversion operation to prevent high-frequency EMI from aliasing into the bandwidth of interest, but lower frequency EMI will degrade the quality of the displayed symbology. Thermal noise causes small random deviations—that exhibit a Gaussian distribution around the norm—in the digitized samples of the analog deflection inputs. The effect of thermal noise can be minimized by oversampling the inputs (i.e., sampling at a higher rate than is required by downstream operations) and then employing a low-pass digital filter. Phase noise results from jitter in the sampling time, which can occur within the A-to-D converter circuit and also from jitter in the sampling clock. But in this application, a more significant error can result from an inconsistent phase of the sampling clock relative to the clock in the symbol generator (i.e., on different updates of the display format). The display anomalies that result from quantization noise can be minimized by employing a high resolution A-to-D converter (e.g., a 12-bit converter may provide a significantly higher resolution than the pixel resolution of the display).

Display anomalies can be categorized as either static or dynamic. A noise or distortion that is the same on every display update would be categorized as static. Its location would be fixed relative to either the display screen or to a symbol on the display screen (i.e., where the symbol itself might be in motion). An example of a static anomaly would be the distortions to individual symbols that result from aggressive low-pass filtering of the deflection inputs (e.g., resulting in symbols with severely rounded corners). A dynamic anomaly can be a noise or distortion that moves across the screen

and/or through the symbology. It can also be located at a specific screen or specific symbol location and be exhibited on either a periodic or random basis. Since humans are adept at detecting motion/change, dynamic anomalies can be more objectionable than static ones. Increasing the amount of low-pass filtering can reduce the dynamic anomalies that result from EMI. However, this would increase the level of static distortion for the symbology.

Because this conventional design approach to a LCD-based caligraphic display with analog inputs results in objectionable display anomalies, it would be beneficial to employ some other design approach that does not have this problem.

BRIEF SUMMARY OF THE INVENTION

The following summary of the invention is provided to facilitate an understanding of some of the innovative features unique to the present invention, and is not intended to be a full description. Although the present invention has numerous applications, this summary will focus on the caligraphic display application. A full appreciation of the various aspects of the invention can be gained by taking the entire specification, claims, drawings, and abstract as a whole.

The present invention provides a pattern recognition solution that can be implemented using semiconductor memory devices. Memory chips are very high-volume commodity parts that are available at very low cost. Moreover, the density of memory chips (i.e., the number of bits in a single

integrated circuit) has increased dramatically with advances in semiconductor technology, and will probably continue to do so. Thus, the invention provides a solution to the pattern recognition problem that achieves a very high level of performance at relatively low cost.

The present invention provides an alternate approach to the design of a LCD-based caligraphic display unit that improves the quality of the displayed symbology. In this alternate approach, the digitized samples of the analog deflection inputs, along with the video input(s), are monitored for the purpose of recognizing the individual symbols that are to be displayed. These symbols can then be regenerated within the display unit using completely digital means, thereby avoiding any degradation in the displayed symbology from the noise and the low-pass filtering at the analog interface. This approach requires a mechanism for reliably recognizing the individual display symbols from the analog deflection and video input(s) to the display unit, even when noise is present on these inputs. Since the symbol recognition function operates in real-time with the inputs to the display unit, the present invention is preferably implemented via hardware for this application, rather than via software. The need for real-time performance would also seem to require a solution that compares the display inputs on a simultaneous basis to the expected inputs for any one of the symbols in a symbol library (i.e., rather than a sequential comparison to one library symbol at a time). The subject invention is a low-cost solution to the symbol/pattern recognition problem that can meet the demanding requirements of this real-time application.

One embodiment of the present invention implements the pattern recognition function by means of a state machine that employs a sequence of table-look-up operations that are designed to recognize patterns in a sequence of inputs. The patterns that are to be recognized (i.e., the library of patterns) are used in deriving the data stored in the memory tables that are used by the table-look-up operations. The tables are organized in a hierarchical, or tree-like, structure having a unique set of memory tables for each level of this structure. The input to this pattern recognition mechanism comprises a time-sequence of memory addresses, which may be derived from the time-sampled and digitized values of one or more analog input signals. This would be the case for the application of a LCD-based calligraphic display unit having analog X and Y inputs. Other applications could derive this sequence of memory addresses by different means.

For the LCD-based calligraphic display, the X and Y input values are normally constant for some time period before the rendering of each display symbol (i.e., except for small changes due to noise). This was required by the older CRT-based display units to allow stabilization of their analog deflection amplifiers. This condition can be detected and used to reset the symbol recognition mechanism before a new display symbol is rendered. In addition, the digitized X and Y inputs can be filtered/averaged during this interval. These filtered values, which are saved, provide an accurate display-screen location for the symbol that is to be rendered.

Not all of the digitized X, Y, and video input samples are required to be used in generating the time-sequence of memory addresses for the symbol recognition mechanism. For example, one algorithm for generating these addresses uses all of the input samples in which the video input changes state and a selected number of the input samples in the intervals between these changes in the video input. For each of these samples, the associated memory address can then be derived by a concatenation of 1) a value representative of the video state just prior to the sample; 2) the relative value of the sampled X input; and 3) the relative value of the sampled Y input. Recall that the X and Y display-screen location prior to the rendering of a symbol is saved. A relative X position—for each of the X input samples—can then be derived by subtracting the saved X symbol position from the X value of the input sample. The relative Y values can be derived similarly. Accordingly, the selected input samples are then used to generate a corresponding time-sequence of memory addresses. This sequence of memory addresses is then used as the input to the symbol recognition operation.

The first memory address in the sequence is used to address a single memory table that resides at the first level in the hierarchy of memory tables. If the value of this address—and hence the values of the inputs for the sample that it represents—is consistent with the rendering of one or more of the symbols in the library, then the output from this address of the memory table points to a specific level-2 table that allows for the possible recognition

of the consistent symbol(s). Then the next memory address in the sequence will be used to address this specified table at the second level in the hierarchy. As this process continues, the sequence of input samples—by means of their associated memory addresses—directs the path through the structure of memory tables until either a symbol is recognized or until it is determined that the input sequence is inconsistent with any of the library symbols.

Three types of outputs are possible for the memory tables. When a completed input sequence is unambiguously recognized as correlating to one of the symbols in the library, the code for that symbol is output from the memory. When an incomplete input sequence is—up to that point in the sequence—consistent with one or more of the library symbols, the memory output points to a specific table at the next level in the hierarchy that will test the next sample of the input sequence against this possibility. And when a given input sample, at any point in an input sequence, establishes that the input sequence is not consistent with any of the library symbols, the output of the memory table indicates this condition. When an input sequence is determined to be inconsistent with any library symbol, the digitized input samples can be used to directly render the symbol into the image memory (i.e., effectively bypassing the symbol regeneration mechanism). But to eliminate the possibility of generating dynamic display anomalies, the symbol recognition mechanism needs to reliably and consistently recognize all of the library symbols (i.e., even when noise is present on the inputs). This requires

that the tables be designed to allow for the variations in the values of the input samples that can be encountered under the worst-case noise conditions.

The novel features of the present invention will become apparent to those of skill in the art upon examination of the following detailed description of the invention or can be learned by practice of the present invention. It should be understood, however, that the detailed description of the invention and the specific examples presented, while indicating certain embodiments of the present invention, are provided for illustration purposes only because various changes and modifications within the spirit and scope of the invention will become apparent to those of skill in the art from the detailed description of the invention and claims that follow.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying figures, in which like reference numerals refer to identical or functionally-similar elements throughout the separate views, and which are incorporated in and form part of the specification, further illustrate the present invention, and together with the detailed description of the invention, serve to explain the principles of the present invention.

FIG. 1 shows a block diagram of a replacement system for older CRT-type calligraphic displays in accordance with the present invention;

FIG. 2 shows an example of how the symbols "N" and "L" would be rendered without rotation and as they would be oriented in the memory tables to allow for the recognition of rotated symbology in accordance with the present invention; and

FIG. 3 is a diagram that shows an example of the operation of the symbol recognition mechanism in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a system, method and apparatus for pattern recognition. Although the invention can be used in a number of applications, the following detailed description relates to a LCD-based caligraphic display unit with analog deflection inputs. By monitoring the analog deflection inputs, along with the video input(s), the symbology that is to be displayed can be recognized and then regenerated within the display unit by completely digital means. This eliminates anomalies and distortions in the symbology from noise and low-pass filtering at the analog inputs. Although the invention will be described relative to its use in this specific application, it should be understood that the present invention can be adapted for use in other applications that require pattern or symbol recognition.

For the caligraphic display application, digitized X and Y deflection values, along with video, are used to select a sequence of points on each of the rendered symbols. This sequence of points is then compared against the expected sequences of points for all of the symbols in a library that is stored in the display unit. If the input sequence is recognized as being a rendering of one of these symbols in the library, the stream of digitized input samples can be discarded and the symbol can be rendered from a list of vectors stored within the display unit. If an input sequence is not recognized as being one of the symbols in the internally stored library, the symbol can be directly

rendered into image memory from the digitized input samples. This requires that the digitized input samples (deflection and video) be saved in a FIFO buffer in order to allow time for the symbol recognition operation. Symbols that are recognized as being in the symbol library are completely re-rendered with digital hardware within the display unit. Therefore, the quality of these displayed symbols is not degraded by noise or by the low-pass filtering at the analog interface.

Figure 1 shows an embodiment of the present invention for the caligraphic display application. This display unit incorporates a flat-panel LCD 344, but it is designed to replace a CRT-based display unit that employs analog deflection inputs for selecting the horizontal and vertical display-screen positions. High frequency noise is removed from the horizontal, or X deflection, input signal 218 by means of a low pass filter comprised of a series resistor 302 and a parallel capacitor 304. A similar low pass filter with series resistor 306 and parallel capacitor 308 removes high frequency noise on the vertical, or Y deflection, input signal 216. The filtered inputs are sampled and converted into digital values on a periodic basis at the deflection interface 310. A high sampling rate (i.e., oversampling) is employed and the stream of sampled values is then processed by a low-pass digital filter 312. The video input 220 may be comprised of multiple signals, and may be analog, digital, or a mix of both analog and digital. The video interface 316 detects the "off-to-on" and the "on-to-off" transitions of the video input 220. These transitions are used to synchronize and control the various operations

within the display unit. The video interface 316 also compensates for any disparity in the timing of the video and the deflection inputs. Such a disparity may exist for an interface that was designed for a CRT-based display. The bandwidth for the deflection amplifiers of a CRT-based display is normally much lower than the bandwidth of the video circuitry. Consequently, the transitions of the video at the input to a CRT-based display are often delayed in order to synchronize with the deflection at the display screen. If this is the case, then the video interface 316 adjusts the video timing to effectively remove this compensation.

Instead of writing symbology directly onto the display screen, as with a stroke type of CRT-based display, the display shown in Fig. 1 writes the symbology into an image memory 336. An image memory is organized as a 2-dimensional array that corresponds—often on a one-to-one basis—with the array of pixels on the display screen. The input address to an image memory is thus comprised of a horizontal, or X, value and a vertical, or Y, value. For some implementations, each location in the memory would then store the video information for a single corresponding pixel of the display screen. This video information may include both the pixel intensity and the pixel color. A total of three image memories 336, 338, 340 are employed. While the symbology is being written into one of the image memories 336, the LCD 344 is refreshed by the video data in another image memory 340 that holds the symbology that was written on the previous display refresh cycle. A third image memory 338 can be simultaneously cleared of symbology so that it can

be used to write updated symbology on the next refresh cycle. The functions of the image memories are then switched, on a rotating basis, at the start of each display refresh cycle.

Display systems have conventionally employed only two image memories, whereby, at any given time, one image memory is used to refresh the display while the second is being updated with new symbology. After the update operation is completed, the functions of these two image memories are interchanged. Some systems—but not all—synchronize this interchange to the completion of the display refresh operation. Following the interchange, the update operation first clears the image memory of the previously updated symbology/image and then writes the new symbology into the image memory. Because this clearing operation is performed time sequentially with the writing operation (i.e., instead of simultaneously) this limits system performance. Both the minimum latency time (i.e., the time delay for displaying updated symbology) and the maximum update rate are constrained by the requirement of performing these two operations sequentially.

By employing three image memories, the three image-memory operations (i.e., display refresh, clearing an image memory of previous symbology, and writing new symbology) can all be performed simultaneously. This allows the display system to achieve a lower latency time and a higher update rate. It should be understood that this technique of utilizing three image memories to improve performance is not limited to the application of a LCD-based caligraphic display, It can be employed in any display system that

refreshes a display with computer generated symbology/graphics from an image memory. It is intended that this feature of the present invention encompass all such display system applications.

In the display system of Figure 1, two different mechanisms are employed for writing symbology into image memory. In one of these, the sampled and digitized X and Y input values are passed through a FIFO pipeline 328 and a selector 324 in order to supply address values to the image memory 336. The sampled video input is passed through a FIFO pipeline buffer 326 and a selector 330 in order to supply the video data input to the image memory 336. When the video data input is on (i.e., not in the off/blank state), it is written into the image memory 336 at the selected X/Y location. This approach emulates the operation of a CRT-based display, but the symbology is written into image memory instead of directly onto the display screen. However, the quality of the displayed symbology can be compromised with this approach as a result of noise on the analog deflection inputs 218 and 216.

The second mechanism for writing symbology into image memory eliminates the degradation to display quality that results from noise on the analog inputs. Instead of writing directly into the image memory, the data stream of sampled inputs is monitored to detect the specific symbols that are to be displayed. The detected symbols are then regenerated by digital means within the display unit. Therefore, they are rendered without the anomalies that can result from the analog interface.

The stream of digitized X and Y values, and the corresponding stream of video data, are first input to a point select mechanism 314. It is not necessary that every sample in the data stream (i.e., every point on a symbol) be processed in order to identify the symbol. Therefore, the point select mechanism 314 is used to reduce the amount of data that is input to the symbol detect mechanism 318. The point selection algorithm selects all of the samples/points in the data stream where a transition in the video occurs (e.g., where the video switches from off-to-on or from on-to-off). Additional points are selected when a relatively long interval occurs between the video transitions (e.g., the algorithm might select every fourth sample/point in the data stream during the interval between the video transitions). The number of points that are selected should be high enough to ensure an unambiguous detection of all of the symbols in a symbol library.

When a specific library symbol is detected, the vector generate mechanism 320 is used to regenerate the symbol from information in an internally stored library. It outputs a sequence of vector commands to the render mechanism 322. The render mechanism 322 then renders each of these vectors into image memory 336. It outputs a sequence of X/Y addresses through selector 324 to the address input of image memory 336 and it outputs video data through selector 330 to the data input of the image memory 336. Video data is written into the image memory at each of the addressed locations along the path of the vector. The vector commands for generating a given symbol designate the locations of the vector end-points as

relative to the location of the symbol on the display. The points along the path of each vector are also initially generated as relative locations. Therefore, the rendering mechanism adds the symbol screen position to these relative values in order to generate the absolute X and Y addresses to the image memory.

When a symbol to be displayed is not recognized as being one of the symbols in the library, the data stream of X, Y, and video samples is used to render the symbol directly into image memory 336 through the FIFO pipeline buffers 326 and 328. The FIFO buffers 326 and 328 are required in order to save the stream of data samples during the time interval that is required for either identifying the symbol or determining that the symbol is not in the library.

Although not explicitly shown in the block diagram of Figure 1, displays that are based on LCD technology generally require the implementation of an anti-aliasing function. Displayed images are actually spatially sampled representations of a desired image, and the quality of the displayed image can be degraded by an aliasing effect between the spatial resolution/frequency of the symbology and the pixel resolution of the display (i.e., the spatial sampling frequency). Aliasing is not a problem with a stroke CRT display because of the low-pass spatial filtering that is inherent with the analog-deflection mechanism of pixel addressing. However, LCD-based displays generally incorporate anti-aliasing circuitry in order to provide a high quality image. To this end, the analog-to-digital conversion of the X and Y

input values employs a higher resolution than the pixel resolution of the LCD screen. Thus each of the X and Y values from the digital filter 312 shown in Figure 1 has both integer and fractional bits. It is the integer portions of these values that are used to address the image memory 336, whereas the fractional bits represent subpixel addressing information that is used for anti-aliasing. One approach to anti-aliasing is to write the X and Y subpixel/fractional addressing information into the image memory 336 along with the video data. This information can then be processed when the image memory is scanned/read in order to implement an anti-aliasing algorithm on the video generated for the display refresh. A second approach is to implement the anti-aliasing function on the input-side of the image memory. This approach requires write operations to additional locations in the image memory 336. Locations that are adjacent or near to those traced by the integer X and Y values would be written with video of a lower intensity.

It should be recognized that the embodiment of the present invention shown in Figure 1 is only a vehicle for describing the subject invention and it represents only a single example of this particular application of the invention. Some designs, for example, might include a capability for arcs as well as vectors in the re-rendering of detected symbols. Also, some display units are designed to allow computer-generated symbology to be overlaid on camera video.

The embodiment of the present invention shown in Fig. 1 allows symbols to be rendered from the internally stored library with very high

quality. But it is equally important that the positions of these symbols on the display screen be stable over multiple display updates. An unstable symbol position would generate a dynamic display anomaly. Fortunately, the operation of a stroke display is compatible with the implementation of a special filtering operation that can accurately determine symbol position. When the symbol generator repositions the X and Y deflection to the initial/start position of a new symbol, it pauses before rendering the symbol so that the deflection amplifiers of a CRT-based stroke display unit have enough time to respond and settle to the correct output voltage. This time interval of stable X and Y input values (i.e., stable within the noise level) can be detected, and the input samples can be filtered during this time period to provide X and Y position values that are stable over multiple updates. For some stroke display systems, a string of alphanumeric symbols may be rendered with only a single pause (i.e., a pause preceding only the first symbol in the string). In this case, the display unit can be designed to position the additional symbols relative to the position of the first symbol in the string, thereby providing stable screen locations for all symbols in the string. The detection of this time period of stable X and Y deflection inputs prior to the rendering of a symbol can also be used to reset the symbol recognition mechanism, which operates as a state machine.

In order to eliminate the generation of dynamic anomalies, the symbol recognition mechanism should consistently recognize any symbol in the library under worst-case noise conditions. Additionally, it should not mistake

some other input sequence for one of the symbols in the library. In other words, any input sequence (library symbol or non-library symbol) should not—even under worst-case noise conditions—reach a threshold level for the detection/recognition of one of the symbols in the library. Because the symbol recognition function is required to operate in real-time with the stream of digitized input samples, it may be necessary to implement this function in hardware rather than in software. The need for real-time performance would also seem to require a solution that compares the sequence of input points on a simultaneous basis to the expected sequences for all of the symbols in the library (i.e., rather than a sequential comparison to one library symbol at a time). The subject invention is a method of pattern recognition that is compatible with all of the above requirements.

The pattern/symbol recognition mechanism will be explained by means of an example that shows the process for detecting the alphanumeric symbol "N". The first step in this process is to select samples from the input data stream that correspond to selected points on the symbol being rendered. These are used to generate a sequence of memory addresses that are used as the input to the symbol recognition mechanism. Figure 2 depicts the rendering of two alphanumeric symbols for both the normal orientation of each symbol and also for a rotated orientation of each symbol. Figure 2A shows the rendering of the symbol "N" 400 in its normal upright orientation and Figure 2B shows the rendering of the symbol "L" 420 in the normal upright orientation. Figure 2 also shows the superposition of two coordinate

systems. The rectangular coordinate system is shown by the horizontal axis X and the vertical axis Y. Radial coordinate r and angular coordinate θ show the superimposed polar coordinate system.

The symbol "N" 400 of Figure 2A is rendered by five vectors, and the reference numbers in Figure 2A designate the end points of these vectors. Before the symbol "N" 400 is rendered, the X and Y deflection inputs to the display are stable—except for noise—at point 402 of Figure 2A. As previously noted, this stable condition is detected in order to derive filtered X and Y values for point 402. The symbol recognition mechanism is also reset. The initial/starting point for rendering the symbol—which in this case is point 402—defines the symbol position. It is also used as the center of rotation when a symbol is rendered as rotated.

The symbol "N" 400 is rendered by first rendering a blank vector from point 402 of Figure 2A to point 404. A "blank" vector is drawn with the video in the off state. This vector is shown as dotted in Figure 2A in order to denote that the video is off. Three contiguous vectors, for which the video is switched on, follow this initial blank vector. Vectors are rendered from point 404 to point 406, from point 406 to point 408, and then from point 408 to point 410. Finally, another blank vector is rendered from point 410 to point 412. This latter vector is included for compatibility with the rendering of strings of alphanumeric characters—it positions the deflection to the next symbol position of a string. Figure 2B shows a rendering of the symbol "L" 420. It is positioned at point 422, and is rendered by first rendering a blank vector from

point 422 to point 424. This is followed by the rendering of two vectors with the video on. The vector rendered from point 424 to point 426 is followed by a vector from point 426 to point 428. This is followed by a blank vector from point 428 to 430, which positions the deflection for a subsequent symbol in a string.

Different algorithms can be employed to select the points on a rendered symbol for generating the inputs to the symbol recognition function. An effective technique is to first select all of the samples in the input stream on which a video transition occurs (i.e., a transition from off-to-on or on-to-off). This would result in the selection of points 404 and 410 on the rendered "N" 400 of Figure 2A. If a pause occurs after the symbol is rendered, detection of this stable period of the sampled X and Y deflection can be used to select the ending point of the symbol (i.e., point 412 of Figure 2A). Additional points are then selected between these initially selected points. A simple and effective algorithm is to select additional points on a periodic basis. For example, every fourth sample following a video transition might be selected. But other algorithms are possible. One possibility is to select the vertex points between contiguous vectors. This would result in the selection of points 406 and 408 in Figure 2A. An algorithm could also be devised for using a combination of vertex selection and the periodic selection of points.

The vertices between the vectors of Figure 2A can be detected by monitoring the changes in the sampled X and Y input values over time (i.e., by monitoring the $\Delta X/\Delta t$ and the $\Delta Y/\Delta t$ values). These values are essentially

constant—except for noise and the limited resolution of the A-to-D conversion—during the rendering of a given vector. But these "velocity values" reflect the orientation of the rendered vector. A change in one or both of these values can be used to detect the location of a vertex/corner between two vectors, and hence to select the additional points on the rendered symbol.

Each vertex in the symbol "N" 400 is formed by a pair of vectors with significantly different orientations. This simplifies the detection of these vertices. A vertex for two vectors with similar orientations is more difficult to detect and can result in a less consistent selection of points on multiple renderings of the same symbol. This is especially true when noise is present. Therefore, in a practical implementation of this invention, the periodic approach to the selection of additional points may be preferred. However, in the interest of simplicity, the vertex selection algorithm will be used in this example. With this algorithm, the points 404, 406, 408, 410 and 412 would be selected on the rendered "N" 400 of Figure 2A.

The next step in the symbol recognition process is to generate a sequence of memory addresses. A single memory address is derived from each of the selected points. The symbol recognition process then uses this sequence of address inputs for a sequence of table-look-up operations. The memory tables for these operations are organized in a hierarchy of levels with one or more tables at each level of this structure and with a separate level for each of the memory addresses of an input sequence. The first address of an

input sequence is used to address a memory table at the first level of this structure. If the location of the first selected point on the rendered symbol is consistent with one or more symbols of the library, then the output of the addressed memory table will point to a specific table at level-2 of the structure that allows for this possibility. Then the second address in the input sequence is used to address this specified table at level-2 of the structure. This process continues with the selected points on the rendered symbol—by means of the corresponding memory addresses—directing the path through the hierarchy of memory tables.

At least three types of outputs are possible for the memory tables. As long as the sequence of points/addresses remains consistent with one or more of the symbols in the library, each memory table outputs a "table address" that points to a specific table at the next level in the hierarchy. Then the next address in the input sequence is used to address this specified table in order to test the corresponding point on the rendered symbol. Should this point be inconsistent with any of the possible library symbols, the memory table output indicates this failure to recognize the rendered symbol. But if a completed sequence of memory addresses is consistent with one of the library symbols, the output from the last memory table is the code for the rendered symbol. The symbol can then be re-rendered by the display unit.

For each selected point on a rendered symbol, the corresponding memory address is comprised of the relative X value of the point, the relative Y value of the point, and a single video bit. For the symbol "N" 400 of Figure

2A, the relative X value of a given point is derived by subtracting the X value of point 402 from the X value at the given point. The relative Y value of the given point is derived by subtracting the Y value of point 402 from the Y value at the given point. The single video bit in the memory address denotes whether or not the video is "on" (i.e., in a non-blank state) prior to the time that the given point is sampled. The complete video information for a rendered symbol (i.e., the intensity and/or color of the rendered symbol) is required for correctly re-rendering the symbol, but only the video "on-off" state is required to recognize the symbol.

The symbol recognition process for the symbol "N" 400 of Figure 2A is diagrammed in Figure 3. The address 502 of Figure 3 is derived from point 404 of Figure 2A. Address 502 is comprised of both the relative X location ΔX1 506 and the relative Y location ΔY1 508, of point 404, and of the single video bit V 504. The state of the video bit V 504 reflects that the video is blank prior to the sampled point 404. Address 502 is used for the first table-look-up operation in the symbol recognition process. It determines the output from the memory table 510 of Figure 3. The first level in the structure of memory tables of Figure 3 is comprised of this single memory table, 510. Level-2 of this structure is comprised of the tables 520, 522, 524, 526, and perhaps additional tables that are not explicitly shown in Figure 3. Level-3 is comprised of tables 530, 532, 534, 536, 538, and perhaps additional tables that are not explicitly shown. Similarly, level-4 is comprised of tables 540, 542, 544, 546, 548, 549, and perhaps additional tables. Level-5 is comprised

of tables 550, 552, 554, 556, 558, 559, and perhaps additional tables. The last level in this structure, which can be designated as the "Nth" level, is comprised of the tables 560, 562, 564, and possibly additional tables. This structure of memory tables may include additional levels, not explicitly shown in Figure 3, that reside between level-5 and the last level.

The symbol recognition operation for the symbol "N" 400 of Figure 2A employs a sequence of five addresses that are derived consecutively from the points 404, 406, 408, 410, and 412 of Figure 2A. Each of these addresses is used for a table-look-up operation at the corresponding level in the memory structure of Figure 3. Only the first address in this sequence (i.e., the address 502 derived from point 404) is explicitly shown in Figure 3. Both the location of point 404 (relative to point 402) and the blank video-state preceding point 404, are consistent with the rendering of a number of possible symbols. These include any of the symbols: A, B, D, F, H, K, M, N, P, R, and X. Therefore, in response to the input address 502, memory table 510 is designed to output a pointer 570 to the level-2 table 520, which is designed to allow for the possibility of any of these symbols. The next address in the sequence, which is derived from point 406, is then used for a table-look-up operation with table 520. The location of point 406 is not consistent with the rendering of the symbols "A" or "X". Therefore, the output 572 of table 520 points to the level-3 table 536, which is designed to allow for only the symbols that remain consistent with both of the points 404 and 406. These comprise the symbols: B, D, F, H, K, M, N, P, and R. The address derived from point

408 is then used for a table-look-up operation on memory table 536. But point 408 is not consistent with the rendering of any of the symbols: B, D, F, H, K, M, P, and R. Only the symbol "N" remains as a possibility. Therefore, the output 574 of table 536 points to the level-4 table 544, which is designed to test point 410 against this last possibility. Point 410 is indeed consistent with the symbol "N". Thus, in response to the address derived from point 410, the output 576 of table 544 points to the level-5 table 552, which is designed to test point 412 for consistency with the symbol "N". Since this is the case, table 552 outputs the code 578 for the symbol "N" to indicate that this symbol was recognized. If the location of point 412 is inconsistent with the rendering of the symbol "N" 400, table 552 outputs a code to indicate that the rendered symbol is inconsistent with any of the symbols in the library.

The content of the memory tables is designed to recognize any of the symbols in a symbol library. For some applications, the sequence of memory addresses for each symbol of the library may trace a single unique path through the structure of memory tables. But this will not always be the case. Consider the following example. For some particular application, the location of a selected point on a rendered symbol might be consistent with two different library symbols, say symbol "S1" and symbol "S2". However, if this point were at a slightly different location—perhaps due to noise, it might be consistent with symbol "S1", but not with the symbol "S2". In this case, the next level of the structure would include two memory tables that allow for the possible recognition of the symbol "S1". One of these would allow for both

"S1" and "S2", and the other would allow for only "S1". Then for this application, at least two paths exist through the structure of memory tables that result in the recognition of the symbol "S1".

This described method of pattern recognition has some similarities to the neural network approach. For example, both of these approaches compare the sequence of inputs on a simultaneous basis to all of the patterns of a library (i.e., rather than comparing on a time-sequential basis to one library pattern at a time). However, for both hardware and software implementations, the subject invention can achieve a higher level of performance. In the case of hardware implementations, the subject invention generally provides a lower-cost solution than the neural network approach. And because it can employ standard off-the-shelf memory devices, it also avoids the risk of future parts obsolescence that is inherent with the neural network approach. As with the neural network approach, the subject invention can also employ a learning method for tailoring a basic design to a specific application.

The data in the memory tables of the subject invention can be derived by analysis, by simulation, by a learning approach, or by a combination of these techniques. For the learning approach, data can be collected from the actual rendering of the library symbols (e.g., at the output of the point select function 314 in the block diagram of Figure 1). If this data is collected in the actual real-world environment of the end product, it will reflect the actual noise conditions of this environment. It can then be analyzed—by means of an

automated process using a computer program—to determine both the nominal values for the selected points on each rendered symbol and also the standard deviation for these values in the actual noise environment. Standard statistical methods can then be employed in the generation of the table data. For example, this data can be generated for a specific error probability in the symbol recognition function. This approach to deriving the table data provides a significant advantage over the strictly empirical approach to learning inherent with neural networks. With a neural network solution, noise margins and error probabilities are known only empirically and much less confidently.

The amount of memory required for a specific application of the invention is not easily determined. For example, the number of memory tables required for a given level of the structure is application-dependent but generally increases with the number of symbols (or patterns) in the library. For the described application, the number of levels in the memory structure is determined by the maximum number of points/samples that will be selected for the rendering of any symbol in the library. But for most symbols, the number of points selected will be less than this maximum. Therefore, most library symbols will not utilize all the levels of the structure. The number of points selected on rendered symbols should be large enough for unambiguous symbol recognition, but the selection of additional/redundant points unnecessarily increases memory size. Given this complexity, it is still useful to examine the potential memory requirements of the invention.

Consider the memory required for a design with a library of 128 symbols and a maximum of 33 points selected on the rendered symbols. The memory structure would then comprise 33 levels with the first level comprised of a single table. For the remaining 32 levels, a reasonable upper bound for the "average" number of tables might be 64, which is one-half the number of symbols. Because of the similarity of many symbols, initial levels of the structure require relatively few tables. And because most rendered symbols have less than the maximum number of selected points, levels toward the end of the structure also have relatively few tables. For any level, the number of tables would generally be somewhat less than 128, which is the total number of library symbols. Therefore, an average of 64 tables in each of 32 levels (i.e., a total of 2048 tables) may represent an upper bound to the actual memory requirement. Assuming 6 bits each for the ΔX and ΔY values, and allowing for the single video-bit, gives a 13-bit address, which requires 8192 words per memory table. (With 6-bits for ΔX and ΔY, relatively large alphanumeric symbols can be accommodated.)

An output word of 8-bits/1-byte may prove sufficient for the memory tables. A single bit in this output can specify the remaining 7 bits to be either a 7-bit symbol-code—to indicate recognition of the coded symbol—or a 7-bit "table-address-code". If each level is limited to a maximum of 127 tables, the 7-bit table-address-code is sufficient to either select one of the tables of the next level or to indicate a failure to recognize the rendered symbol. (The level number is also employed in this overall memory-addressing scheme.) A total

of 128 Mbits of memory is then required (i.e., 32 levels • 64 tables • 8192 words • 8 bits = 128 Mbits). With current technology, this memory can be implemented with a single FLASH-type memory chip.

Some applications require that large full-screen symbology be displayed. If the present invention is employed to recognize and re-render only small alpha-numeric symbols, larger symbols would be written directly into image memory without being re-rendered. But the subject invention can also be adapted to recognize full-screen symbology. One approach to make this practical is to scale the ΔX and ΔY values that are used in the memory addresses to a lower resolution—otherwise the size of the memory tables would become prohibitively large. For some applications these values might be scaled to 1/8 or to 1/16 of the pixel resolution of the display screen. The same scale factor—or a similar factor—would also be applied to the frequency of point selection on the rendered symbols. For a system designed to recognize and re-render both full-screen symbols and smaller alphanumeric symbols, two independent circuits can be employed for symbol recognition. These can operate in parallel, with one circuit used to recognize alphanumeric symbology and the other used to recognize large symbology. An alternate approach for handling large symbols is to modify the algorithm used in calculating the ΔX and ΔY values that are used in the memory addresses. In this approach, only the first selected point on the rendered symbol is referenced to the symbol start position. For the remaining points,

the ΔX and ΔY values are referenced to the location of the previously selected point in the sequence. This alternate approach has the advantage of employing a higher resolution to the recognition of large symbols. Although it could also be used for smaller symbology, it may be advantageous to employ independent symbol recognition mechanisms for small and large symbols because a lower frequency of point selection would normally be used for large symbology. Some applications display only a limited set of specific display formats that incorporate a mix of large and small symbols. In these applications, two symbol recognition circuits can be used in conjunction to identify specific display formats by the mix and/or sequencing of rendered symbols.

Many systems can display multiple sizes of a basic set of alphanumeric symbols. The symbol recognition mechanism can treat each size of a given symbol as a completely different symbol. The total of 128 different symbols in the prior example may allow two or three different sizes of a basic set of symbols to be recognized. This approach may be the most cost-effect method of handling multiple-sized symbology. However, if the symbol recognition mechanism is fast enough to process multiple data streams on a time-shared basis, another approach to recognizing multiple symbol sizes is possible. The memory tables can be designed to recognize "normal size" symbols via the primary stream of memory addresses. Additional data streams can be then be generated for each alternate symbol size by scaling the ΔX and ΔY values from the alternate size to the "normal

size". The symbol recognition mechanism would then be used on a time-shared basis by all of these data streams. The actual size of a recognized symbol can then be determined from the scale factor of the particular data stream. An alternate approach allows the recognition of multiple sized symbols from only a single data stream. In this approach, a specific point on each rendered symbol (e.g., the point at the first video transition) is scaled to a fixed value and the same scale factor is then applied to all of the selected points on the rendered symbol. (To simplify this approach, the scale factor might be derived by scaling the larger of either ΔX or ΔY to the fixed value.) Each library symbol is then stored in the tables with a symbol-specific scale factor that is consistent with the scaling algorithm used on the rendered points. The actual size of a recognized symbol can then be derived from the table scale factor and the scale factor that was applied to the rendered points.

The invention can also be employed to recognize rotated symbols. For optimum real-time performance, the symbol recognition mechanism should then be designed to not only test the sequence of input points against all library symbols on a simultaneous basis, but to also recognize these symbols at any possible orientation. If rotated versions of library symbols were included in the memory tables, this would significantly increase memory size. But a modification to the symbol recognition mechanism depicted in Figure 3 allows the recognition of rotated symbols without increasing memory. In this modified approach, the selected points on the rendered symbol are defined by their polar coordinates. The ΔX and ΔY values for a given point can be

converted into the equivalent polar coordinate values (r and θ) by a table-look-up operation. For the prior example, ΔX and ΔY are both 6-bit signed values. These can be converted into a 5-bit magnitude value for r and a 7-bit angular value for θ. As with the ΔX and ΔY values, these polar coordinate values can be relative to the symbol position. Then for the symbol "N" 400 of Figure 2A they are referenced to point 402, and for the symbol "L" 420 of Figure 2B they are referenced to point 422. The memory addresses used for symbol recognition are then comprised of the single-bit video value, an r value, and a θ value. The data stored in the memory tables—reflective of the library symbols—is then in accordance with the polar coordinate system.

The algorithm for handling rotation requires that the θ values in the sequence of memory addresses be modified from the θ values of the corresponding points on the rendered symbol by the subtraction of a "rendered-offset-angle". Because of this offset angle, the rendered symbol is effectively rotated before it is input to the symbol recognition mechanism. The angle of the first selected point on a rendered symbol is saved and used for the value of the rendered-offset-angle. Therefore, regardless of its actual orientation, a rendered library symbol will always be presented to the symbol recognition mechanism at the same orientation (i.e., with the angle of the first selected point equal to "0"). Thus, the memory tables need only store a single orientation of each library symbol—where the orientation of a given symbol in the memory tables can be unique to that particular symbol. (Note

that the single memory table at level 1 of the structure could then be smaller than the other tables since the angle bits in this memory address will always be "0".)

The normal, or non-rotated, orientation for any given symbol of the library is defined herein as corresponding to a rotation angle equal to "0". Figure 2A shows a rendering of the symbol "N" 400 at position 402 for its normal orientation (i.e., at a rotation angle equal to "0"). The angle of the first selected point on this symbol (i.e., point 404) is shown as the angle θN. Figure 2C shows a rendering of the symbol "N" 440 at position 442 with an orientation of -θN. This is the orientation of the symbol "N" that is stored in the memory tables and that is presented at the input to the symbol recognition mechanism. Figure 2B shows the symbol "L" 420 rendered at position 422 in its non-rotated orientation. The first selected point on the symbol "L" 420 (i.e., point 424) has the angle θL. Figure 2D shows the symbol "L" 450 rendered at the point 452 with the orientation -θL. This is the orientation of the symbol "L" in the memory tables and the orientation presented to the input of the symbol recognition mechanism. In this manner, each library symbol is stored in the memory tables with a "table-offset-angle" that is specific to the symbol and that corresponds to an orientation that gives an angle of "0" for the first selected point on the rendered symbol. (The table-offset-angle for the symbol "N" is equal to -θN, and the table-offset-angle for the symbol "L" is equal to -θL.) When a symbol is recognized, the actual orientation of the rendered

symbol is given by the sum of the rendered-offset-angle and the table-offset-angle. The symbol can then be re-rendered at this orientation.

Recall that during the time period of relatively stable X and Y deflection prior to the rendering of a symbol, the sampled X and Y values are filtered in order to derive the symbol position. This provides a stable/consistent symbol position over multiple display updates, thereby preventing a dynamic display anomaly. To avoid a dynamic anomaly for a rotated symbol, it is also necessary to derive an accurate rotation angle that is stable over multiple updates. The above approach for recognizing rotated symbology relies on only two symbol points to derive the symbol orientation. Even if the first of these points—the symbol position—is known to be accurate, the coordinates of the second point (e.g., point 404 on the symbol "N" 400 of Figure 2A) are subject to noise. Therefore, the derived rotation angle—although adequate for symbol recognition—may not be stable enough over multiple updates to avoid a dynamic display anomaly. However, all of the digitized X and Y inputs to the point select mechanism 314 in Figure 1 are saved in the FIFO pipeline buffer 328. They are saved to allow for a direct rendering into image memory when a symbol is not recognized as being one of the symbols of the library. Some of these points on the rendered symbol can be used in a post-processing operation (i.e., after a symbol has been recognized) to derive a more accurate orientation angle for the rendered symbol. Multiple points, which can be symbol-specific points in the sampling sequence, can be used to provide multiple computations of the symbol orientation angle. The

expected angle of a given point—for a non-rotated rendering of the symbol—is subtracted from the actual angle of the sampled point. These multiple samples of the symbol orientation can then be averaged to provide a more accurate orientation angle.

Some display formats employ the same rotation angle for both large and small symbology (e.g., this is true of some real-time avionics display formats), and the value of this rotation angle can be derived more accurately from the larger symbology. As previously noted, it is possible to identify a display-format from the mix and/or sequence of the rendered symbology. In such cases, the value of the rotation angle that is derived from the larger symbology can be applied to the re-rendering of the smaller symbology.

It should be recognized that this example application of the invention is used as a vehicle for describing the invention and is not intended to be limiting. Even within this application, variations in the implementation and use of the invention are possible. For example, the approach for recognizing rotated symbols can be used with the approaches for recognizing multiple symbol sizes and/or with the approaches for recognizing large full-screen symbology. Also, it is not necessary to re-render recognized symbols in the same way that they are originally rendered. For example, they could be rendered in a different size or in a different style font. It is also possible to implement a "symbol substitution" whereby a recognized symbol is re-rendered as a completely different symbol.

The invention can also be used for the classical problem of character recognition, in which characters are already rendered in 2-Dimensional space (e.g., where characters have been rendered from a raster video signal). Such characters can be defined by means of a 2-D bitmap employing a single bit of video. The memory address inputs to the symbol recognition mechanism could then be generated directly from this bitmap (e.g., with the top row of the bitmap used for the address at level-1, the second row used for the level-2 address, and etc.). For relatively large bitmaps, two or more memory address inputs could be derived from each row of the bitmap in order to reduce the required size of each memory table. Of course, the bitmap could be preprocessed prior to this operation by normalizing the character size (i.e., scaling in X and Y) and by rotation to either select for horizontal or vertical lines or to maximize vertical and/or horizontal symmetry.

Many systems require the recognition of patterns in a continuous real-time stream of sampled data parameters. The data samples can be converted into—or used directly for—a corresponding stream of memory addresses for input to the pattern recognition mechanism. The continuous stream of memory addresses is then used to compare the sequence of real-time inputs against a set of finite sequences that comprise a set of library patterns. The data stored in the structure of memory tables is derived from the set of library patterns. (The set of library patterns can comprise sequences of different length—in terms of the number of samples in a pattern. The number of levels in the structure of memory tables corresponds

to the number of samples in the longest library pattern.) In many such applications, it is not possible to identify (in advance) the specific locations within the input data stream that may coincide with the start of a library pattern. Therefore, the symbol recognition mechanism cannot be synchronized to the beginning of a possible data-stream pattern via a reset operation. It is then possible for a given memory address from the input data stream to align with any position in the sequence of a library pattern (e.g., including the first position, which corresponds to the first level in the structure of memory tables). And since each level in the memory structure is associated with a corresponding sample position in the library patterns, this requires that each memory address input be tested—by means of a table-look-up operation—at every level of the memory structure. It is possible, with a hardware embodiment of the invention, to perform table-look-up operations for each memory address input at every level of the memory structure on a simultaneous basis. Thus, no performance penalty need result from the inability to discern, in advance, the beginning locations of patterns in the input data stream.

Consider a hardware implementation of the invention with separate independent memories at each level of the memory structure. It is then possible to access all of these memories simultaneously (e.g., on every cycle of a controlling clock signal). The address input to each of these memories is comprised of two parts. One part is provided by the stream of memory address inputs to the pattern recognition circuitry (i.e., with the current

memory address input supplied to all of the memories). This part of the address is used to select the specific entry within a selected table. The second part of each address is used to select a specific table within each memory. The table address to each memory/level is provided by the output of the previous clock cycle from the memory at the next higher level in the structure (except that level-1 of the structure may have only a single memory table).

Now consider the operation of this pattern recognition circuit beginning with the clock cycle hereby designated as cycle "i". During this clock cycle, the memory address input—which corresponds to sample "i" of the real-time input parameter(s)—is used for table-look-up operations at each memory/level. The table-look-up operation at level 1 is used to test input sample "i" for consistency with the very first sample in the sequence of each one of the stored library patterns. If this input is consistent with one or more of the library patterns—based only on the first sample in the patterns—the memory table at level 1 outputs a table-address that selects a specific level-2 table that allows for such possibility. Then, on the next clock cycle, input address "i+1" is used for a table-look-up operation on this specified table of level 2. (It is used for table-look-up operations at the other levels, as well. For example, it is used at level 1 to test input sample "i+1" for a possible match to the first sample of each of the library patterns. But we are only concerned here with possible pattern matches that begin with input sample "i".) This table-look-up at level 2 tests input sample "i+1" against the second

samples in each of the sequences of the library patterns that remain as possible matches (i.e., given sample "i" as the first sample in the sequence). If this test determines that one or more library patterns remain as possible matches to this sequence of two input samples, the output of the level-2 table will be a table address that selects a level-3 table that allows for this possibility. On the next clock cycle, the memory address for input sample "i+2" is used for a table-look-up operation on this specified level-3 table in order to test the third sequential input sample against the third sample in the sequence of each of the library patterns that remain as possible matches. This process continues, advancing one step down the structure of memory tables with each clock cycle. If at some given clock cycle, and some corresponding level of the memory structure, the input sample is not consistent with the remaining library pattern(s), this is indicated by the output of the memory table. Then the sequence of input samples that begins with sample "i" does not match any of the patterns in the library. But if the process continues down the structure of tables to the last sample of a library pattern, and if the corresponding input sample is found to match this last sample in the library pattern, then the memory output will indicate that the input sequence (beginning with sample "i") matches this library pattern.

From the above, it is seen that a new compare operation is started on every single clock cycle, where this compare operation is based on the possibility of the new input being the first sample in a sequence that matches one of the library patterns. Once started, each of these compare operations

continues on subsequent clock cycles until it is retired because either a match is found or it is determined that the sequence does not match any of the library patterns. It should be understood that specific applications might require different embodiments of this basic circuit. For example, if the input memory addresses are modified by a preprocessing operation that uses a value from some parameter at the start of a possible pattern (e.g., modified by scaling or rotation) then a different preprocessing value is required at each level of the structure. (This results from the fact that, on a given clock cycle, each level of the structure tests for data patterns at different starting positions in the input data stream.) Then each level/memory in the circuit would require independent hardware for preprocessing the input memory address.

The pattern recognition circuit described above, or a part of this circuit, could be implemented as a single integrated circuit device. This device would then include the independent memories for each level of the structure and the addressing means for these memories (i.e., including the routing of the table addresses through the memory structure). Such a device could be designed to allow some reconfiguration of the memory structure in regard to the size of the memory tables and the allocation of the available memory relative to the memory at each level and the total number of levels. It could also be designed to accommodate expansion in order to allow systems to be configured from multiple such devices.

Some applications can benefit from the use of RAM in the memory tables, as this would allow the downloading of new table data, reflective of

new sets of library patterns. For example, this would allow noise margins to be altered in real-time by an adaptive system. This can be facilitated by including additional information in the output from the memory tables. For example, an additional output bit could be used to indicate whether a pattern was recognized within or outside of some noise margin. (In regard to implementation, this could be viewed as equivalent to a simple doubling of the number of patterns/symbols in the library.) The tables could also be designed to output probability values to allow a confidence factor to be generated for each recognized pattern/symbol. Of course, these techniques can also be of value for non-adaptive systems (e.g., they can be used for system self-monitoring and failure detection/prediction). Some adaptive systems could also make use of the learning method for deriving the table data. Also, if multiple pattern-recognition circuits are used in a system, new table-data can be loaded into one of these circuits while another remains on-line.

The invention has been shown to have very broad application to the general problem of pattern recognition, and it can be employed in many different applications. It could be used in place of neural networks or in place of DSP correlation techniques to improve real-time performance and/or to reduce cost. The number of input parameters and the methods and algorithms for any preprocessing of the input data and for generating the sequence of memory-address inputs would generally be application-specific. Preprocessing could include DSP operations, such as the Discrete Fast Fourier Transform (DFFT). As demonstrated, judicious preprocessing of the

input data (e.g., by scaling/normalizing and/or by coordinate transformation and rotation) can be used to minimize the size and/or number of the required memory tables. These preprocessing techniques can also be used in applications that require the recognition of patterns in 3-dimensional space or the recognition of the projections of 3-D objects onto a 2-D plane. If an application has parameters that are mutually independent—or can be made independent by preprocessing—then independent pattern recognition mechanisms can be implemented for the independent parameters. The outputs from these mechanisms can then be correlated. This approach can significantly reduce the total memory size over the approach of including all of the parameter values in a single memory address.

In addition to symbol and character recognition, other applications of the invention include speech recognition, threat detection and identification, automated manufacturing and test systems, and vision/image processing systems (e.g., for robotic pick-and-place machines used in manufacturing). It should be understood that many other applications for the invention and many other embodiments of the invention will be apparent to those skilled in the art. The following claims should be interpreted as covering all applications and embodiments that fall within the true spirit and scope of the invention.

The embodiments and examples set forth herein are presented to best explain the present invention and its practical application and to thereby enable those skilled in the art to make and utilize the invention. It is intended

that the scope of the present invention be defined by the claims appended hereto, giving full cognizance to equivalents in all respects.

CLAIMS

1. A method for recognizing patterns, the method comprising the steps of:

deriving a sequence of memory addresses, wherein the memory addresses comprise values that are reflective of an input that is to be compared to at least one known pattern set having at least one member pattern;

applying the sequence of memory addresses to a corresponding sequence of memory table-look-up operations to produce a result; and

determining, in accordance with the result of the memory table-look-up operations, whether the input is recognized as being a probable match to any member pattern of the at least one known pattern set.

2. The method of Claim 1 further comprising the step of loading a plurality of memory tables with data that reflects expected sequences of memory address values when the input is a probable match to any member pattern of the at least one known pattern set, and wherein at least a portion of the plurality of memory tables are employed in the sequence of memory table-look-up operations.

3. The method of Claim 2, wherein the step of applying the sequence of memory addresses comprises applying the sequence of memory addresses to a structure of memory tables comprised of the plurality of memory tables and having multiple levels with at least one memory table at each level and wherein a corresponding sequence of memory tables used for the sequence of memory table-look-up operations progresses through at least a portion of the multiple levels of the structure of memory tables.

4. The method of Claim 3 further comprising the step of ascertaining, in accordance with the result of the memory table-look-up operations, at least one value that reflects the probability that the input is a match to at least one member of the at least one known pattern set, and wherein the step of loading a plurality of memory tables with data includes loading of data for providing a result reflecting the probability that the input is a match to at least one member of the at least one known pattern set.

5. The method of Claim 3, wherein the at least one known pattern set having at least one member pattern comprises a single pattern.

6. The method of Claim 3, wherein the at least one known pattern set having at least one member pattern comprises a multiplicity of patterns and wherein the step of applying the sequence of memory addresses comprises selecting the specific memory tables that comprise the sequence of memory tables at least partly from the values of the memory addresses by means of the result of the table-look-up operations.

7. The method of Claim 6 further comprising the step of identifying, in accordance with the result of the table-look-up operations, at least one specific member of the at least one known pattern set that is a probable match to the input for the case that the input is recognized as being a probable match to any member pattern of the at least one known pattern set.

8. The method of Claim 7 further comprising the step of ascertaining, in accordance with the result of the memory table-look-up operations, at least one value that reflects the probability that the input is a match to at least one member of the at least one known pattern set, and wherein the step of loading a plurality of memory tables with data includes the loading of data for providing a result reflecting the probability that the input is a match to at least one member of the at least one known pattern set.

9. A pattern recognition apparatus comprising:

a transformation system for generating a sequence of memory addresses, wherein said memory addresses comprise values that are reflective of an input that is to be compared to at least one known pattern set comprising at least one member pattern;

a plurality of memory tables residing within at least one physical memory device;

addressing circuitry for applying said sequence of memory addresses to address inputs of a corresponding sequence of memory tables from said plurality of memory tables to perform a sequence of memory table-look-up operations for producing a result; and

output circuitry for determining, in accordance with said result from said sequence of memory table-look-up operations, whether said input is recognized as being a probable match to any member pattern of said at least one known pattern set.

10. The apparatus of Claim 9, wherein at least one set of data values is stored in said plurality of memory tables and wherein said at least one set of data values reflects expected sequences of memory address values for said input being a probable match to any member pattern of said at least one known pattern set.

11. The apparatus of Claim 10, wherein said plurality of memory tables comprises a structure of memory tables having multiple levels with at least one of said plurality of memory tables at each of said multiple levels and wherein said sequence of memory tables progresses through at least a portion of said multiple levels of said structure of memory tables.

12. The apparatus of Claim 11 further comprising storing circuitry for storing data into said plurality of memory tables, wherein said data comprises said at least one set of data values.

13. The apparatus of Claim 11, wherein at least one of said at least one set of data values is derived from said sequences of memory address values that are generated by said transformation system from inputs that match at least one known pattern set.

14. The apparatus of Claim 11, wherein said result from said sequence of memory table-look-up operations further comprises at least one output value that quantifies a probability that said input is a match to at least one member of said at least one known pattern set.

15. The apparatus of Claim 11, further comprising:

look-up circuitry for performing a memory table-look-up operation at each of said multiple levels of said structure of memory tables simultaneously;

at least one controlling clock signal for advancing said sequence of memory addresses; and

sequencing circuitry for starting a new sequence of table-look-up operations in response to said at least one controlling clock signal, wherein each position in said sequence of memory addresses is used as a possible starting position for comparing said sequence of memory addresses to said at least one known pattern set.

16. The apparatus of Claim 11, wherein said pattern recognition apparatus comprises at least one integrated circuit.

17. The apparatus of Claim 11, wherein said at least one known pattern set having at least one member pattern comprises a single pattern.

18. The apparatus of Claim 11, wherein said at least one known pattern set comprising at least one member pattern comprises a multiplicity of patterns and wherein said result from said sequence of memory table-look-up operations, which reflects said values of said memory addresses, is used in determining specific memory tables that comprise said sequence of memory tables.

19. The apparatus of Claim 18 wherein, if said input is recognized, at least one specific member of said at least one known pattern set that is a probable match to said input is identified by said result of said memory table-look-up operations.

20. The apparatus of claim 19 wherein said transformation system for generating a sequence of memory addresses further comprises a translation system.

21. The apparatus of claim 19 wherein said transformation system for generating a sequence of memory addresses further comprises a scaling system.

22. The apparatus of claim 19 wherein said transformation system for generating a sequence of memory addresses further comprises a vector rotation system.

23. The apparatus of claim 19 wherein said transformation system for generating a sequence of memory addresses further comprises at least one discrete fast fourier transform.

24. The apparatus of Claim 19 further including storing circuitry for storing data into said plurality of memory tables, wherein said data includes said at least one set of data values.

25. The apparatus of Claim 19, wherein at least one of said at least one set of data values is derived from said sequences of memory address values that are generated by said transformation system from inputs that match at least one known pattern set.

26. The apparatus of Claim 19, wherein said result from said sequence of memory table-look-up operations further comprises at least one output value that quantifies a probability that said input is a match to at least one member of said at least one known pattern set.

27. The apparatus of Claim 19, further comprising:

look-up circuitry for performing a memory table-look-up operation at each of said multiple levels of said structure of memory tables simultaneously;

at least one controlling clock signal for advancing said sequence of memory addresses; and

sequencing circuitry for starting a new sequence of table-look-up operations in response to said at least one controlling clock signal, wherein each position in said sequence of memory addresses is used as a starting position for comparing said sequence of memory addresses to said at least one known pattern set.

28. The apparatus of Claim 19, wherein said pattern recognition apparatus comprises at least one integrated circuit.

29. A display system apparatus comprising:

a display device comprising at least a two-dimensional array of display pixels;

at least one two-dimensional image memory that stores pixel values that reflect a specific display image;

circuitry that refreshes said display device from said pixel values stored in said at least one two-dimensional image memory;

a signal input set comprised of at least one input signal wherein the variation over time of input values for said signal input set reflects symbols for display on said display device;

a pattern recognition apparatus that recognizes the symbols of a first symbol set from said signal input set and further comprising an output that identifies recognized symbols of said first symbol set;

a rendering circuit that independently renders any of the symbols of a second symbol set into said at least one two-dimensional image memory; and

interfacing circuitry that, in response to said output from said pattern recognition apparatus, commands the rendering of symbols from said second symbol set by said rendering circuitry.

30. The apparatus of Claim 29, wherein said signal input set comprises an analog X deflection input, an analog Y deflection input, and at least one video input signal.

31. The apparatus of Claim 30 further comprising:

first circuitry that converts time-sampled values of said analog X deflection input into a sequence of X digital values;

second circuitry that converts time-sampled values of said analog Y deflection input into a sequence of Y digital values; and

third circuitry that transforms said at least one video input signal and said sequence of X digital values and said sequence of Y digital values into a sequence of address values and provides said sequence of address values as an input to said pattern recognition apparatus.

32. The apparatus of Claim 29, wherein said signal input set comprises raster video.

33. The apparatus of Claim 30, wherein said first symbol set and said second symbol set are the same.

34. The apparatus of Claim 30, further comprising direct storage means that, when the variation over time of the input values of said signal input set reflects a symbol that is not recognized as being a symbol of said first symbol set, writes pixel values that are reflective of said at least one video input signal into at least one two-dimensional image memory at locations determined by said analog X deflection input and said analog Y deflection input.

35. The apparatus of claim 30 wherein said pattern recognition apparatus further comprises:

a plurality of memory tables residing within at least one physical memory device; and

addressing circuitry that applies said sequence of address values to address inputs of a corresponding sequence of memory tables from said plurality of memory tables to perform a sequence of memory table-look-up operations for producing a result that comprises said output that identifies recognized symbols of said first symbol set.

36. The apparatus of Claim 35, wherein a set of data values is stored in said plurality of memory tables and wherein said set of data values reflects the expected sequences of address values for the case of said input values for said signal input set being a probable match to any symbol of said first symbol set.

37. The apparatus of Claim 36, wherein said plurality of memory tables is organized into a structure of multiple levels having at least one of said plurality of memory tables at each of said multiple levels and wherein said sequence of memory tables progresses through said multiple levels of said structure.

38. A display system apparatus comprising:

a display device comprising at least a two-dimensional array of display pixels;

at least three image memories each comprising at least a two-dimensional array for storing pixel data that reflects a specific display image;

refresh circuitry for refreshing said display device from the pixel data stored in one of said at least three image memories;

reset circuitry for clearing the pixel data stored in one of said at least three image memories;

update circuitry for writing updated pixel data into one of said at least three image memories wherein said updated pixel data is reflective of an updated display image;

control circuitry for periodically interchanging functions of said at least three image memories wherein an image memory that was most recently updated is used to refresh said display device and wherein an image memory that was previously used to refresh said display device is cleared and wherein updated pixel data is written into an image memory that is cleared.

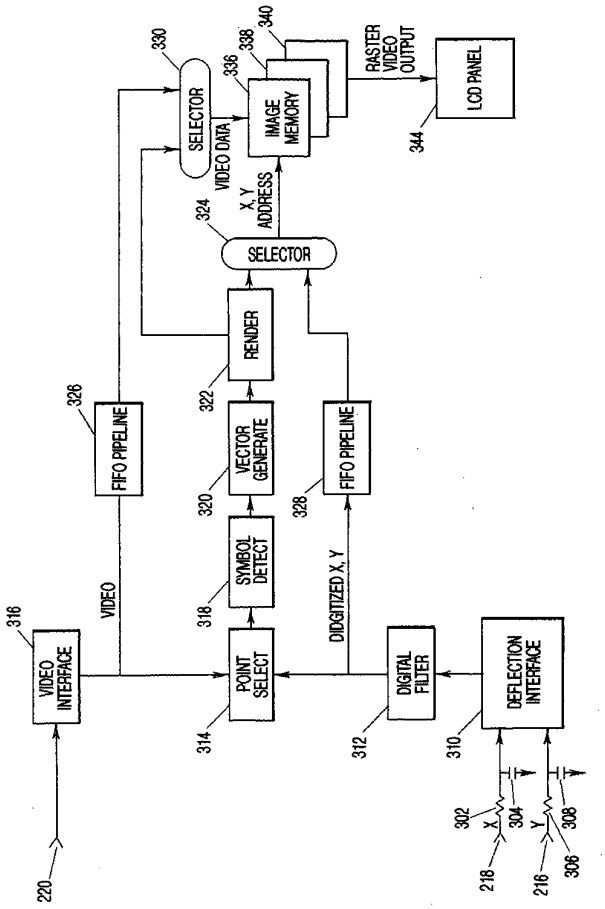

330
SELECTOR
338
336
340
IMAGE MEMORY
RASTER VIDEO OUTPUT
LCD PANEL
344
VIDEO DATA
X, Y ADDRESS
324
SELECTOR
RENDER
322
326
FIFO PIPELINE
VECTOR GENERATE
320
328
FIFO PIPELINE
SYMBOL DETECT
318
VIDEO
DIGITIZED X, Y
316
VIDEO INTERFACE
POINT SELECT
314
DIGITAL FILTER
312
DEFLECTION INTERFACE
310
302
X
304
308
Y
306
218
216
220

FIG-1

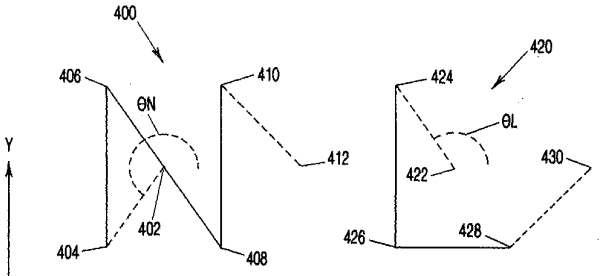


FIG-2A

FIG-2B

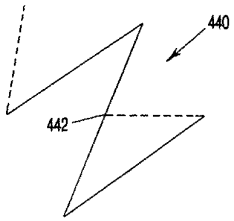


FIG-2C

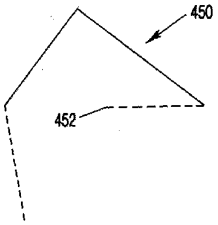


FIG-2D

FIG-2

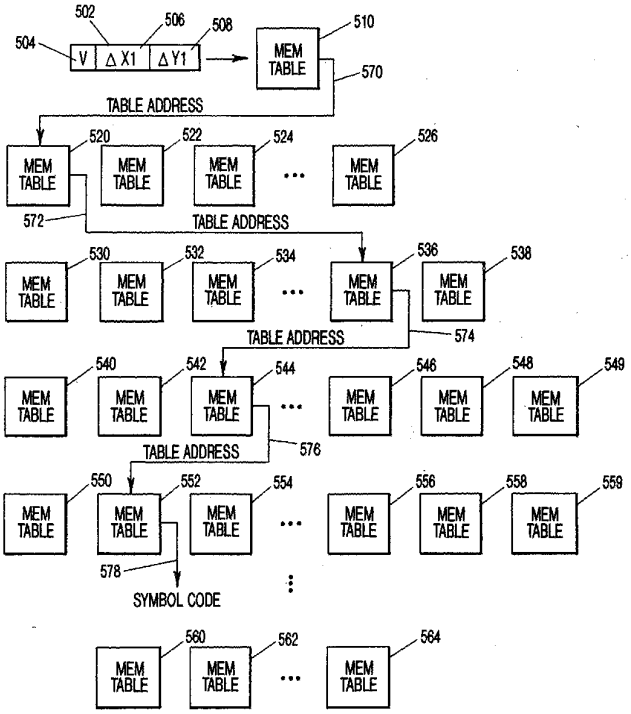

502
504
506
508
V
Δ X1
Δ Y1
510
MEM TABLE
570
TABLE ADDRESS
520
522
524
526
572
530
532
534
536
538
574
540
542
544
546
548
549
576
550
552
554
556
558
559
578
SYMBOL CODE
560
562
564

FIG-3

【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 July 2001 (05.07.2001)

PCT

(10) International Publication Number
WO 01/048729 A3

(51) International Patent Classification⁷: G06K 9/68, G09G 1/16, 1/07

(21) International Application Number: PCT/US00/34682

(22) International Filing Date:
21 December 2000 (21.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/474,667 29 December 1999 (29.12.1999) US
Not furnished 20 December 2000 (20.12.2000) US

(71) Applicant *(for all designated States except US)*: **HONEYWELL INTERNATIONAL INC.** [US/US]; 101 Columbia Road, P.O. Box 2245, Morristown, NJ 07960 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **ANDERSON, Bruce** [US/US]; 11600 Academy Road NE, Apartment 1215, Albuquerque, NM 87111 (US).

(74) Agents: **HOIRIIS, David** et al.; Honeywell International Inc., 101 Columbia Road, P.O. Box 2245, Morristown, NJ 07960 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

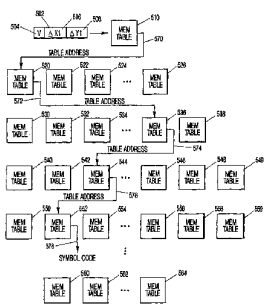
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
12 June 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM, METHOD AND APPARATUS FOR PATTERN RECOGNITION WITH APPLICATION TO SYMBOL RECOGNITION AND REGENERATION FOR A CALIGRAPHIC DISPLAY



(57) Abstract: The invention provides a technique for pattern recognition that employs a state machine that incorporates a sequence of table-look-up operations. A sequence of input parameters, derived according to an application-specific algorithm, generates a corresponding sequence of memory addresses for these operations. The memory tables are organized in a hierarchical structure that corresponds to the input sequence. Table data is designed to recognize a specific library of input patterns. An input sequence traces an input-specific path through the memory tables until one of the patterns in the library is recognized (578) or until it is determined that the input sequence is inconsistent with any of the library patterns. For each library pattern, the table data is designed to accommodate the variations in the input values that are specific to the application (e.g., variations due to noise and/or tolerances). Table data can be derived by analysis, simulation, learning or a combination of these methods. The invention can replace neural networks or DSP correlation techniques in real-time applications. It achieves very high performance by comparing the input sequence to all of the patterns in the library simultaneously. The invention can be employed to improve the image quality of a caligraphic display system that uses flat-panel display technology. A method for improving the performance of display systems that employ image memories to refresh a computer generated image is also disclosed.

WO 01/048729 A3

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/34682

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06K9/68 G09G1/16 G09G1/07

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 4 453 267 A (PAGANINI BRUNO J ET AL) 5 June 1984 (1984-06-05) column 7, line 5 - line 45; figure 4 ---	1-21, 24-28
X	US 4 542 526 A (SATOH KOJI ET AL) 17 September 1985 (1985-09-17) column 5, line 25 -column 7, line 10; figures 1-5 ---	1-21, 24-28
X	EP 0 632 401 A (IBM) 4 January 1995 (1995-01-04) page 4, line 16 -page 5, line 52; figures 1-23 ---	1-21, 24-28
A	US 5 914 711 A (FARWELL RANDALL S ET AL) 22 June 1999 (1999-06-22) -----	

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 November 2002	21 02 2003
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Sonius, M

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 00/34682

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-28

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International Application No. PCT/US 00/34682

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-28

Finite state machine pattern recognition

2. Claims: 29-37

display system including rendering circuitry operating on the output of pattern recognition apparatus

3. Claim : 38

display system including at least three memories for storing pixel data

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/34682

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
US 4453267	A	05-06-1984	US	4308522	A	29-12-1981
			CA	1141473	A1	15-02-1983
			DE	3071181	D1	21-11-1985
			EP	0027451	A1	29-04-1981
			JP	1028426	B	02-06-1989
			JP	56500107	T	05-02-1981
			WO	8002080	A1	02-10-1980
US 4542526	A	17-09-1985	JP	59075375	A	28-04-1984
			CA	1209264	A1	05-08-1986
			DE	3377751	D1	22-09-1988
			EP	0107196	A2	02-05-1984
EP 0632401	A	04-01-1995	US	5390260	A	14-02-1995
			EP	0632401	A2	04-01-1995
			JP	2739036	B2	08-04-1998
			JP	7021325	A	24-01-1995
US 5914711	A	22-06-1999	NONE			

Form PCT/ISA/210 (patent family annex) (July 1992)

フロントページの続き

(51)Int.Cl.[7]	FI		テーマコード（参考）
	Ｇ０９Ｇ 5/00	５１０Ｈ	
	Ｇ０９Ｇ 5/36	５１０Ｚ	

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GW,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CR,CU,CZ,DE,DK,DM,DZ,EE,ES,FI,GB,GD,GE,GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,PL,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TR,TT,TZ,UA,UG,US,UZ,VN,YU,ZA,ZW

(74)代理人 100096013
弁理士 富田 博行
(74)代理人 100107696
弁理士 西山 文俊
(72)発明者 アンダーソン，ブルース
アメリカ合衆国ニューメキシコ州８７１１１，アルバカーキ，アカデミー・ロード・ノースイースト １１６００，アパートメント １２１５

Ｆターム(参考) 5B064 AB04 AB13 BA05 DA05 DA13 DD03 DD10 DD11 DD16
5C006 AF01 AF06 AF13 AF45 AF51 AF53 AF61 BF02 BF14 BF24
5C080 AA10 BB05 DD21 EE17 GG12 JJ02 JJ05
5C082 AA01 AA12 AA21 BA02 BA12 BB15 BB51 BD02 CA85 DA53
DA71 MM05 MM10

【要約の続き】
。本発明は、リアルタイム・アプリケーションにおけるニューラル・ネットワークまたはＤＳＰ相関技術に取って代わることができる。これは非常に高いパフォーマンスを、入力シーケンスをライブラリにおけるすべてのパターンと同時に比較することによって達成する。本発明を使用して、フラットパネル表示技術を使用するカリグラフィック表示システムのイメージ品質を向上させることができる。イメージ・メモリを使用して、コンピュータにより生成されたイメージをリフレッシュする表示システムのパフォーマンスを向上させるための方法も開示される。