

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4101290号
(P4101290)

(45) 発行日 平成20年6月18日 (2008. 6. 18)

(24) 登録日 平成20年3月28日 (2008. 3. 28)

(51) Int. Cl. F 1
G 0 6 K 9/32 (2006. 01) G 0 6 K 9/32

請求項の数 12 (全 28 頁)

(21) 出願番号	特願平8-531182	(73) 特許権者	リバス・テクノロジー、インコーポレイテッド
(86) (22) 出願日	平成8年4月10日 (1996. 4. 10)		アメリカ合衆国カリフォルニア州94086
(65) 公表番号	特表平10-501913		サニーヴェイル、コバルト・ウェイ・349、ケア／オブ・マインドワークス・コーポレーション
(43) 公表日	平成10年2月17日 (1998. 2. 17)		
(86) 国際出願番号	PCT/US1996/005011	(74) 代理人	弁理士 古谷 聡
(87) 国際公開番号	W01996/032691		
(87) 国際公開日	平成8年10月17日 (1996. 10. 17)	(74) 代理人	弁理士 古谷 馨
審査請求日	平成15年4月7日 (2003. 4. 7)		
(31) 優先権主張番号	08/419, 135	(74) 代理人	弁理士 溝部 孝彦
(32) 優先日	平成7年4月10日 (1995. 4. 10)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 書式処理時の自動ページ登録及び自動領域検出のためのシステム及び方法

(57) 【特許請求の範囲】

【請求項 1】

第1のイメージ内の少なくとも1つのユーザ定義領域に関連した第2のイメージ内のユーザ定義領域を自動的に識別し、前記第2のイメージ内の傾斜と移動を自動的に補正する方法であって、該方法は、

前記第1のイメージ内で連結されたピクセルの第1の複数のブロックを識別するステップと、

前記第2のイメージ内で連結されたピクセルの第2の複数のブロックを識別するステップを含み、各ブロックは、線が除去されたイメージデータをスムージングすることによって形成され、さらに、

前記第1の複数のブロックの1つのブロックからその他のブロックへと向かう少なくとも1組の第1のベクトルを定義するステップであって、前記第1の複数のブロックの任意の2つのブロック間の第1のベクトルは、前記第1のイメージが回転や移動をしても不変である、ステップと、

前記第2の複数のブロックの1つのブロックからその他のブロックへと向かう少なくとも1組の第2のベクトルを定義するステップであって、前記第2の複数のブロックの任意の2つのブロック間の第2のベクトルは、前記第2のイメージが回転や移動をしても不変である、ステップと、

前記少なくとも1組の第1のベクトルを、前記少なくとも1組の第2のベクトルと比較し、前記第1のベクトルと前記第2のベクトルとで一致するベクトルがあるか否かを判定す

10

20

るステップと、
前記第1のベクトルと前記第2のベクトルとで一致するベクトルが存在することに応答して、該一致する2つのベクトルのうち一方のベクトルが定義される前記第1のイメージ内のブロック対と、他方のベクトルが定義される前記第2のイメージ内のブロック対との位置のオフセットを判定して、前記第2のイメージ上のユーザ定義領域の位置を求めるステップを含む、方法。

【請求項2】

少なくとも1つの前記ユーザ定義領域を前記第1のイメージ内で定義するステップを更に含むことを特徴とする、請求項1に記載の方法。

10

【請求項3】

前記第1及び第2のイメージを閾値操作して2色イメージを得るステップを更に含むことを特徴とする、請求項1に記載の方法。

【請求項4】

前記第1及び第2のイメージをサブサンプリングして各イメージの解像度を低減させるステップを更に含むことを特徴とする、請求項1に記載の方法。

【請求項5】

第1のイメージ内で連結されたピクセルの第1の複数のブロックを識別する前記ステップが、

前記第1のイメージ内の傾斜角、及び方向を判定するステップと、
前記傾斜角、及び方向に従って前記第1のイメージの傾斜を除去するステップと、
前記第1のイメージから水平線を除去するステップと、
前記第1のイメージから垂直線を除去するステップと、
連結されたピクセルの前記第1の複数のブロックを生成するために、前記第1のイメージをスムーズにするステップ
を含むことを特徴とする、請求項1に記載の方法。

20

【請求項6】

前記第1の複数のブロックのそれぞれを明確に識別するステップと、
各ブロックについて、データの分類を判定するステップ
を更に含むことを特徴とする、請求項5に記載の方法。

30

【請求項7】

少なくとも1組の第1のベクトルを定義する前記ステップが、
選択されたブロックのそれぞれの中心の間で、他の選択されたそれぞれのブロック内の中心に対するベクトルを判定するステップ
を更に含むことを特徴とする、請求項1に記載の方法。

【請求項8】

前記少なくとも1つのユーザ定義領域のそれぞれの中心の間で、選択されたそれぞれのブロック内の中心に対するベクトルを判定するステップを更に含むことを特徴とする、請求項7に記載の方法。

40

【請求項9】

各ベクトルが距離と方向を含むことを特徴とする、請求項7に記載の方法。

【請求項10】

第1のイメージ内の傾斜角と方向を判定する前記ステップが、
前記第1のイメージ内のピクセル・データをスムーズにし、複数のブロックにするステップと、
連結されたピクセルの組を形成するブロックを識別するステップであって、各ブロックが高さと幅を有することからなるステップと、
所定の範囲内の高さと幅を有するブロックを有効ブロックとして適格にするステップと、
各有効ブロック毎に、ブロック傾斜角と方向を判定するステップと、
前記有効ブロックの大多数の方向を判定するステップと、

50

前記有効ブロックのブロック傾斜角を平均するステップを更に含むことを特徴とする、請求項5に記載の方法。

【請求項11】

第1のイメージから水平線を除去する前記ステップが、分裂した水平線セグメントを、連続した水平線セグメントに変換するために選択されたランレングス・パラメータを使用して、前記第1のイメージにランレングス・スムージングを行うステップと、
最大の高さより小さい高さの水平線セグメントを識別するステップと、
前記水平線セグメント内の設定されたピクセルを未設定ピクセルに変換することによって、識別された水平線セグメントを除去するステップ
を更に含むことを特徴とする、請求項5に記載の方法。

10

【請求項12】

第1のイメージから垂直線を除去する前記ステップが、一時水平線を前記第1のイメージの上部と下部に加えることによって前記第1のイメージを標準化するステップと、
前記第1のイメージを90°回転させるステップと、
分裂した垂直線セグメントを、連続した垂直線セグメントに変換するために選択されたランレングス・パラメータを使用して、前記第1のイメージにランレングス・スムージングを行うステップと、
最大の高さより小さい高さの垂直線セグメントを識別するステップと、
前記垂直線セグメント内の設定されたピクセルを未設定ピクセルに変換することによって、識別された垂直線セグメントを除去するステップと、
前記第1のイメージを90°回転して戻すステップと、
前記一時水平線を除去することによって前記第1のイメージを非標準化するステップ
を更に含むことを特徴とする、請求項5に記載の方法。

20

【発明の詳細な説明】

発明の背景

発明の技術分野

本発明は光学スキャナ、及び光学文字認識を用いた書式処理の分野に関し、より詳しくは、ページの傾斜を自動的に検出して補正し、翻訳するシステム及び方法に関する。

30

発明の背景

光学文字認識(OCR)は、印刷された、又は手書きの文書及び書式からテキストを抽出するために広く用いられている。通常、文書は光学スキャナ上で走査され、ビットマップ形式のイメージ・データが生成される。OCRソフトウェア・アプリケーションは、そのイメージ・データを処理して、そこから全てのテキストを抽出する。次に、ユーザは結果のテキスト・ファイルを見直して、必要な情報だけを得るように他のソフトウェアを用いてそのファイルを更に処理するか又は、ワープロ、データベース、スプレッドシート、会計システム等の有用な書式でそのファイルを保存することが多い。

書式処理の際には、事前印刷されたテキスト・ベースの書式から、ユーザによって付加されたテキスト・データだけを抽出することが望ましい。例えば、会社の名前、住所といった既存のテキスト・データ、及び「顧客名」、「顧客住所」、「項目」、「数量」等といった様々なテキスト・ブロックの記述を含む請求書に関しては、記述テキストではなく、こうしたブロックに入力された情報だけを抽出することが望ましい。このことは、より効果的な処理をテキスト・データに加え、テキスト全てに事後処理をすることなく有用な情報を抽出することを可能にする。

40

書式処理の従来システムは、テンプレートを使用しており、ユーザは、所望のテキストが見つかる可能性の高い、入力文書イメージ上の位置を識別するために、ユーザ定義領域をそのテンプレート内に配置する。これらのシステムでは、テンプレート内のユーザ定義領域が、識別されたイメージの境界、即ちビットマップ内に定義されたイメージの上部、下部、及び側部に関連して配置される。

50

こうした従来のシステムの問題点は、後続の文書がテンプレートで処理される場合、ユーザ定義領域が、文書から生成されたイメージ上に正確に登録、又は配置されないということである。このことは、文書が走査されるときに、スキャナのベッドに対して回転するため、又は文書が文書フィーダによって誤って供給されたり、ユーザによって不適当な位置に置かれたりするために発生する。更に、文書はスキャナのベッドの側部から移動される可能性もある。どちらの場合でも、結果のビットマップ・イメージは、元のテンプレートのビットマップ・イメージから、恐らくかなり変化したものとなり、ユーザ定義領域はイメージの正しい部分に位置づけられなくなる。結果的に、システムは、それぞれのユーザ定義領域内で、正しいテキストを把握できないことになる。

従って、大きな傾斜や移動があっても、入力イメージ上のテンプレートからユーザ定義領域を正しく識別することができるシステム及び方法を提供することが望ましい。

10

発明の概要

本発明は、大きな傾斜や移動があっても、文書内のユーザ定義領域を自動的に識別する方法及びシステムによって、前述の制限を克服するものである。本発明の基本的コンセプトは、傾斜のないイメージ内の様々なテキストやイメージ要素間の関係は、傾斜、移動した位置、ページ・イメージ解像度、又は所与の明るさ及びコントラストの設定に関係なく同じということである。特に、これらのテキスト及び／又はイメージ・エリアの相対距離は、上述のように、どんなパラメータによっても変化しない。従って、所与のページの異なるテキスト、及びイメージ・エリアの間の関係が把握され、ページ登録、及びユーザ定義領域検出に使用される。異なるテキストとイメージ・エリアの関係は、動的データ・ネットワークを使用することによって把握される。

20

動的データ・ネットワークは、イメージ内に存在する、ある限定されたシステム・ブロックに接続されたリンクやベクトルのネットワークである。動的データ・ネットワークは、特定の明るさ及びコントラストの設定において、各イメージに対しユニークである。動的データ・ネットワークを含むテンプレートは、好ましくは書式からの、後続の入力文書を処理するために使用される入力イメージを処理することによって形成される。これらの文書が後に入力された時、動的データ・ネットワークが各入力イメージに対して生成され、選択されたテンプレートの動的データ・ネットワークと比較される。一致した場合、システムはイメージ上に正確にユーザ定義領域を配置することができ、それによって、その位置でイメージからテキストやユーザ定義情報を抽出する光学文字認識機能を使用できる。テンプレートは多くの処理動作によって形成される。好ましくは、入力イメージはおよそ50dpiにサブサンプリングされる。傾斜が検出されると、イメージは、その傾斜を除去するために回転される。データ準備段階では、水平、垂直線が除去され、次に残りのイメージ・データがスムーズにされ、主にテキスト・データとピクチャタイプ・データとなったイメージ・データは接続されたピクセルのブロックに入る。これらのブロックは次に、ラベル付けされ、識別番号が提供され、更にテキスト、イメージ又は2重ブロックに分類される。

30

分類されたブロックは更にサイズ及び形状でフィルタされ、動的データ・ネットワークでの使用に適合するような、多くのシステム・ブロックを提供する。イメージ内の位置及びサイズに従って、このシステム・ブロックの1つが選択され、これらから動的データ・ネットワークが生成される。動的データ・ネットワークは、解像度、サイズ、傾斜、及び移動が変化しない方法で、システム・ブロック間の関係を定義する。所与の明るさ及びコントラストの設定において、システム・ブロックは大きく変化しないので、動的データ・ネットワークを定義するベクトルは、傾斜、移動等の変化の影響をあまり受けない。

40

【図面の簡単な説明】

図1は、テンプレートを使用して入力文書内のユーザ定義領域を自動的に登録し、検出するシステムのブロック図である。

図2は、テンプレート定義、及びページ登録の全体的な方法を示す流れ図である。

図3は、テンプレートを定義する処理の流れ図である。

図4は、テンプレート定義を生成するために使用される様々なモジュールに関する定義執

50

行部のデータフロー図である。

図5a及び5bは、入力イメージにおける傾斜を検出するための傾斜検出処理を示す図である。

図6は、傾斜検出処理の流れ図である。

図7a及び7bは、イメージ上のユーザ定義領域を定義するためのユーザ・インタフェースの例を示す図である。

図8は、入力イメージの準備を行うための様々なモジュールと関連するデータ準備モジュールのデータフロー図である。

図9は、データ準備処理の流れ図である。

図10は、ラベル付けモジュールのブロック識別処理の流れ図である。

10

図11は、データ・タイプ分類処理の流れ図である。

図12は、動的データ・ネットワークを生成する処理の流れ図である。

図13a及び13bは、テンプレートで入力イメージを登録する処理の流れ図である。

図14は、50dpiでサブサンプリングされ、傾斜が除去された2色イメージを示す図である。

。

図15は、ランレングス・スムージングに従う、図14のイメージを示す図である。

図16は、図15のイメージに関して、水平線の除去を行った図である。

図17は、水平線の無い図14のイメージを示す図である。

図18は、標準化、及び回転された図17のイメージを示す図である。

図19は、図18のスムーズにされたイメージを示す図である。

20

図20は、図19から垂直線を除去した結果を示す図である。

図21は、図18及び図20のイメージの結合からの、水平線、又は垂直線の無い、傾斜が除去され回転されたイメージを示す図である。

図22は、非標準化（denormalization）に従って正確に回転された図21のイメージを示す図である。

図23は、図22のイメージに関する水平スムージングを示す図である。

図24は、図21のイメージに関する垂直スムージングを示す図である。

図25は、データ・ブロックを形成するための垂直、及び水平スムージングに従うイメージを示す図である。

図26は、図25のイメージ内のデータ・ブロックを洗練するために、そのイメージに関して更にスムージングを行うことを示す図である。

30

図27は、データ・ブロックを図26のイメージ内にラベル付けした結果を示す概略図である。

。

図28aないし28hは、様々な構築段階における動的データ・ネットワークを示す図である。

発明の詳細な説明

システム・アーキテクチャ

ここで図1を参照すると、本発明の改良型自動ページ登録システムを使用するためのシステムが示されている。このシステム100は、走査された文書及びページ登録テンプレートを長期間保存するための2次記憶装置107、コマンド及びデータを受信し、システム100内に出力するための入力装置109と出力装置116、及びプロセッサ111による実行のために様々なコード・モジュールを記憶するためのアドレス指定可能メモリ113を有するコンピュータ101を含む。システム100は、入力文書を走査でき、その入力文書に関してグレイスケール、白黒、又はカラー・ビットマップ・ファイルのいずれかを生成できるスキャナ115を含む。スキャナ115は、少なくとも50dpiの解像度を有することが望ましい。システム100は、走査された文書又は、システム100内にある他の文書を含む文書を印刷するためのプリンタ117を備える。

40

アドレス指定可能メモリ113は、一緒になって本発明の自動ページ登録方法を管理する実行可能アプリケーションを構成する、多くのコード・モジュールを含む。より詳しくは、アドレス指定可能メモリ113は、アプリケーション実行部137、スキャナ・コントロール・モジュール139、定義執行部121、登録執行部135、縮小化モジュール123、傾斜検出モジュ

50

ール125、傾斜除去モジュール127、データ準備モジュール129、ラベル付けモジュール131、テンプレート・モジュール133、及び光学文字認識モジュール145を含んでいる。こうした様々なモジュールの動作は、自動ページ登録を支援する処理の記述に従い、以下に説明される。アドレス指定可能メモリ113は更に、1次イメージ・バッファ141、及び2次イメージ・バッファ143を含み、これらは、モジュールによる変換の様々な段階でイメージを記憶するために使用される。

アプリケーション執行部137は、残りのモジュールを含む、システム100の全体コントロールを提供する。アプリケーション執行部137は、ユーザとの通信のためのユーザ・インタフェース及び、ユーザ・コマンド及びデータを受信するためのユーザ・インタフェースを提供する。適切なユーザ・インタフェースを介し、アプリケーション執行部137は、ユーザが文書をスキャナ115やその他の資源でシステム100に入力し、定義執行部121で文書を処理するためのテンプレートを生成し、更に登録執行部135で、定義されたテンプレートと共に入力文書を処理することを可能とする。

システム動作

I. 全体処理フロー

システム100は、傾斜の除去と翻訳のために入力文書を自動的に登録するための改良された方法を提供し、それによって、光学文字認識を用いた、より正確な文書からのデータの抽出を提供する。この方法は2つの基本処理を含む。1つは文書又は書式の所与のタイプに関するテンプレートを生成する処理である、もう1つは書式からデータを識別し、抽出するために、入力文書にテンプレートを適用する処理である。

図2を参照すると、本発明の方法の全体が流れ図で示されている。システム100に関連して、ユーザは所与の書式に関するテンプレートを定義する(201)。各テンプレートは、抽出されるテキスト・データを含む多くのユーザ定義領域と、システムによって識別される多くのブロックを指定することによって、書式の内部構造を記述する。このテンプレートは、システム定義されたブロック間の関係によって定義される。ユーザは任意の数のテンプレートを生成することができ、各テンプレートは特定の書式に対応している。例えば、ユーザは、請求書、購買注文、業務用応答名詞、調査書、質問票、ファックス、業務用名詞などに対し、異なるテンプレートを定義することができる。これらは、テンプレートが作成されうる多くの異なる可能な書式のうちのほんの一部にすぎない。従って書式は、ここでは、異なるデータを含む複数の例が生成され、使用される文書の任意のタイプを意味すると理解される。テンプレートの生成及び定義は、定義執行部121によって管理される。

ユーザは次に、定義されたテンプレートの組の中から特定のテンプレートを選択する(203)。例えば、ユーザが請求書の山を処理しようとする場合、ユーザはこれらの文書用に適当な請求書のテンプレートを選択することになる。

ユーザは、システム100内への文書入力(205)に進む。これは、スキャナ115及びスキャナ制御モジュール139で行われることが好ましい。代替案として、ユーザは文書を、直接ファックス受信を含む他の資源から入力する事もできる。文書を入力する任意の資源が、各文書ごとにビットマップ・イメージ・ファイルを作成するために使用されうる。代替案として、ビットマップ・イメージに適切に変換できるのであれば、ベクトル・ベースのファイルが使用可能である。

各文書が入力されるとき(205)、選択されたテンプレートが文書のイメージに適用される(207)。テンプレートは文書上で、ユーザが抽出したいテキスト・データを含むユーザ定義ブロックを識別する。この識別は自動的に行われ、テンプレートに定義された元の書式に関して、イメージに傾斜や移動が存在する場合であっても、ユーザ定義ブロックを識別する。従って、例えば、入力文書が誤ってスキャナ115に供給され、その結果イメージが傾斜した場合であっても、システム100は依然として、ユーザ定義領域を識別することができる。更に、書式上に、走査する間に文書を位置決めし、方向付けるために従来より使用されている追加の登録マークを配する必要がない。テンプレートのイメージへの適用は、登録執行部135によって実行される。

10

20

30

40

50

ユーザ定義領域が識別されると、システム100は従来の方法で光学文字認識モジュール145を適用し、識別されたユーザ定義領域からテキスト・データを抽出する(209)。このデータは後で検索、処理を行うために記憶される(211)。ユーザは次の文書を続けて入力することができ(205)、又処理を終了させることもできる。

II. テンプレート定義

ここで図3を参照すると、テンプレートを定義する処理(201)の流れ図が示されており、図4を参照すると、テンプレートの定義中に他のモジュールと協働する定義執行部121のデータフロー図が示されている。定義執行部121は、モジュール123ないし133の動作、要求された各モジュールの発行、必要な入力のモジュールへの送信、及びテンプレートを生成するための、そのモジュールからの出力の受信を管理する。図14ないし図28は、所与の入力文書がテンプレート定義の様々な段階を通過する図を示している。

10

A. イメージ生成

テンプレート定義は、書式上の指定エリア又は指定領域内で選択されたデータを把握しようとする文書のタイプの典型である書式をシステム100に入力する(301)ことによって開始される。例えば、ユーザは請求書のテンプレートを作成しようとし、光学文字認識によって、顧客名、住所、請求書番号、製品の説明、品質と価格、及び合計額といった指定エリア内のテキスト・データを把握しようとするかもしれない。

文書を入力する(301)ために、ユーザはスキャナ・コントロール・モジュール139と関連してスキャナ115を使用することが望ましい。スキャナ・コントロール・モジュール139は、スキャナ115の動作、例えば明るさ、コントラスト、解像度等の設定を制御する。スキャナ115は、イメージ・ファイルを出力し、それはグレイスケール、2色、又はカラー・ビットマップでありうる。スキャナ115がグレイスケール又はカラー・イメージを出力する場合、スキャナ・コントロール・モジュール139は、システム内での使用のために2色イメージを生成するための閾値操作を行うことが望ましい。任意の標準閾値操作技法が、2色イメージを得るために使用される。基本的に、どんな所与のグレイスケール又はカラー・イメージ(1ピクセルにつき4、8又は24ビット)も、所与のグレイスケール又はカラー・ピクセルが所定の閾値 x (ここで、 x は通常 2^k (k は深さ/2)に設定され、「深さ」はイメージ内のピクセル毎のビット数である)を越えたかどうかを判定することによって、1ピクセルあたり1ビットに変換される。越えている場合、ピクセル値は1に割り当てられ、越えていない場合、ピクセル値は0に割り当てられる。1のピクセル値はピクセルが実施に好都合なように設定されていることを意味し、又逆も同様である。更に、いかなる他の資源からのイメージ・ファイルも、システム100に入力され、本発明に従い処理されうる。こうした資源は、ファックス・イメージ、グラフィック・アプリケーションから生成されたグラフィック・イメージ、又は走査された写真やビデオその他から得られたイメージまでも含む。

20

30

B. イメージ縮小化

定義執行部121はスキャナ制御モジュール139から、アプリケーション執行部137を介して、又は他の任意の資源から2色のイメージを受信する。定義執行部121はその2色イメージ・ファイルを縮小化モジュール123に送信する。縮小化モジュール123は、書式処理とテンプレート定義により適合するように、元の解像度からより低い解像度にイメージの解像度を落とすことによって、入力2色イメージを最小化するか又はサブサンプリングする(303)。スキャナ115のような従来のスキャナは、非常に正確な光学文字認識を達成するのに有用な高解像度イメージ(少なくとも600dpi)を生成する。しかし、この解像度のレベルは書式処理のためのテンプレート生成、及び定義には必要でなく、更に、余分にメモリと記憶域を消費し、性能に大きな影響を与える。従って、縮小化モジュール123は元の資源の解像度を検出し、イメージの縮尺係数又はサブサンプリング係数を計算する。

40

好適実施例において、元の解像度から約50dpiに入力イメージの解像度が低下され、適切な縮尺係数がこの結果について計算される。例えば、元の解像度が200dpiのイメージは、縮尺係数が4で、結果のサブサンプリングされたイメージは元のイメージのサイズの25%になる。元の解像度が50dpiの倍数にならない場合、縮尺係数は、結果の解像度が50dpiに

50

最も近くなるように選択され、イメージのサイズが64kb以下になることが好ましい。イメージ・サイズの制限は、全体性能の改善に有用で、既存のx86アーキテクチャと開発支援ツールによって生じる制限を取り除く。縮小化モジュール123の好適実施例は除去技法を使用し、そこで対応するピクセル数が、サブサンプリング処理の間に元のイメージより小さくなる。例えば、2色イメージが300dpiである場合、各ビットが選択される時に次の6ビットが無視される。縮小化モジュール123は、イメージ・バッファ141内にサブサンプリングされたイメージを出力する。このイメージ・バッファ141は、処理の間、処理されている入力イメージの1つの現在の状態を維持するために使用される。

C. 傾斜検出

イメージ・ファイルが縮小化されると(303)、定義執行部121は、入力イメージの傾斜の存在と程度を識別する(305)のために、傾斜検出モジュール125を起動する。傾斜検出モジュール125は、縮小化モジュール123によって出力されたサブサンプリングされたイメージを得るために、イメージ・バッファ141にアクセスする。好適実施例では、傾斜検出モジュール125は -60° と $+60^{\circ}$ の間の傾斜を検出可能である(ここで 0° はスキャナ115上で入力文書が完全に水平、垂直に関して整列されたことをいう)。

図5a及び5bは、傾斜検出分析を図で示したものである。走査されたイメージ500は、ある程度傾斜のあるテキストの線501を含む。このテキストの線501が滑らかにされると、それぞれが連続する一連のピクセルを含む、多くのテキスト・ブロック503になる。また、イメージ内に存在するものは、線、陰等のような非テキストからなるデータを含むイメージ・ブロック(図示せず)となりうる。連結されたブロック503が識別される。連結されたブロックの組503の左端と右端の間の仮想線505が計算され、この線とブロック503の左端から延びる水平線507の間の傾斜角Fが判定される。この処理はイメージ500内の連結されたブロックの組のそれぞれに対して繰り返され、傾斜角Fが、その連結されたブロックの組503毎に関連づけられる。個別の傾斜角Fの大部分は誤差の指定範囲内でほとんど同じであり、個別の傾斜角Fの全てが平均されて、ある方向(時計回り/反時計回り)に向くイメージ500の傾斜角が生成される。

図6を参照すると、傾斜検出モジュール125によって傾斜検出の好適方法の流れ図が示されている。傾斜検出モジュール125は、イメージ・バッファ141内のサブサンプリングされたピクセル・データに関してランレングス・スムージングを実行し(601)、スムーズにされたブロックを生成する。これは、スムージング・モジュール801(図8)をコールすることによって実行される。スムージング・モジュール801は、入力ランレングス・パラメータを使用して、入力イメージ内の未設定(白)ピクセルに対して汎用ランレングス・スムージング操作を用い、未設定ピクセルを設定された(黒)ピクセルに変換する。好適実施例では、この時点でのスムージングはランレングス値5で実行され、この値はイメージ(50dpi)の好ましい解像度、及びテキスト文書内の典型的なフォント・サイズに基づいている。システムで他の解像度が使用された場合、別のランレングス・パラメータが指定される。スムーズにされたイメージは1次イメージ・バッファ141に保持される。

傾斜検出モジュール125は次に、各ビットをワード(2バイト)に拡大し(603)、イメージ内の各ピクセルを表現する。各ビットを16ビット表現に拡大することによって、ユニークな識別値、及びピクセルを他のピクセルに関連づける情報のような、各ピクセルについての追加情報を記憶することが可能となり、それによって個別のブロックを識別する。追加のビット幅がブロック識別番号、又は他の情報で必要とされる場合、追加バイトへの拡張が使用される。各ワードが後にラベル付けモジュール131によって使用され、ブロック識別値が記憶される。例えば、スムージング操作の後、ピクセルが1111111110000000のような部分を含む場合(ここで、「1」はデータが存在すること(黒のピクセル)を表し、「0」はデータが存在しないこと(白のピクセル)を表す)、拡張後に各ビットは2バイト・ワード: 0000000000000001, 0000000000000001, 0000000000000001, 0000000000000001, 0000000000000001, 0000000000000001, ... 及び0000000000000000, 0000000000000000, 0000000000000000等として記憶される。

傾斜検出モジュール125は次に、ラベル付けモジュール131をコールし(605)、イメージ

10

20

30

40

50

内のピクセルをブロックにセグメント化する。このラベル付けモジュール131の操作は、以下で更に説明する。ラベル付けモジュール131は、ピクセル・データからブロックのそれぞれを識別し、更に連結されたブロック、即ち黒の（設定された）ピクセルによって連結されたブロックの各組を識別する。連結されたブロック（1つ又は複数）の各組は、ユニークな識別番号でラベル付けされ、連結されたブロックの全てが同じ識別番号を有する。

イメージ内のブロック全てが、イメージの全体の傾斜の判断に有用なわけではない。特に、全体イメージ・サイズに関して小さすぎる、又は大きすぎるブロックは、正確な全体傾斜に大きな貢献をしない傾向にある。小さなブロックは非常にわずかな情報を提供し、大きなブロックは過大な値を提供する。従って、傾斜検出モジュール125は、傾斜角の判定の前に、所定の範囲の高さ及び幅にないブロックをフィルタして除去する（607）。好適実施例において、傾斜検出モジュール125は、99ピクセルより高い高さとし、149ピクセルより広い幅を有するブロック、または6ピクセルより低い高さとし、25ピクセルより狭い幅を有するブロックをフィルタし除去する。フィルタ・パラメータもまた、サブサンプリングされたイメージの解像度に基づいており、他の解像度に従って適応させることもできる。残りのブロックから、平均の高さと平均の幅が判定され、これらの値を超えた大きさのブロックだけが更に処理される。フィルタで除去されなかったブロックが有効なブロックである。

傾斜検出モジュール125は次に、前述の通り、残りのブロックから、連結されたブロックの組のそれぞれに対し、傾斜角 F と傾斜の方向を判定する（609）。仮想線は、ブロックの左端中央（ y 軸内）から右端中央に計算されるのが好ましい。

好適実施例において、傾斜角は $\tan^{-1}(\Delta Y/\Delta X)$ であり、ここで ΔX は、ブロックの左端中央の X 軸座標と右端中央の X 軸座標との間の差であり、 ΔY は、両端の Y 軸座標の差である。

別の実施例では、仮想線はブロックの左上から右下の角に向かって取られ、従って ΔX はブロックの左上と右下の角の X 軸座標の差であり、 ΔY はその角の Y 軸座標の差である。この実施例では、角は以下のように判定される。ブロック内の各ピクセル列に対し、その列の下部が、一連の設定されたピクセルの後に続く未設定ピクセルとして検出される。その列の下部の座標が記憶される。現在のピクセル列の下部と次のピクセル列の下部の間の差、 ΔC が、各ピクセル列に関し記憶される。 ΔC は「h,l,d」のような上に出る部分を持つ文字（ascender）及び「g,p,y」のような下に出る部分を持つ文字（descender）によって生じる可能性のある傾斜角誤差の可能性を除去するために計算される。平均の ΔC の値は次に、ブロック全体に適用される。平均 $\Delta C \pm 2$ ピクセルの範囲の外側の ΔC 値を持つ（50 dpi イメージを想定）ピクセル列は、傾斜角の判定に用いられる仮想線の両端を判定する際に無視される。これが「 ΔC テスト」である。ブロックの左上の角から横断する間、 ΔC テストを合格した ΔC の値を有する最初のピクセル列の下部が、仮想線の始まりを構成し、この制約を満たす最後のピクセル列の下部が、仮想線の方の端を構成する。この処理は、有効な各ブロックに対して繰り返され、方向（回転の方向）及び傾斜角が、各ピクセルの拡張ワード内に記憶される。

傾斜検出モジュール125は、イメージがかなりの大きさの方向と傾斜を有していることを確認するために、一連のテストを実行する。これらの方向と傾斜は両方とも重要で、システム100によって補正可能である。傾斜検出モジュール125は、有効なブロックの総数が2以下であるかどうか判定する（611）。こうした少ない適格ブロックでは、全体イメージの傾斜を予測できない。従って、こうした場合には、傾斜検出モジュール125はイメージ傾斜角 F を 0° に設定する。

それ以外の場合、傾斜検出モジュール125は次に、最も数の多い連結されたブロックの組によって保持されている方向を判定する（613）。この数が連結されたブロックの組の総数のうち大多数（好ましくは少なくとも75%）であれば、そのブロックの方向が全体イメージの方向として採用される。傾斜検出モジュール125は、有効ブロックの個別の傾斜検出角 F を平均化（615）することによって、イメージの傾斜角 F を計算する。

傾斜検出モジュール125は、イメージ傾斜角 F が $|60^\circ|$ より大きいかどうか判定する (619)。大きい場合、その文書は極端に誤って配置されている可能性があり、正確に走査され、又はその他の方法で入力された可能性が低く、従って識別及び処理が成功する可能性は低い。この場合、傾斜検出モジュール125は、イメージ傾斜角 F に 0° を設定する (617)。これ以外の場合、イメージ傾斜角 F が境界条件を満たせば、傾斜検出モジュール125はこの値に戻す。これは図4で、検出された傾斜角に戻されることとして示される。

D. 傾斜除去

再び図3を参照すると、検出された傾斜角で、定義執行部121はイメージの傾斜除去を行うかどうかを判定する (307)。好適実施例では、定義執行部121は、イメージ傾斜角が 1° 以下であれば、このわずかな傾斜はテンプレートに関する書式の登録に影響しないので、イメージの傾斜除去を行わない。

イメージ傾斜角 F が 1° を越えた場合、定義執行部121は傾斜除去モジュール127を発行し、サブサンプリングされたイメージと元のイメージ両方を、正確な水平、垂直方向に戻すよう傾斜除去を行う (309)。傾斜除去モジュール127は、イメージ・ファイルから元のイメージを読みとり、イメージ・バッファからサブサンプリングされたイメージを読みとって、それらを検出された傾斜方向と反対の方向にイメージ傾斜角 F だけ回転させる。回転は従来の技法を用いて行われる。

図14は2色イメージ・ファイル内に走査され、50dpiでサブサンプリングされ (フル・ページ・サイズに印刷され)、更に傾斜除去される入力文書の例である。

E. ユーザ定義領域の選択

再び図3を参照すると、定義執行部121が傾斜除去されたイメージをアプリケーション執行部137に提供し、アプリケーション執行部137が更に、その傾斜除去されたイメージを所望のユーザ・インタフェース・ディスプレイを介してユーザに提供する。アプリケーション執行部137は更に、従来の描画ツールに類するツールに、ユーザがユーザ定義領域をグラフィックで作成 (311) できる機能を提供する。このことは、矩形又は多角形作成ツールのような適当な描画ツールを選択し、それを、所望のテキスト情報を含む個別のエリア又は領域を選択するために傾斜除去イメージに適用する。図7aは適当なユーザ・インタフェース705の一例であり、傾斜除去された文書700を示している。図7bは、現在多くのユーザ定義領域701を含む同じ文書を示している。描画ツール703のパレットも示されており、ユーザ定義領域701を選択するための様々なグラフィカル・ツールを備えている。ユーザが多くの領域を定義すると、ユーザ定義領域の各境界の座標が記憶され、ユーザ定義領域が矩形の場合は左上の座標と右下の座標を用いることが好ましい。一般的な多角形のユーザ定義領域に関しては、各頂点の座標も記憶される。

F. データ準備

再び図3を参照すると、定義執行部121が次に、データ準備モジュール129を発行し (313)、不必要な、又は重要でないイメージ内の要素をフィルタして除去するために、サブサンプリングされたイメージに対して多くのタスクを実行する。これには、スムージング、垂直線と水平線の除去、及びブロックの形成が含まれる。データ準備モジュール129は、ラベル付けモジュール131によってラベル付けされる準備の整ったスムーズにされたブロックを有するイメージを生成する。図8を参照すると、データ準備モジュール129の基礎となるモジュールのデータフロー図が示されている。データ準備モジュール129は、スムージング・モジュール801、線除去モジュール803、イメージ回転モジュール805、及び結合モジュール807を含む。図9を参照すると、データ準備モジュール129の動作の流れ図が示されている。この図で、処理ブロック間のテキストは処理されたイメージの現在の状態を参照する。連結矢印の上のテキストは、イメージが1次イメージ・バッファ141にあることを示しており、連結矢印の下側のテキストは、イメージが2次イメージ・バッファ143にあることを示している。使用されるテキストの略語に関して、キーが図内に提供されている。更に、テキストは、所与の処理段階でイメージの例を示す図を識別するためにラベル付けされている。

水平線と垂直線の除去

データ準備モジュール129は、傾斜除去されたイメージをスムーズにする（901）ため、スムージング・モジュール801を発行する。スムージングは水平線の除去の正確性を向上させる。スムージング・モジュール801は、入力ランレングス・パラメータを使用して、従来の方法でランレングス・スムージングを実施する。走査の間、文書内の垂直線及び水平線が、多くの分解された線セグメントに分離されうる。これは、明るさ／コントラスト、走査技法、文書の方向等の変化から生じる。こうした分解された線は、線除去モジュール803による線除去段階においては、識別及び除去がより困難である。同様に、破線又は点線で境界線が引かれているエリアを含む書式も又除去が難しい。従って、線除去モジュール803の正確性を増大させるために、こうした分解された線、破線、又は点線に連続した線を形成させることが望ましい。これは、スムージング・モジュール801によって行われる。

10

スムージング・モジュール801は、これもまたサブサンプリングされたイメージの解像度に依存して、発行され、ランレングス・パラメータが渡され、前記の分離された線が設定されたピクセルによって連結される。好適実施例では、50dpiの好適な解像度が与えられた場合、ランレングス・パラメータが最大で5にまで設定されるが、ランレングス・パラメータは他の解像度用にも調整される。これは、どんな5以下の一連の白のピクセルも一連の黒のピクセルに変換する。この送信は、破線、点線、又は同様の線を、連続した線セグメントに連結する。小さいランレングス値は更に、文字「i」及び「j」の中の点、又は文字「t」、「f」、及びその他の十字を消してしまわないよう意図したものである。

図15は、スムージング・モジュール801によるランレングス・スムージングに従う、図14のイメージを示す図である。

20

データ準備モジュール129は次に、線除去モジュール803を発行し、スムーズにされたイメージ内の水平線を除去する（903）。図16は、図15のイメージ上で水平線を除去した（903）結果を示している。線除去モジュール803はイメージ・バッファを走査し、少なくとも15のランレングスで、設定された（黒の）ピクセルを識別する。これは、潜在的な線セグメントを識別する。線除去モジュール803は次に、潜在的な線セグメントの上の第2の線をチェックし、この線が潜在的な線セグメントと同じランレングスに関して、未設定の一連のピクセルを有するかどうか判定する。代替案として、線除去モジュール803は、設定されたピクセルから未設定ピクセルへの遷移、又は未設定ピクセルから設定されたピクセルへの遷移の数を判定し、この値を所与のランレングスに関する所定の最大遷移と比較する。

30

上記条件が満たされると、これは潜在的な線セグメントの上端を表す。線除去モジュール803は同様に、その潜在的な線セグメントの下第2の線をチェックし、潜在的な線セグメントの下部を識別する。このようにして、潜在的な線セグメントは最大3ピクセルの高さを有することができる。これらの条件の両方が真であれば、潜在的な線セグメントは実際の線セグメントである。線除去モジュール803は次に、設定されたピクセル全てを未設定ピクセルに変更することによって、線セグメントを除去する。この処理はイメージの上部から下部にかけて繰り返される。結果のイメージは2次イメージ・バッファ143に記憶される。指定された数の線、及び線の高さはここでも、イメージの解像度に基づいており、その解像度に従って調整される。

40

結合モジュール807はイメージ・バッファ141、143内の2つのイメージを結合する。このモジュール807は、線が除去された後で、データ準備モジュール129によって発行され、垂直線及び水平線の両方が除去されたイメージを生成する（905）。結合は、2つの入力イメージの各ビットの論理ANDによって行われる。結合モジュール807は水平線の除去の後で発行され、傾斜の除去されたイメージを水平線のないスムーズにされたイメージと結合する。この結果、水平線のない、傾斜が除去されたイメージが生成される。図17は、水平線のない、傾斜が除去されたイメージを示しており、図14と図16を結合したものである。

水平線を除去した後、データ準備モジュール129は、水平線のない、傾斜が除去されたイメージを標準化する（907）。標準化は、イメージの上部と下部で一時的な水平線（1ピクセル幅）を描く手順である。この手順は、スムージング操作の間にイメージの上部及び

50

下部に引かれた境界線を明確に示すことによって、イメージ内の垂直線を除去する場合の正確性を改善するために行われる。一時線がなければ、スムージングの間にイメージの一部がページ境界を跨ってスムーズにされる。垂直線が除去された後、一時線も除去される。

標準化が行われると、傾斜の除去されたイメージは90°回転される。この回転は、イメージ回転モジュール805によって行われ(909)、垂直線が線除去モジュール803によって除去されるように行われる。図18は、図17の傾斜が除去されたイメージの標準化及び回転を示している。

データ準備モジュール129は、標準化された、水平線のない、傾斜が除去されたイメージをスムーズにする(911)ために、スムージング・モジュール801を発行する。図19は、図18のイメージを滑らかにした結果を示している。線除去モジュール803は次に、垂直線を除去する(913)。但し、回転されたイメージにおいてそれらは現在水平線となっている。図20は、図19における水平線(回転する前までは垂直線)を除去した(913)結果を示している。

垂直線が除去された後(913)、そのイメージはステップ909によって、2次イメージ・バッファ143内の回転され、標準化され、水平線のない、傾斜が除去されたイメージとは別に、1次イメージ・バッファ141に記憶される。結合モジュール807が次に、ステップ909で生成された、回転された、水平線のない、傾斜が除去されたイメージを、垂直線のない、スムーズにされたイメージを結合するためにコールされ、そこで水平線又は垂直線のない、回転され、傾斜の除去されたイメージを生成する。図21は図20の垂直線の除去されたイメージと、図18の傾斜が除去され、回転され、標準化されたイメージを結合した結果を示しており、この結合によって水平線、即ち垂直線のない、回転され、傾斜の除去されたイメージが生成される。除去されうる垂直線及び水平線の位置の指定は、適当な変更により切り換えることができる。

水平線、及び垂直線が除去されると、イメージ回転モジュール805が、結合されたイメージを元の方向に戻すよう回転させる(917)。データ準備モジュール129は、そのイメージを非標準化し(919)、標準化の間に生成された一時線を除去する。この結果、イメージは垂直線又は水平線のない、傾斜の除去されたイメージとなる。このイメージは1次イメージ・バッファ141内に記憶される。図22は、図21のイメージの回転を戻し、一時線の上部と下部を取り除くためのデータの非標準化を行った結果を示している。

システム・ブロックの形成

データ準備モジュール129の第2の主要な機能は、ラベル付けモジュール131によってシステム・ブロックとして次にラベル付けするよう考えられているブロックの形成である。ブロックは以下のようにして形成される。

ブロックは、線除去処理と同様の一連のステップによって形成される。水平ランレングス・スムージング921は、スムージング・モジュール801によって、連続ピクセルの大きなブロックを生成するランレングス・パラメータを用いて、水平線及び垂直線のない、傾斜の除去されたイメージに対して実施される。好適実施例では、スムージング操作921は、各ピクセル線に沿って80ピクセル以下で出現する白の(未設定)ランレングスが、黒の(設定された)ピクセルに変換、又は充填されるように、およそ80(50dpiで)の値のランレングスで実施される。結果のイメージは1次イメージ・バッファ141内に記憶される。図23は、図22のイメージをスムージングした(921)結果を示している。

スムージングは次に、ステップ915から得られた垂直線のないイメージ・データ上で実行され(923)、2次イメージ・バッファ143に記憶される。図24は、図21のイメージに垂直スムージングを加えた結果を示している。垂直スムージングがなされたイメージは次に、縦長方向に戻るよう回転され、2次イメージ・バッファ143に記憶される。

水平スムージングが施された1次イメージ・バッファ141内のイメージ、及び垂直スムージングが施された2次イメージ・バッファ143内のイメージは、結合モジュール807によって結合される(925)。この処理はイメージ内に1組のデータ・ブロックを生成する。図25は、データ・ブロックを生成する水平及び垂直スムージングの結合結果を示しており、

ここでは図23と図24（後に回転が行われる）の結合である。

いくつかの例では、出力の中に、更にスムーズにされなければならない可能性のある小さな隙間があることがある。結果のデータに3ピクセルのランレングス値の更なるスムージングを行うと（927）、それによって、これらの残りの要素が除去される。結果イメージは1次イメージ・バッファ141に記憶され、ラベル付けモジュール131による処理の準備が整ったブロックを含む。図26は、図25の残りの要素を除去するスムージング（927）の結果を示している。

G. システム・ブロックのラベル付け、及び分類

再び図3を参照すると、データ準備に続いて、データ・ブロックがラベル付けモジュール131によってラベル付けされ、分類される（315）。普通、ラベル付けモジュール131は、データ準備モジュール129によって生成されたブロックを分析し、ユニークなブロック識別番号を有する各ブロックを識別し、各ブロックをテキスト・ブロック、イメージ・ブロック、又は2重ブロックとして分類する。テキスト・ブロックは、元のイメージ上のテキスト・エリアの存在によって生成された可能性のあるブロックである。イメージ・ブロックは、元のイメージのグラフィック、又は絵の存在によって生成された可能性のあるブロックである。2重ブロックは、テキスト・ブロックにもイメージ・ブロックにも識別されなかったブロックである。連結されたブロックの各グループには、ユニークなブロック識別番号が割り当てられる。

ラベル付けモジュール131は、それぞれがブロックに対応しているブロック識別番号のリストを保持し、各ブロック識別番号毎にブロックに関する統計、及び分類データを記憶する。好適実施例では、ブロック識別リストはブロック構造からなる配列であり、各ブロック構造はブロック識別番号に関する属性、及び以下で更に説明する他のデータ値を有する。

ラベル付けモジュール131は、4ピクセル連結トレーシング技法を用いて、選択されたピクセルがその隣接ピクセルに連結されているかどうかを判定することが好ましい。こうしたピクセルの収集は集合的にクラスタを形成し、それによってブロックのタイプの形成に貢献する。形成された各ブロックに関して、それをブロックのタイプの1つに属するユニークなブロックとして識別するために、様々な特徴が抽出される。

ブロック識別及びラベル付け

ここで図10を参照すると、ラベル付けモジュール131によるブロック識別処理の流れ図が示されている。ラベル付けモジュール131は、1次イメージ・バッファ141を介してデータ準備モジュール129からブロックのリスト、及びそれらの識別値と共に、スムーズにされた、水平線又は垂直線のないイメージを受信する。ラベル付けモジュール131は、そのイメージ内の各ビットを前述のように、2バイト表現に拡張する（1001）。追加ビット幅は、識別されたシステム・ブロックに関する情報を記憶するのに使用される。拡張されたデータはアドレス指定可能メモリ113に記憶される。

ラベル付けモジュール131は、設定されたピクセルを識別するために、そのイメージの拡張されたデータを走査する（1003）。実際にイメージ・データに対応する拡張されたデータ内のビットだけがチェックされる。この走査処理は左から右、上から下に進み、（解像度に基づく）そのイメージ内の線の数のカウントが、全イメージが走査された時を判定するために保持される。

設定されたピクセルが見つかり、同じ線内のその設定されたピクセルのすぐ隣のピクセル、及び既にチェックされている前の線（従って、左、左上、上、及び右上）が、それら全てが同じブロック識別番号を有しているかどうかを判定する（1011）ために、チェックされる（1009）。全てが同じである場合、その設定されたピクセルは同じその識別番号に割り当てられる（1013）。ラベル付けモジュール131は、現在の線がイメージ内で第1の線か、又は空白線に隣接しているかを判定する（1005）。この条件を満たせば、すぐ左のピクセルだけが上記のようにチェックされる。

ブロック識別番号が隣接するピクセル間で異なる場合、ラベル付けモジュール131は、設定されたピクセルが複数のブロック識別番号を有するかどうか判定する。この条件を満た

す場合、ラベル付けモジュール131は、そのピクセルに、隣接するピクセルのうちで最も多くのピクセルに保持されているブロック識別番号を割り当てる（1021）。また、他の隣接するピクセルはどれも、同様にこうして変更されたブロック識別番号をブロック識別番号として有することになる。ブロック識別番号は、利用可能なブロック識別番号のリストから割り当てられる。従って、変更されたピクセルのブロック識別番号は、利用可能なブロック識別番号のリストに加えられる（1023）。従って、各ブロック識別番号は、連結されたピクセルの1組に関連づけられる。

ブロック識別番号が割り当てられると、ブロック識別番号に関する多くの統計が更新される（1025）。特に、ラベル付けモジュール131は、各ブロック識別番号に対して、以下のものを保持する。

ブロック識別番号を有するイメージ・バッファ内にある設定されたピクセルの数（NUMB）；

ブロックの左上、及び右下の設定されたピクセルによって判定された、ブロック内に広がるエリア（AREA）；

ブロック内で設定されたピクセルから未設定ピクセルに遷移した数（左から右／上から下に走査）（TRANS）；

ブロックの左上の設定されたピクセルと右下の設定されたピクセルのY軸座標の間の差としての、ブロックの高さ（HEIGHT）；

ブロックの左上の設定されたピクセルと右下の設定されたピクセルのX軸座標の間の差としての、ブロックの幅（WIDTH）；

NUMB/TRANSの商によって決定されたラン係数、ラン係数はブロック内の設定されたピクセルの平均ランレングスを示す（RUN）；

WIDTH/HEIGHTの商によって決定される縦横比（ASPECT）；及び

NUMB/AREAの商によって決定される密度係数、密度係数はブロック内のピクセルあたりの、設定されたピクセル平均数を示す（DENSITY）。

これらのデータ値は、それぞれ走査されたピクセルで更新される（1025）。

ラベル付けモジュール131は、イメージ内の全ての線が処理されたかどうか判定する（1027）。まだ処理されていない場合、ラベル付けモジュール131は、拡張されたデータの走査を続ける（1003）。全てが処理された場合は、連結されたピクセルの全ての別の、異なったブロックが識別され、ブロック識別番号が供給され、各ブロック識別番号が、連結されたピクセルの1グループのみに適用される。

ブロックの全てに、ブロック識別番号が割り当てられると、ラベル付けモジュール131は、全体イメージに関する平均ブロック・データを判定するために選択されたブロックを適格にする（1029）。適格にされたブロックは、システムによって容易に識別できる適当なサイズと形状を有するブロックである。好適実施例では、50dpiのイメージを使用すると、適格にされたブロックは以下の基準を満たす。

HEIGHT/RUN > 4.00

HEIGHT <= 22行

ASPECT >= 5.00

DENSITY >= 0.5

各ブロックはこれらの基準に関してチェックされ（1029）、ブロック識別番号のリストが、そのブロックが適格かどうか示すために更新される。この値は好適な解像度に関連しており、他の実施例では、他の基準が使用されうる。

適格にされたブロックから、高さの平均とランレングスの平均が計算される（1031）。高さの平均は適格にされたブロックの高さを合計し、その適格にされたブロックの数によって除算されることによって求められる。ランレングスの平均も同様にして求められる。

図27は、図26のイメージに関してラベル付けを行った結果を概略例示している。図27のイメージにおいて、同じピクセル密度パターンを有するブロックは、1つのブロックの部分としてラベル付けされ、同じブロック識別番号を有する。

ブロックの分類

ラベル付けモジュール131は次に、各ブロック（上述のように適格にされたものと適格でないものの両方）のデータ・タイプの判定を続け、それがテキスト、イメージ、又は2重のうちどれであるかを判定する。図11を参照すると、ラベル付けモジュール131のデータ・タイプ分類処理の流れ図が示されている。

ラベル付けモジュール131は、適格判定処理によって適格にされたブロックが存在するかどうかをテストする（1100）。適格にされたブロックがなければ、ラベル付けモジュール131は全てのブロックを無効ブロックと2重ブロックに分類する。ラベル付けモジュール131は、ブロックのリストにわたってループし（1101）、各ブロックに対し、そのエリア（AREA）が100ピクセル（50dpiと仮定して）より小さいかどうか判定する（1103）。該当するブロックがあれば、そのブロックは無効ブロックで、処理が続けられる。そのエリアが100ピクセルより大きい場合、そのブロックは2重ブロックとして分類される（1105）。
どちらの場合でも、ブロックのリストは割り当てられた値で更新される。

ブロックが最初に分類されると、又は、適格にされたブロックがある場合、ラベル付けモジュール131は、そのブロックがテキスト又はイメージ・ブロックであるか、又は単に不確定なものであるかを判定する。これは、適格にされたブロック、又は2重ブロックのリストにわたって、及び各ブロックに対してループする（1109）ことによって実施され、ブロックの高さが、イメージのブロックの高さの平均から導かれたブロックの高さの特定の範囲内にあるかを判定する（1111）。好適実施例では、好適なイメージ解像度が50dpiである場合、テストは、各ブロックの高さがブロックの高さの平均の4倍以下であるかどうかというものである。他のイメージ解像度では他の近似比率が用いられうる。

ブロックの高さが所望の範囲内であれば、このことはテキスト・データやイメージ・データでありうるブロックの媒体サイズに対し小さいブロックであることを示す。従って、ラベル付けモジュール131は更に、適格にされたブロック全ての平均ランレングスに関するブロックのランレングス（RUN）を判定する（1113）。好適実施例では、ブロックのランレングスが平均ランレングスのおよそ6倍＋調整係数2より小さい場合、そのブロックはテキスト・ブロックであり、それに従ってブロック識別リストが更新される（1115）。これは、そのブロックが、一旦データ準備によってスムーズにされたテキスト・ブロックと共通のエリアと長さを有するかどうか効果的にテストする。これらの様々なテストを満足させたブロックは、テキスト・ブロックである可能性が高いので、そのように分類される。

ランレングスがこの値を超えてもまだ、ブロックはイメージ・ブロックかまたは不確定なものである可能性があり、この場合、そのブロックは2重ブロックとして分類されたままとなる。従って、ラベル付けモジュール131は、ブロックの高さ（HEIGHT）が3行を越えるかどうかテストする（1117）（ここでも、50dpiと仮定し、この値は他の解像度でも基準付けされる）。ほとんどのテキスト・ブロックは、この値より小さいので、高さがこの値を超える場合には、そのブロックはイメージ・ブロックである可能性が高く、それに従ってブロック識別リストが更新される（1119）。そうでない場合、これ以上の分類は行われず、ブロックは2重ブロックとして分類されたままとなる。

条件1111に戻って、ブロックの高さがブロックの高さの平均のおよそ4倍より高い場合、そのブロックはイメージ・ブロックである可能性が高いものの、この分類を確認するための更なるテストは行われぬ。ラベル付けモジュール131は、ブロックの縦横比（ASPECT）がおよそ0.2以上かどうか判定する（1121）。およそ0.2以上であれば、このことは、ほとんどのイメージで共通な、高さに対する近似幅をブロックが有することを示し、より小さい値はイメージである可能性の低い非常に狭いブロックを示す。従って、ラベル付けモジュール131は、イメージ・ブロックを示すようにブロック・リストを更新する（1119）。

このテストが不成功に終わると、ラベル付けモジュール131は、ブロックの幅が3ピクセルより大きいかどうか判定する（1123）。3ピクセルより大きい場合、そのブロックはここでもイメージ・ブロックとして分類される（1119）。それ以外の場合、ブロックは2重ブロックとして分類されたままとなる。

ラベル付け、及び分類処理の終わりで、イメージ内のブロックは、テキスト、イメージ、

10

20

30

40

50

又は2重ブロックのどれかに明確に識別、及びタイプ分けされる。結果のブロックの組は、「システム・ブロック」であり、テンプレートの構造を定義するのに用いられる。

H. 動的データ・ネットワーク

再び図3を参照すると、テンプレートの構造を記述する動的データ・ネットワークが、有効な、ラベル付けされ分類されたブロックから定義される(317)。ここで図12を参照すると、テンプレート・モジュール133によるテンプレートの動的データ・ネットワーク作成処理の流れ図を示している。

テンプレート・モジュール133は、任意のシステム・ブロックが任意のユーザ定義領域と交差するかどうかを判定する(1200)。これらのシステム・ブロックは更なる処理から除かれる。テンプレート・モジュール133はまた、設定されたピクセルの現れた、傾斜除去されサブサンプリングされたイメージの第1の線を判定し、設定されたピクセルの最後の線を判定する。これらの線は任意のデータが存在するイメージの領域を判定し、それによってこれらの線の外側のイメージ・エリアを処理する必要性をなくす。

テンプレート・モジュール133は、最初の線と最後の線の間のイメージ・データを、同じ領域を有する多くの領域にセグメント化する。イメージを複数セグメントにセグメント化することによって、各セグメント内のブロック・サイズを標準化することができ、大きなブロックが動的データ・ネットワークに関するブロックの選択に過度に影響することがない。セグメントを使用して、イメージの各領域内のブロックがテンプレートで使用され、更に強力なテンプレートが導入される。好適実施例では、イメージが15のセグメントにセグメント化され、イメージを上から下まで分割する。

各セグメントで、テンプレート・モジュール133が、テキスト・ブロックの平均領域を判定し、個別にセグメント内のイメージ・ブロックを判定する(1203)。各セグメント内の各ブロック・タイプの平均エリアが次に記憶される。テンプレート・モジュール133は次に、ブロック・タイプに関連して、セグメントの平均ブロック・エリアとほぼ同じ領域を有するこうしたブロックを、各セグメント内から選択する(1205)。好適実施例では、選択されたブロックは、そのタイプの平均ブロック・エリアの±20%の範囲内である。代替案として、平均ブロック・サイズより小さく、これに近い(順序づけられたリストにおいて)エリアを有するブロック、及び平均ブロック・サイズより大きく、これに近いエリアを有するブロックが選択されうる。例えば、セグメント内のブロックが104、100、118、132、138、102、及び128の四角のピクセルからなるエリアを有すると仮定すると、平均ブロック・エリアは120ピクセルになる。ここで選択されるブロックは118、及び128の四角のピクセルのブロックとなる。この選択はイメージ・ブロックとテキスト・ブロックに関して行われ、各セグメントに4つのブロックが選択される。

テンプレート・モジュール133は次に、少なくとも5つのブロックが選択されたかを判定する(1207)。そうであれば、テンプレートに関する動的データ・ネットワークが作成されうる。経験的に、5つのブロックのうち最小のものが動的データ・ネットワークを作成するのに適していると判定されている。5つのブロックが選択されなかった場合、テンプレート・モジュール133は、書式処理定義執行部121に、テンプレート定義が不成功であることを通知する。

全体イメージに関して5より多いブロックがある場合、そのイメージに関する動的データ・ネットワークが作成される(1211)。動的データ・ネットワークは、平均サイズ要件を満たすシステム・ブロック間、好ましくはそれらの中心間のベクトルのリストを含む。より詳しくは、各システム・ブロックに関して、システム・ブロックの中心から他の全てのシステム・ブロックに対してベクトルが定義される。従って、各システム・ブロック B_i に関して($i=1$ から n 、ここで n はシステム・ブロックの数)、システム・ブロック B_i とシステム・ブロック B_j の間のベクトル $V_{(i,j)}$ を含むベクトル・リスト V_i が存在する。

$$V_i = [V_{(i,j)}, V_{(i,j+1)}, \dots, V_{(i,n)}]$$

好適実施例では、各ベクトルはシステム・ブロックの中心間の距離、及び方向として定義されるのが望ましい。

$$V_{(i,j)} = [((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}, (B_i \rightarrow B_j)]$$

ここで $(B_i \rightarrow B_j)$ はシステム・ブロック B_i からシステム・ブロック B_j へのベクトルの方向を示す。

ベクトルが、あるシステム・ブロックを別のブロックに不変に関連づける限り、他のベクトル定義も使用されうる。例えば、各ベクトルは、必要に応じて距離と方向が計算できる、その両端の点の座標として記憶される。

$$V_{(i,j)} = [(x_i, y_i), (x_j, y_j)]$$

イメージに関する動的データ・ネットワーク D は、システム・ブロックに関するベクトル・リストの組 V を含んでいる。

$$D = [V_1, V_2, V_3, \dots, V_n]$$

イメージの動的データ・ネットワークは、システム・ブロック間のベクトルの関係がそのイメージに関して固定されており、イメージが回転されたり大きく移動された場合でさえも同じであるため、イメージの構造を記述する。図28aは、システム・ブロックとしてテキスト・エリアが使用されている書式の例を示し、図28bないし28hは各システム・ブロックと他のシステム・ブロックの間の様々なベクトル・リスト V_i を示している。

更に、動的データ・ネットワーク D の一部はユーザ定義領域に関して定義される。これは、各ユーザ定義領域 Z_k ($k=1$ からユーザ定義領域の数 m) に対して、領域 Z_k と各システム・ブロック B_i の間のベクトル $U_{(k,i)}$ を含む、ベクトル・リスト U を確立することによって行われる。

$$U_i = [u_{(i,j)}, u_{(i,j+1)}, \dots, u_{(i,m)}]$$

この実施例では、テンプレートを極めて強力にするユーザ定義領域の間のベクトルが見つかっておらず、従って記憶されない。代替実施例では、テンプレートを改良する、これらのベクトルが使用される。どちらの場合でも、ユーザ定義領域の間のベクトルが含まれれば、動的データ・ネットワーク D が更に、ユーザ定義領域に関するベクトル・リストの組 U を含むことになる。

$$D = [V_1, V_2, V_3, \dots, V_n, U_1, U_2, U_3, \dots, U_m]$$

テンプレート・モジュール133は次に、作成されたテンプレートをユーザが入力した名前及び、テンプレート内の各システム・ブロックの位置、各システム・ブロックに関する統計データ、動的データ・ネットワーク、及びユーザ定義領域の座標を含む、定義されたテンプレート・データと共に記憶する (1213)。テンプレート・モジュール133は、制御を定義執行部121に戻す。

III. テンプレート登録

再び図2を参照すると、1つ以上のテンプレートが定義された後 (201)、ユーザはそのテンプレートで文書の処理を開始できる。テンプレートの選択、及び文書の入力とは上述した。テンプレートの適用は、テンプレートを使用して文書を自動的に登録する処理である。

一般に登録処理は、新しいテンプレートを作成しないで、各入力イメージ内で識別されたシステム・ブロックがテンプレートのシステム・ブロック及び動的データ・ネットワークと比較されること以外は、テンプレート定義処理に非常に似ている。一致した場合、システムはテンプレートのユーザ定義領域を入力イメージに適用し、そこから光学文字認識を用いて任意のテキストを抽出する。

より詳しくは、図13aと13bを参照すると、登録処理の流れ図が示されている。登録で用いられるモジュールは以前に記載されており、その基礎的動作も事実上同じである。登録処理は登録執行部135によって管理される。

登録執行部135はスキャナ115とスキャナ制御モジュール139を制御し、入力文書を走査し、イメージ・データを得る (1301)。このイメージ・データは前述のように閾値操作され、2色イメージを作成する。登録執行部135は縮小モジュール123を発行し、イメージ・データを所望の解像度に縮小化する (1303)。

登録執行部135は、傾斜検出モジュール125を発行し、イメージ内のどんな傾斜も検出し (1305)、イメージ傾斜角と方向を報告する。登録執行部135は、その傾斜が 1° を越えているかどうか判定し (1306)、越えていれば傾斜除去モジュール127をコールして縮小化

10

20

30

40

50

されたイメージと元のイメージ・データの傾斜を除去する（1307）。縮小化されたイメージは1次イメージ・バッファ141に記憶される。

登録執行部135はデータ準備モジュール129を発行して水平線及び垂直線を除去し（1309）、そのイメージをスムーズにし、個々のデータ・ブロックにする（1311）。次にラベル付けモジュール131が、ブロック識別番号でデータ・ブロックをラベル付けし分類するために発行される（1313）。ラベル付けされたブロックは上述のように適格にされ、適格にされたブロックのリストを生成する。適格にされたブロックは上述のように、テキスト、イメージ、又は2重ブロックのどれかに分類される。

登録執行部135はテンプレート・モジュール133をコールし、そのモジュールはテンプレートの動的データ・ネットワークをロードする（1315）。次にテンプレート・モジュール133は入力イメージに対する動的データ・ネットワークを生成する。これは、分類されたデータ・ブロックに更なる適格化を行う（1317）ことによって実施される。この適格化はテンプレートのシステム・ブロックと一致する潜在的なシステム・ブロックであるブロックを識別する。入力イメージ内の各データ・ブロックは、テンプレート内の各システム・ブロックと比較される。適格化の基準は、

データ・ブロックの分類が同じであること（テキスト又はイメージ）、

ブロック幅の差の絶対値が5ピクセル以下であること、

ブロックの高さの差の絶対値が5行以下であること、及び

データ・ブロックとシステム・ブロック内の設定されたピクセルの数の差の絶対値が、システム・ブロック内の設定されたピクセルの数の10%より小さいこと、である。

テンプレート・モジュール133は、入力イメージ内の適格にされたシステム・ブロックの数が、テンプレート内のシステム・ブロックの数の少なくとも20%であるかどうかテストする（1319）。上記条件を満たさない場合、テンプレートは入力イメージと一致せず、テンプレート・モジュール133は、不成功の状態で登録執行部135に戻る（1333）。

適格にされたシステム・ブロックの数が閾値を満たせば、テンプレート・モジュール133は上述したように入力イメージに関する動的データ・ネットワークを作成する（1321）。この動的データ・ネットワークはテンプレートとしては記憶されず、むしろユーザに選択されたテンプレートの動的データ・ネットワークと比較される（1323）。

この比較処理1323は入力イメージの動的データ・ネットワーク内の各ベクトル・リストにわたって繰り返され、即ち入力イメージ内の各システム・ブロックにわたって効果的に繰り返される。各イメージ・ベクトル・リスト内のベクトルは、テンプレート・ベクトル・リスト内のベクトルと比較される。入力イメージ内のベクトルは、それが寸法誤差の所定範囲に入る場合、テンプレート内のベクトルと一致する。50dpiのイメージを用いた好適実施例では、寸法誤差は3ピクセルである。即ち、入力イメージ・ベクトルの長さがテンプレートのベクトルの長さの±3ピクセルの範囲にあれば、ベクトルは一致する。他の実施例では、適当に基準とされた他の寸法誤差が使用されうる。ベクトルは両端の点の座標によっても比較され、寸法誤差はほぼ2乗された値となっている。テンプレート・モジュール133は一致したベクトルの数のカウントを保持する。

テンプレート・モジュール133は次に、一致したベクトルの数が、つまり一致したシステム・ブロックが、テンプレート内のシステム・ブロックの数の少なくとも10%であるかどうか判定する（1325）。上記条件を満たす場合、動的データ・ネットワークは一致し、登録は成功する。

入力イメージの動的データ・ネットワークとテンプレートの間で一致があれば、対応するシステム・ブロックの各対（1つはテンプレート内、1つは入力イメージ内）の中心の間の(x,y)オフセットが判定される（1327）。これは、オフセット値(x,y)のリストを生成する。最も多い数のシステム・ブロックの対に適用するオフセット値がイメージのオフセットとして採用される。例えば、5つのシステム・ブロックの対がオフセット(2,3)を有し、2つのシステム・ブロックの対がオフセット(2,4)を有し、6つのシステム・ブロックの対がオフセット(3,3)を有する場合、イメージ・オフセットは(3,3)のピクセルである。イメージ・オフセットはユーザ定義領域のそれぞれの座標に加えられ、テン

10

20

30

40

50

プレートに記憶され、入力イメージ上のユーザ定義領域の座標を求める(1329)。
 入力の中のユーザ定義領域の座標は、光学文字認識モジュール145に提供される(1331)。
 次に光学文字認識モジュール145は、イメージ上のこうした領域でテキストを抽出し(図2:209)、そのデータをファイル、データベース、又は他の記憶装置に供給する。
 テンプレートとの一致がない場合、登録執行部135は不成功の状態でアプリケーション執行部137に戻り(1333)、ユーザにそのテンプレートが適用されなかったことを示す。このことは、文書が書式と同じタイプでないかまたは、イメージ内にテンプレートが導出される元のイメージ内より極めて多い、又は少ないピクセルが設定されているといった、全く異なる明るさとコントラストの設定で走査された場合に起きる。この場合、ユーザはスキャナ115の設定をチェックし、文書を再走査することができる。
 好適実施例では、マッチングが失敗すると、ユーザはアプリケーション執行部137によって、ユーザ定義領域の再描画を含む、テンプレートの再定義が(1335)可能になる。これによってユーザは、書式の変更によって、テンプレートを何度も調整し、進化させ、例えば新しいバージョンの基礎的な書式を作成する。登録執行部135は次に、変更されたテンプレートとの一致を再度試みる。ユーザは、そのテンプレートを再定義する必要がない場合、終了を選択することもできる。

10

【図1】

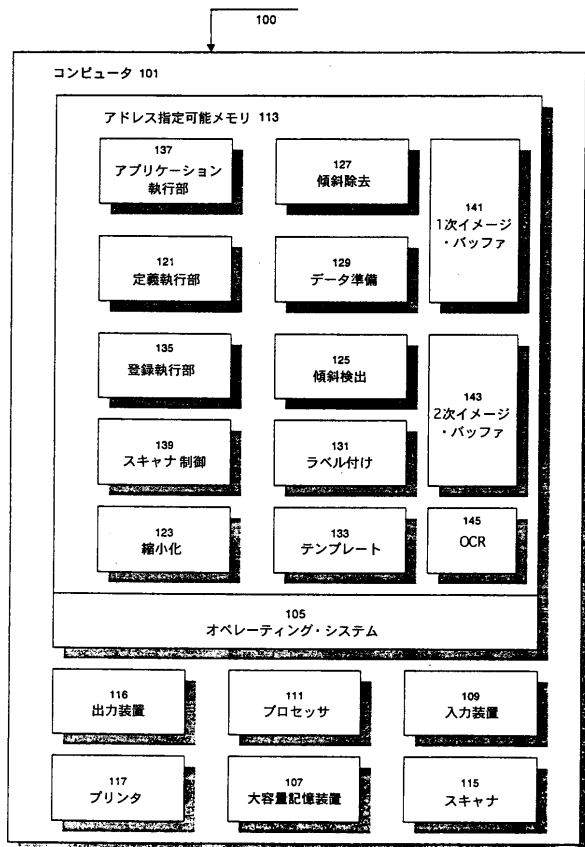


FIGURE 1

【図2】

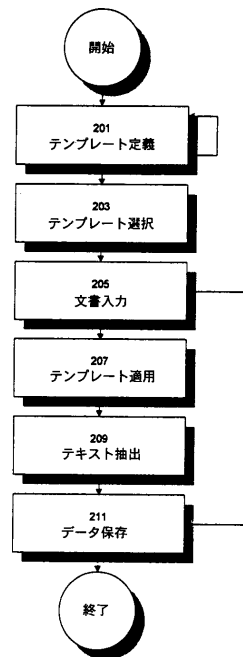
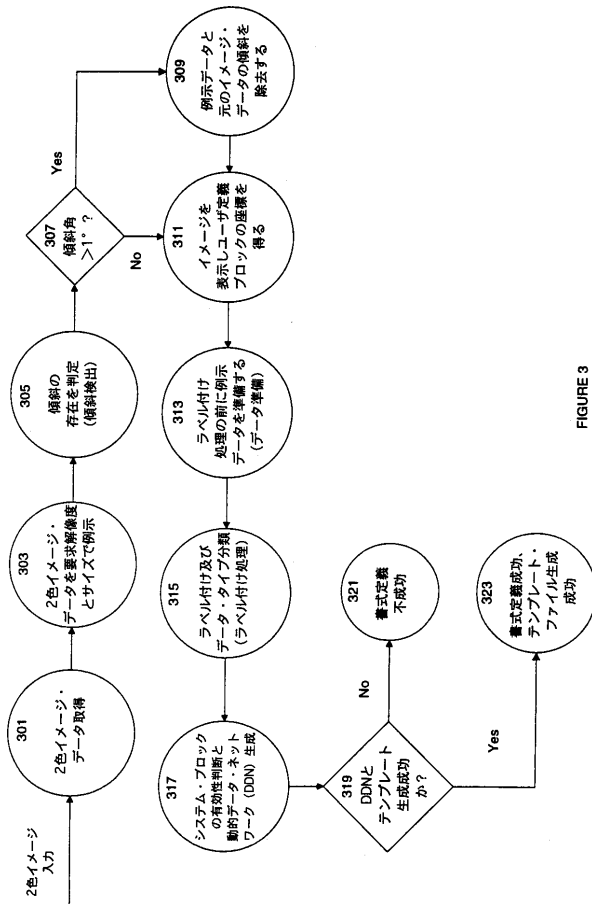


FIGURE 2

【図 3】



【図 4】

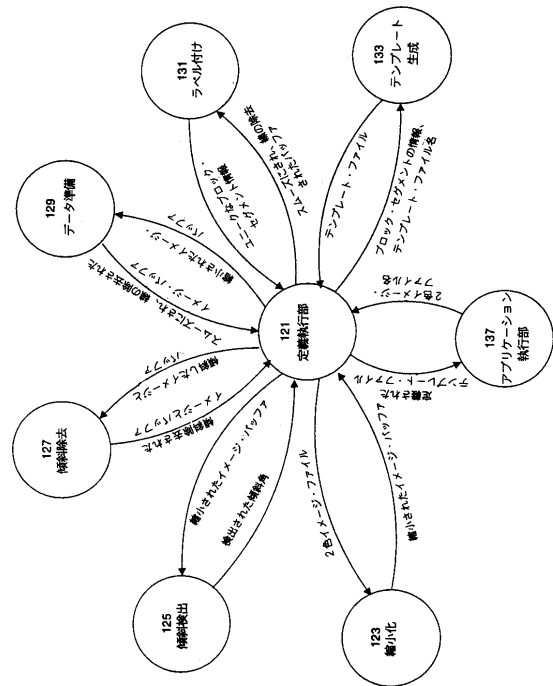


FIGURE 4

【図 5 a】

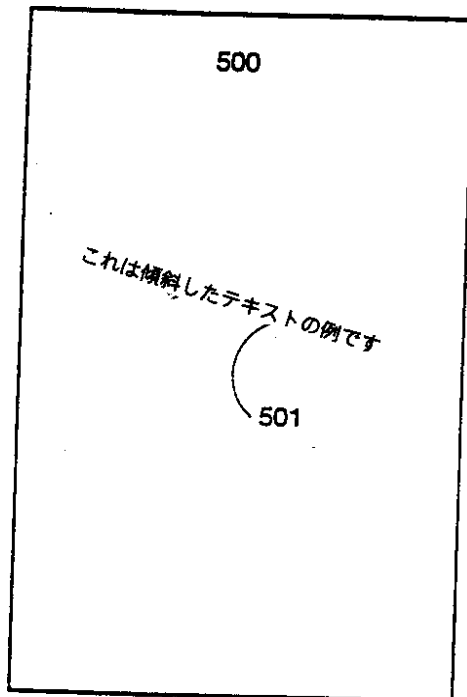


FIGURE 5a

【図 5 b】

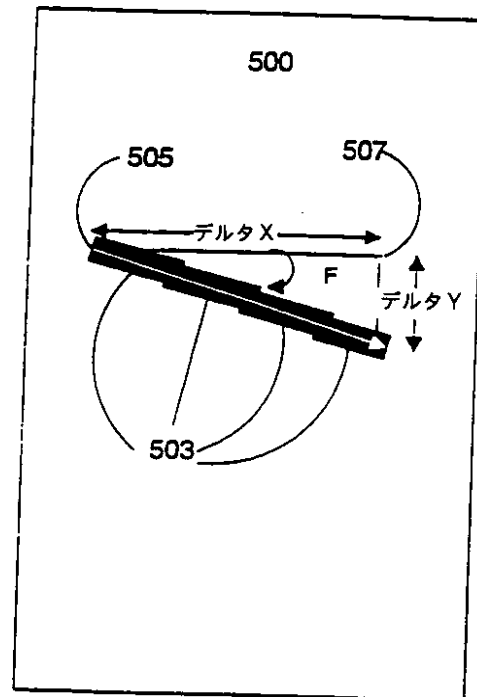
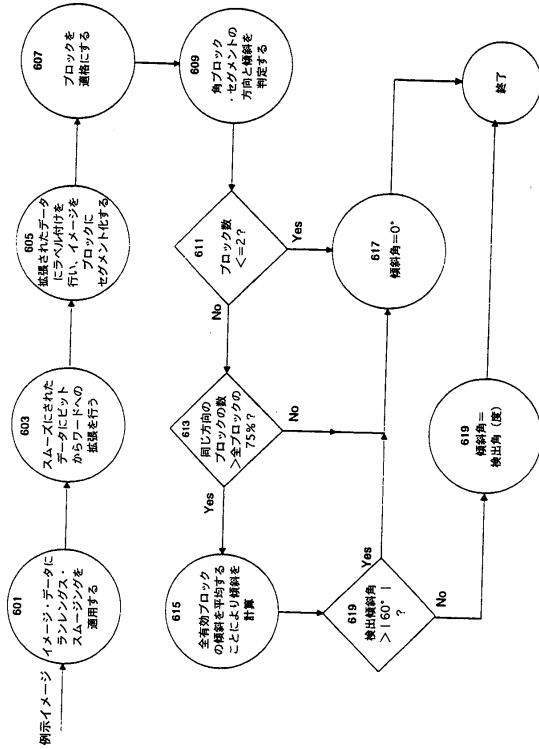


FIGURE 5b

【図 6】



【図 7 a】

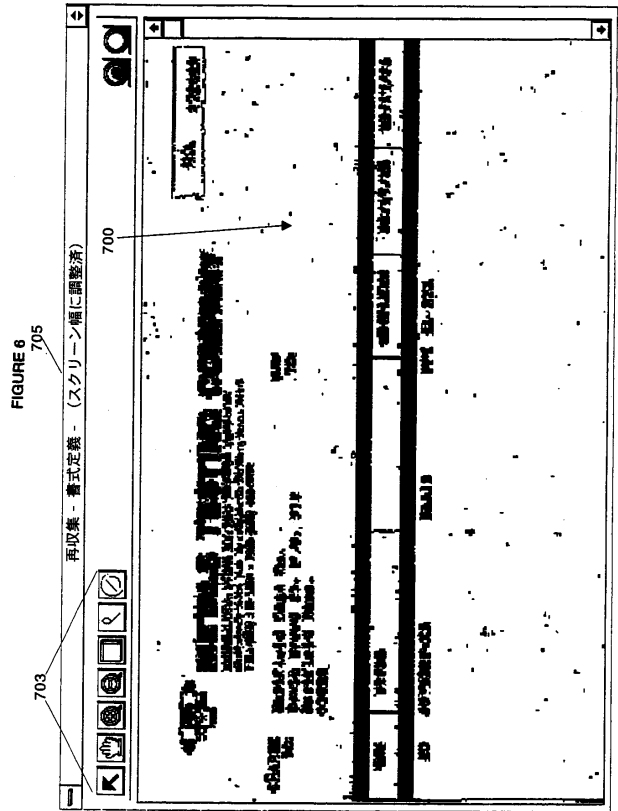
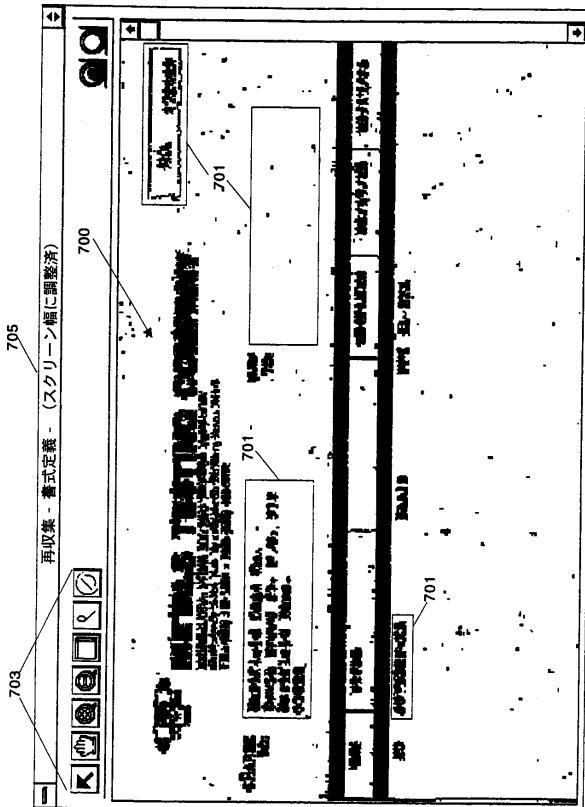


FIGURE 7a

【図 7 b】



【図 8】

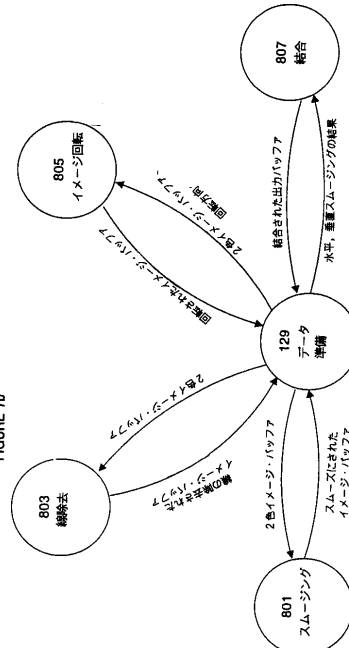
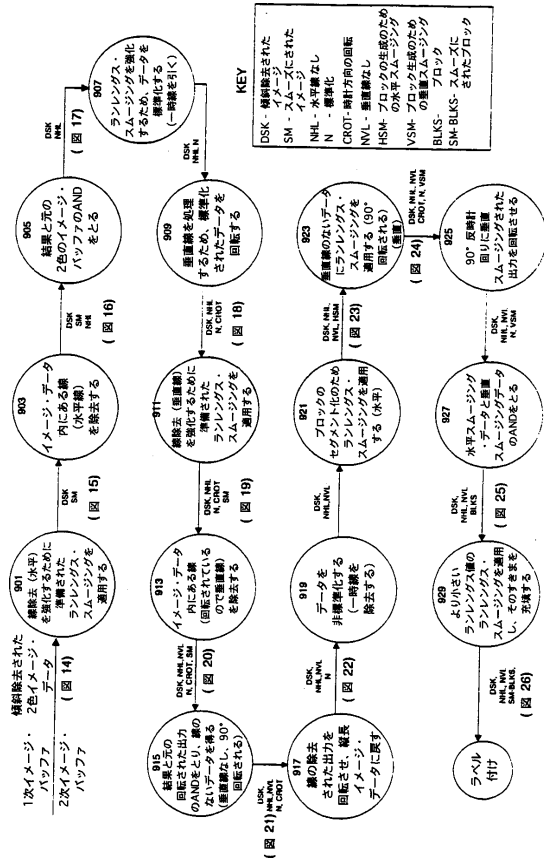


FIGURE 8

【図 9】



【図 10】

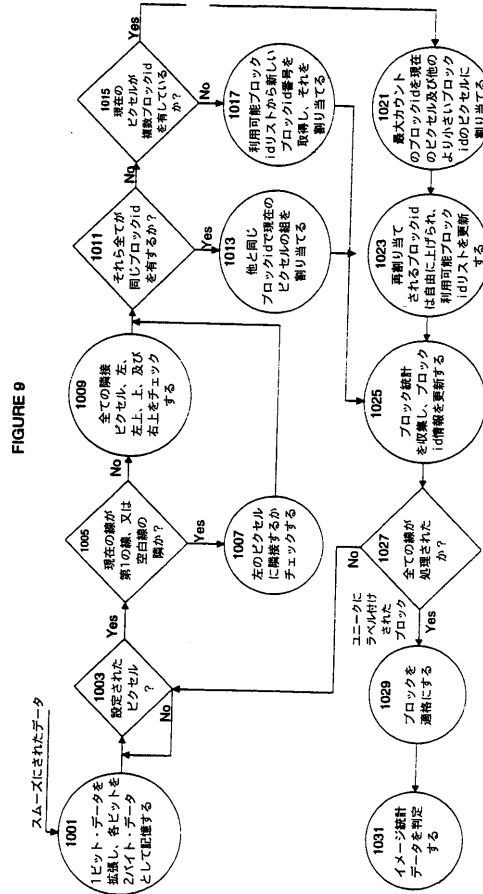


FIGURE 10

【図 11】

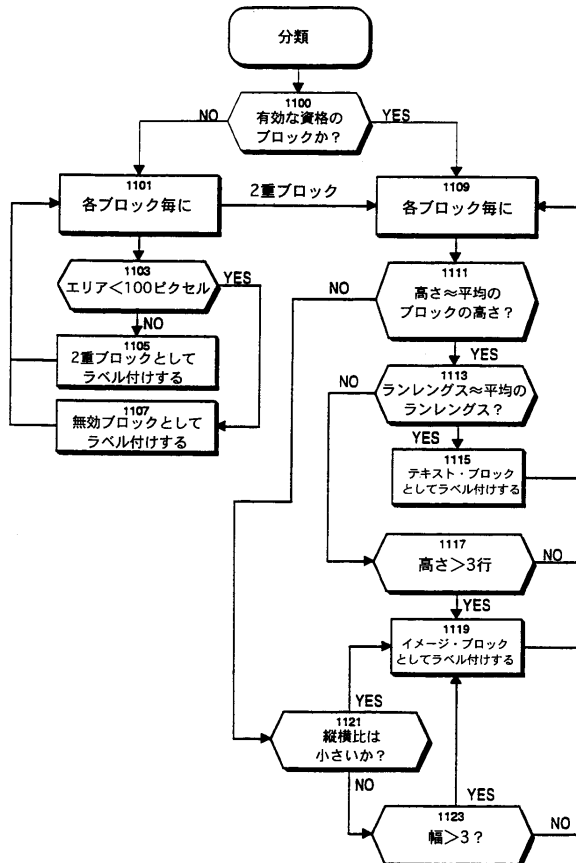


FIGURE 11

【図 12】

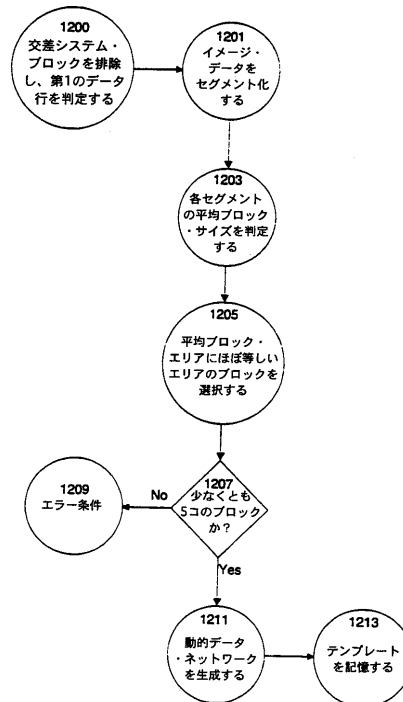
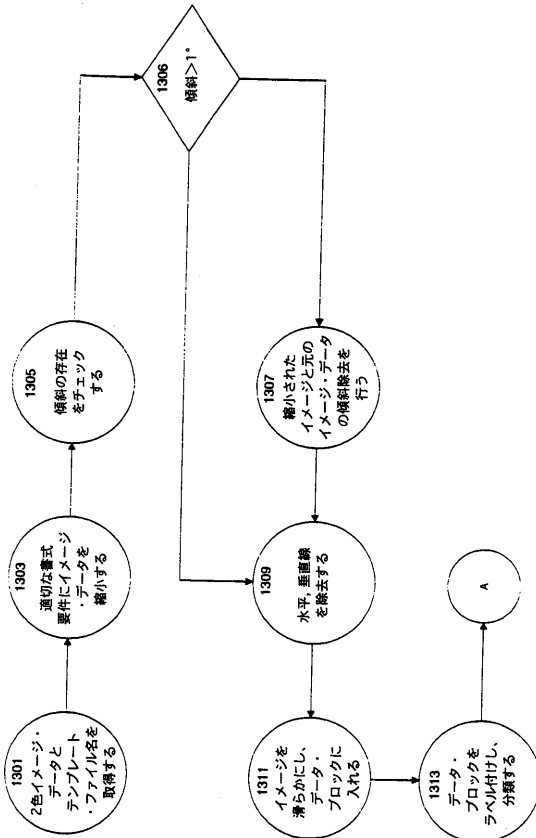
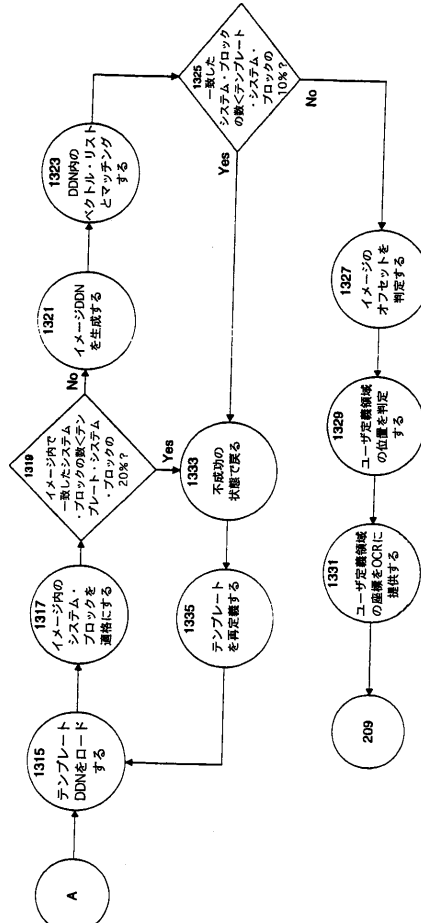


FIGURE 12

【図 1 3 a】



【図 1 3 b】



【図 14】

[illegible]

【図 15】

[illegible]

Figure 14

Figure 15

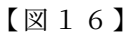


Figure 16

【図 17】

Figure 17

【图 18】

Figure 18

【図 19】

Figure 19

【図 20】

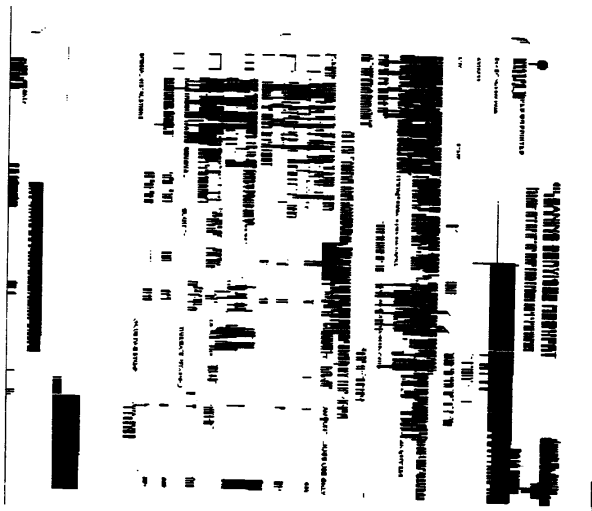


Figure 20

【図 21】

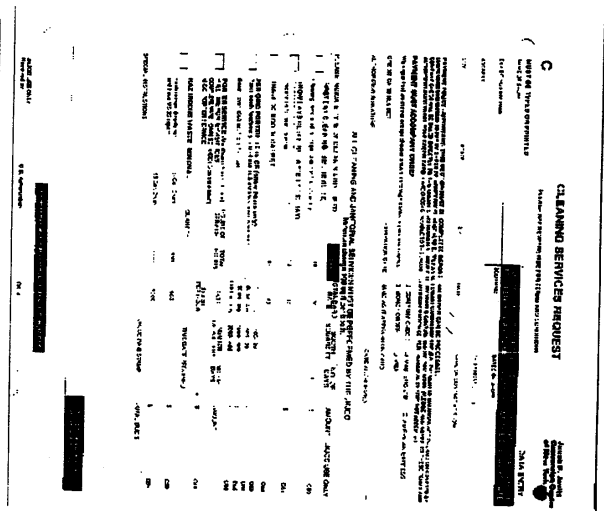


Figure 21

【図 22】

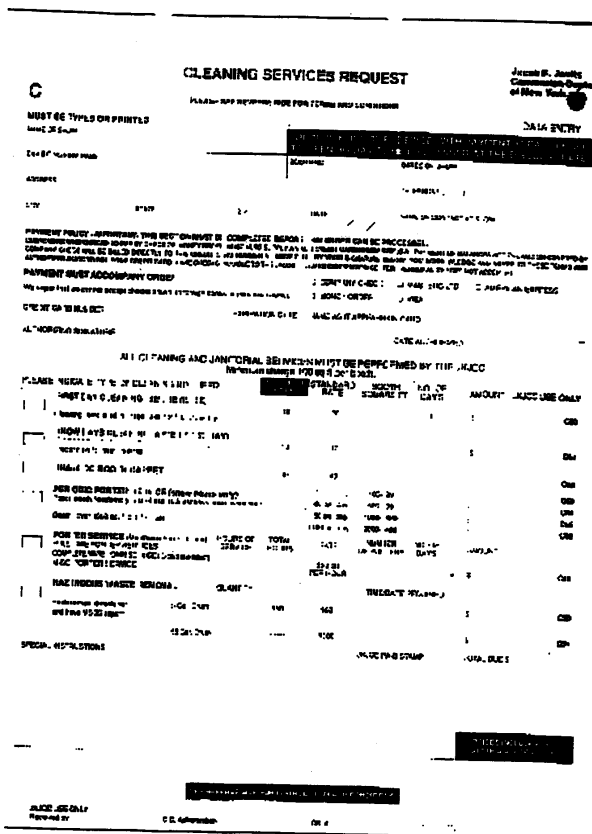


Figure 22

【図 23】

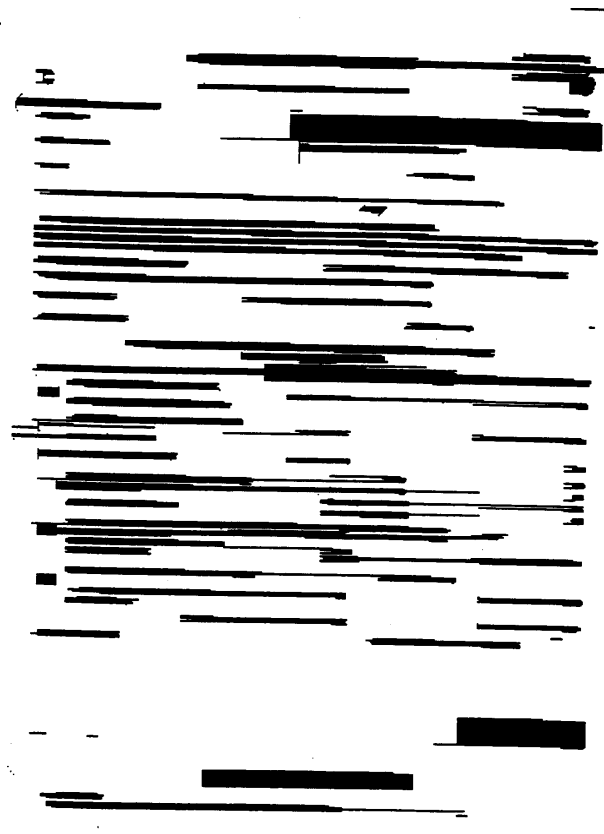


Figure 23

【図 24】



Figure 24

【図 25】

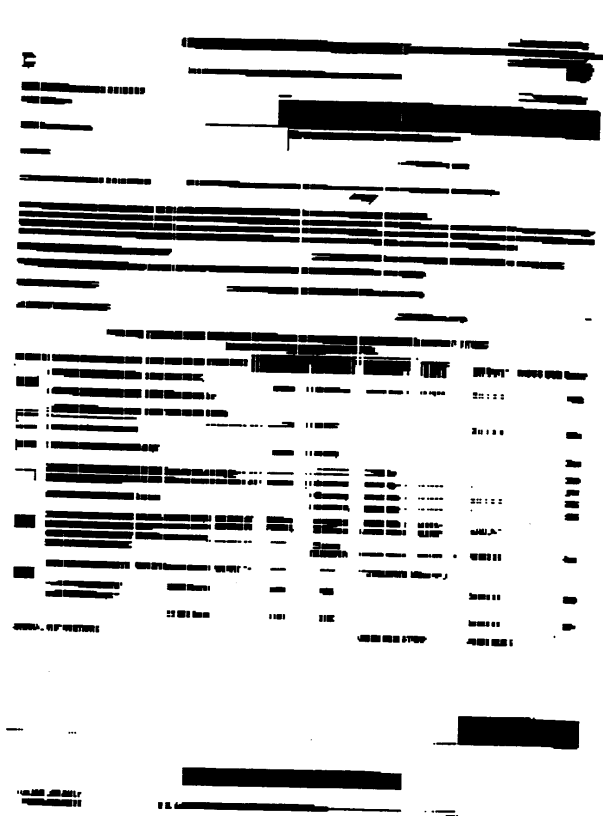


Figure 25

【図 26】

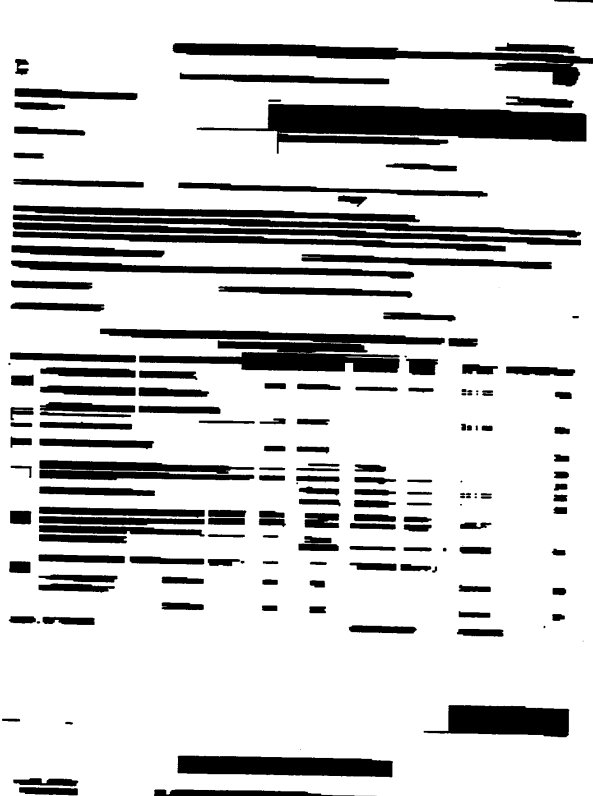


Figure 26

【図 27】

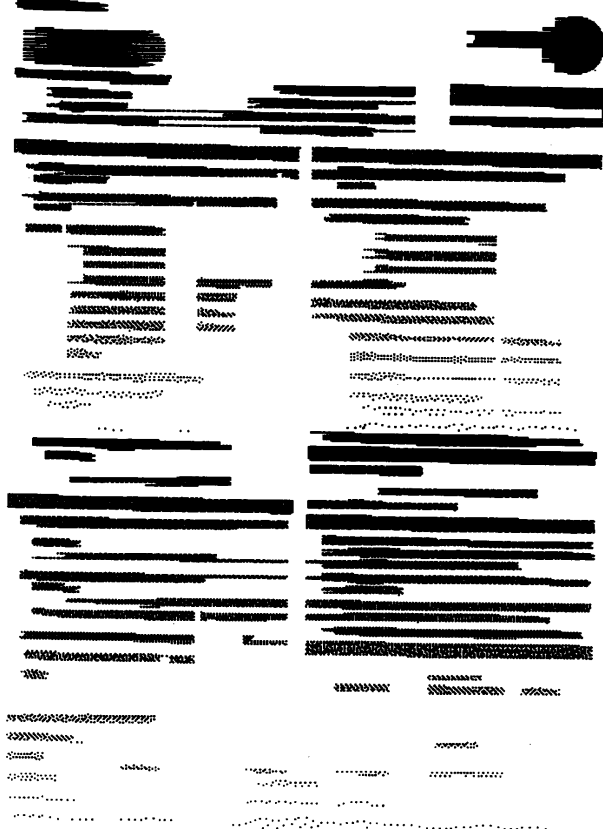


Figure 27

【図 28 a】

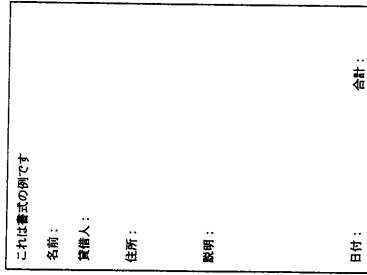


FIGURE 28a

【図 28 b】

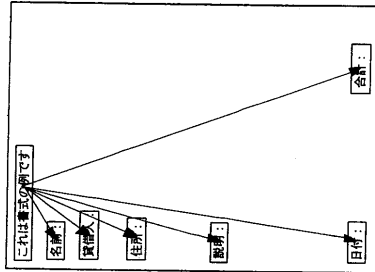


FIGURE 28b

【図 28 c】

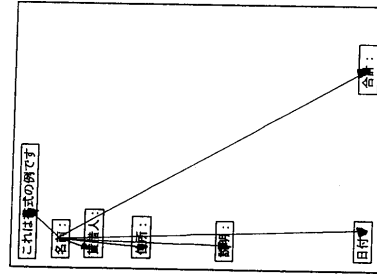


FIGURE 28c

【図 28 d】

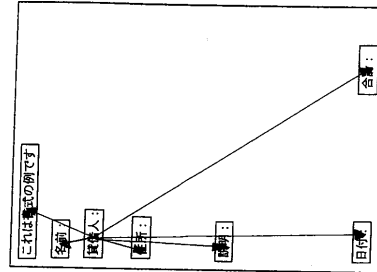


FIGURE 28d

【図 28 e】

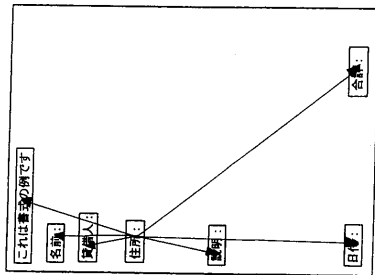


FIGURE 28e

【図 28 gg】

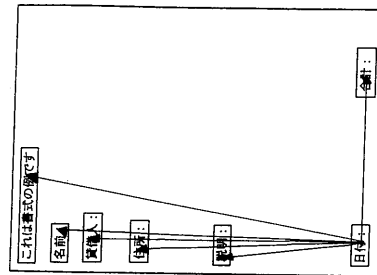


FIGURE 28g

【図 28 f】

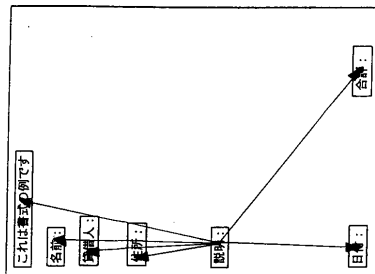


FIGURE 28f

【図 28 h】

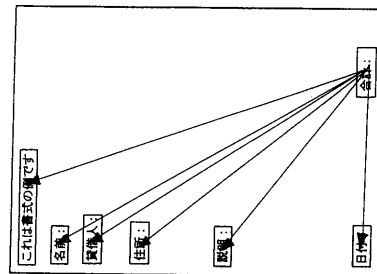


FIGURE 28h

フロントページの続き

(72)発明者 ランガラジャン, ヴィジャクマール
アメリカ合衆国カリフォルニア州95123 サン・ノゼ, ドン・ピザロ・コート・5509, エ
イヤスワミィ, ケアー・オブ・パーザサラシィー

審査官 飯田 清司

(56)参考文献 特開平06-195504 (JP, A)
特開平07-200731 (JP, A)

(58)調査した分野(Int.Cl., DB名)
G06K 9/00 - 9/32
G06T 7/00, 7/40