

光学文字読取装置の
アーキテクチャの研究

2001年3月

花野井歳弘

序

本論文は、筆者が株式会社日立製作所において 22 年間にわたり行った光学文字読取装置の研究開発のうち、装置のアーキテクチャに関する研究をまとめたものである。

本論文は次のような内容からなっている。

第 1 章では、光学文字読取装置における本論文の位置付けを行う。1.1 節から 1.3 節では、光学文字読取装置の入力装置としての位置付けを述べ、その構成・作用、利用状況などにつき概説する。1.4、1.5 節では、製品の歴史的経緯からソフトウェア化の動向をまとめ、1.6 節では本論文の位置付けを行う。

第 2 章では、光学文字読取装置のアーキテクチャの要点と課題を明らかにする。2.1 節では、光学文字読取装置に要請される役割を述べ、2.2 節ではこれに対応する光学文字読取装置のアーキテクチャの特質を述べる。2.3 節では文字認識処理の観点からハードウェアとソフトウェアとの対比を行い、両者の特質を明らかにする。2.4 節では、以上を踏まえて、光学文字読取装置のアーキテクチャをいかに構成するかを述べる。

第 3 章では、光学文字読取装置の開発手順からアーキテクチャの決定フェーズおよび決定手法を述べる。3.1 節では、開発手順を概説、方式決定の位置付けを述べる。3.2 節では方式決定の手法を述べる。動作方式の決定に基づき、採用された認識方式の実用化のため計算アルゴリズムの開発、ハードウェアとソフトウェアの最適化を図る。特に、計算アルゴリズムが重要かつ効果的であることを述べる。

第 4 章からは、上記で報告した手法で実現した例を述べる。

第 4 章では、可変閾値法による印刷文字認識処理において実現した相関計算の高速ハードウェアにつき述べる。4.1 節では前提となる仕様を、4.2 節では、可変閾値法実用化のため、その理論式をハードウェアを前提とした場合の計算アルゴリズムに展開、その内容につき述べる。また、ここでは、9 位置相関法、マルチスライス認識方式などの精度向上手法が加えられており、その内容についても述べる。4.3 節では、具体的に 1200 字/秒の高速を実現したハードウェアにつき述べる。

第 5 章では、非線形マッチング法における手書文字認識で課題となったダイナミックプログラミングによる照合計算のハードウェア化について述べる。5.1、5.2 節では、ダイナミックプログラミングの実用化のため開発された計算アルゴリズムを述べる。特に対角線追跡法の開発により大幅に繰返し処理の削減および単純化を実現した内容を述べる。5.3 節では具体的に実現したハードウェアにつき述べる。

第 6 章では、ソフトウェア主体の例を述べる。手書き文字認識に図形検証法を採用するに当たり、課題であった文字輪郭線の 1 次元チェーンコード変換処理をソフトウェア主体により実現した。6.1 節では、速度評価のため作成したモデルパターンを示し、6.2 節では新たな 2 次元画像の数値表現形式を述べる。これを用いて輪郭線追跡のフローを作成、評価し、ハードウェア論理の追加による処理速度向上を図った手法および結果を述べる。

第 7 章では、代表的な 2 次元画像処理であるマスク処理につき述べる、7.1、7.2 節では

前処理の内容とハードウェアとソフトウェアの対比を行う。7.3 節では、マスク論理の基本回路を述べ、7.4 節ではこれを用いた処理フローを述べる。

第 8 章では、今後の発展の方向を示す。8.1、8.2 節では、光電変換部と 2 値化部の現状分析を行い、8.3 節では今後の電子技術の発達に従い、2 値化部、認識方式および光電変換部の今後の予測を述べる。

第 9 章では、本論文全体のまとめを行う。

目次

第1章 緒言	1
1.1 光学文字読取り装置の位置付け	3
1.2 光学文字読取装置の利用状況	4
1.3 光学文字読取装置の構成と作用	5
1.3.1 光学文字読取装置の構成	5
1.3.2 光学文字読取装置の動作	7
1.4 製品の経緯	7
1.5 ソフトウェア化の動向	9
1.6 本論文の位置付け	10
第2章 アーキテクチャの要点と課題	11
2.1 光学文字読取装置の役割	13
2.1.1 入力データの信頼性	13
2.1.2 入力コストパフォーマンス	14
2.1.3 制限緩和	15
2.2 光学文字読取装置のアーキテクチャの特質	16
2.2.1 改善可能性（柔軟性）	16
2.2.2 高速度処理	17
2.2.3 低コスト	18
2.3 ハードウェアとソフトウェアとの対比	18
2.3.1 処理の特徴	18
2.3.2 処理の適性	19
2.3.3 アーキテクチャへの要請項目	20
2.4 ハードウェアとソフトウェアとの最適化	27
2.4.1 文字認識方式とその特性	27
2.4.2 最適化手法のアプローチ	31
2.5 第2章のむすび	33
第3章 アーキテクチャの決定手法	35
3.1 光学文字読取装置開発手順	37
3.2 方式設計	39
3.2.1 動作方式	39
3.2.2 各部方式の決定	46
3.2.3 認識方式の決定	46
3.2.4 詳細アーキテクチャ設計	47
第4章 高速関連器	51
4.1 印刷文字認識仕様	53
4.2 可変閾値法の理論式と計算アルゴリズム	54

4.2.1	可変閾値法の理論式	54
4.2.2	可変閾値法の計算アルゴリズム	55
4.2.3	追加アルゴリズム	56
4.3	詳細アーキテクチャ設計	58
4.3.1	可変閾値法の認識フローチャート	58
4.3.2	相関部のフローチャート	59
4.3.3	詳細アーキテクチャの決定	59
第5章	非線形マッチング法における照合論理	69
5.1	ダイナミックプログラミングの理論式	71
5.2	ダイナミックプログラミングの再現	71
5.3	計算アルゴリズム	72
5.3.1	シミュレーションプログラムの問題点	72
5.3.2	計算アルゴリズムの開発	72
5.4	ハードウェアの構成	75
5.4.1	構成	75
5.4.2	動作	79
第6章	輪郭線追跡処理のソフトウェア化	81
6.1	モデルパタン作成	83
6.2	輪郭線追跡処理	86
6.2.1	輪郭線座標表現形式	86
6.2.2	輪郭線追跡フローチャート	89
6.2.3	単一動作プロセッサのアーキテクチャと輪郭線追跡フローチャート	89
6.2.4	プロセッサ機能のハードウェアによる強化	90
6.2.5	効果	95
第7章	前処理部	97
7.1	前処理部の処理内容	99
7.2	ハードウェアとソフトウェアとの対比	101
7.3	マスク論理の基本回路	102
第8章	今後の動向	107
8.1	光電変換部、2値化部の処理内容	109
8.2	光電変換部と2値化部の構成	112
8.3	今後の動向	113
8.3.1	2値化部のソフトウェア化	113
8.3.2	認識方式の動向	114
8.3.3	制限緩和	114
第9章	結 言	115
	謝辞	122
	引用文献	123

第 1 章

緒 言

第 1 章 緒 言

本章では、光学文字読取装置（Optical Character Reader : OCR）のコンピュータ周辺機器としての位置付けおよび構成と作用を概説し、本論文の位置付けを述べる。

1.1 光学文字読取装置の位置付け

コンピュータを利用した情報処理は、不断の大きな進歩を遂げてきた。

この間、多くの大きな課題があり技術進歩により逐次解決されてきた。しかしながら、当初より課題とされ、現在でも十分な解決が見られていないものにインプットネックと称される入力の問題がある。

この課題は、機械であるコンピュータと、生物である“人”との全く異質な世界のインターフェースをどのように成立させるかという問題であり、特に、“人”からコンピュータへのインターフェースである入力作業の負担がコンピュータの役割の増大に比例して大きな問題として続いている。

この課題の解決のため多くの入力機器・手段が開発されてきたが、現在でも十分な解決が見られていない。

入力機器には、“人”がコンピュータに歩み寄ったキーボード入力の機器が古くから主流であり、現在もほとんどを占めている。しかしながら、この方式は“人”がコンピュータに合わせる方法であるため、“人”の負担が大きいことが問題となる。

このため、機器が人に歩み寄り人の負担を少なくする入力方法も開発されてきた。これらは、人の目を摸した紙上の文字を読取る光学文字読取装置、人の耳を摸した音声入力装置、あるいは、文字記入時点で入力するオンライン手書入力装置などがある。これらの機器は、装置内に人を摸した認識処理が組み込まれているため自動認識機械と呼ばれる。しかしながら、多くの研究がなされているにもかかわらず“人”と同等の認識能力を持つことは困難な課題となっている。この技術を適用した自動認識機械も認識能力では、“人”の能力に達していないが、ある種の制限を加えることにより実用に供されている機械が多い。

この自動認識機械の中で光学文字読取装置は、最も実用化に成功した装置である。光学文字読取装置は“人”の世界で最も重要な情報媒体である紙とそれに書かれた文字を人と共有し、コンピュータの入力データとして直接入力しようとするものである。この便利さから、紙が必然的に必要な業務における入力手段として他の自動認識機械よりも多くの成功をおさめている。

また、マークシートリーダあるいはバーコードリーダのように紙を媒体とする入力機械も存在するが、それらは紙を媒体とはしているが、認識する図形は文字でなく、人には読みにくいマークあるいはバーコードであり、紙上にコンピュータ用のデータを記入したものである。このため、入力データの信頼性は高く数多く普及しているが、“人”よりコンピュータに合わせたものであり、人の負担を犠牲にして妥協を図ったものである。

1.2 光学文字読取装置の利用状況

表 1.2.1 に光学文字読取装置の利用状況をまとめる。

表 1.2.1 光学文字読取装置の利用状況

業種／分野	ターンアラウンド	手書き文字入力
保険業	月払保険入金票	保険契約申込書
金融業	公金伝票 利札	集中振込票 総合振込票 稟議書
官公庁・自治体	市町村民税 固定資産都市計画税 特別徴収税納入済通知書	確定申告書
流通業 チェーンストア 百貨店 生活協同組合 通信販売	統一仕入伝票 百貨店協会統一商品券	せり販売原票 中元歳暮申込書 注文書 葉書注文書
学校	授業料払込票	入学願書 履修届
輸送業	配送伝票 タクシーチケット	
医療		検査依頼票 レセプト
電力ガス水道	検針票	配電帳票
製造業	注文票／現品票	検査票
出版業	書籍スリップ	
建築土木		積算データ
経理伝票		入出金伝票 旅費清算 棚卸票
人事勤休		勤休記録票 人事査定票 自己申告書 アンケート集計

(注) はイメージも入力し利用する。

光学文字読取装置は、表 1.2.1 に示すように、各業種で利用されている。ターンアラウンド処理は、主に印刷文字を読取るもので、大量の伝票の入力処理に欠くことのできない入力装置である。また、手書きデータの入力も各種申込書など取引・契約関係や経理・人事関係で用いられている。

近年の傾向としては、イメージデータの利用で、帳票のイメージファイリングやイメージの転写による配送伝票の発行など種々の業務に広がりを見せており、近々過半の業務がイメージと連携すると予想されている。

1.3 光学文字読取装置の構成と作用

1.3.1 光学文字読取装置の構成

代表的な光学文字読取装置の構成^{16),17)}を図 1.3.1 に示す。

光学文字読取装置は大きく紙面の画像情報を光学的に走査してメモリに入力するスキヤナ部と、メモリ上の画像信号を認識する認識部、および全体の制御を行なう制御部より構成される。

- スキヤナ部

スキヤナ部は、認識入力すべき文字が記入されている紙（帳票）の画像情報を電気信号に変換する部で、紙送り機構部および光電変換部より構成される。また、処理速度に影響が大きい部である。

- ▶ 紙送り機構部

紙送り機構部は、ホッパにセットされた帳票を1枚ずつ光電変換部に送り、文字認識終了後ナンバリング印字を行ないアクセプトあるいはリジェクトスタックに送出する。

紙送り機構部では、特にホッパから帳票を確実に1枚ずつ安定して送り出すことが求められる。

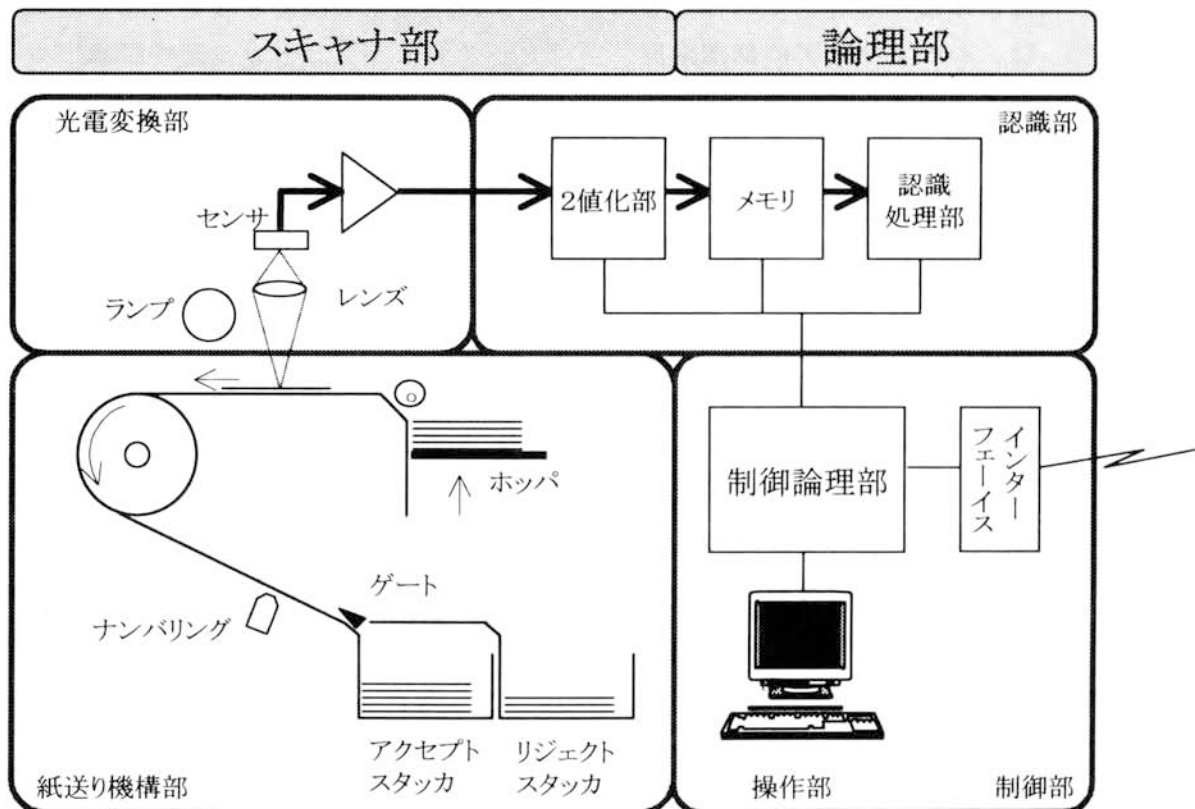


図 1.3.1 光学文字読取装置の構成

➤ 光電変換部

光学文字読取装置の名称となったように、帳票の紙面に光を照射し、反射光をセンサにより感知し、紙面の濃淡に比例したアナログ電気信号を得る。現在は、光源は蛍光灯、センサとしては1次元CCDセンサが使用されることが多い。

光電変換部では、認識精度を高めるため、最適な画像情報を得ることが求められる。このため、経時的および紙面上の場所的な変動を無くす処理が加えられる。

● 認識部

認識部は、スキャナ部により画像情報に変換された文字の画像を認識し、コードとして出力する部である。

➤ 2値化部

光電変換部で電気信号に変換された文字の画像を、2値のデジタル画像に処理しイメージメモリに格納する。2値化部は文字認識の精度に大きな影響を及ぼす部で、特に光学文字読取装置では文字認識しやすいような2値画像になるように工夫されている。具体的には、文字の濃淡によっても文字線の太さが変わらないように、また欠けなど少なくなるような対策が加えられている。

➤ 認識処理部

メモリに入力された文字画像を、採用された文字認識方式に従い、前処理・特徴抽出処理および照合処理などを経て認識結果を出力する部である。ここでは、ハードウェアの認識論理、プロセッサおよびソフトウェアの認識プログラムで実現されている。

● 制御部

制御部は、オペレータの操作および光学文字読取装置全体の制御を行なう部で、以前は専用プロセッサ^{9),10),13)}を設計製作することもあったが、現在ではパソコンが使われている。

➤ 操作部

通常、ディスプレイとキーボードからなり、光学文字読取装置の動作の指示、および認識データの確認修正をおこなう。

特に、認識データの確認修正は光学文字読取装置に独特な機能であり、文字認識段階で生じた不読および誤読を操作者が確認あるいは修正を行なう。操作を容易にし、専門要員でなくても従事できるように、認識結果と合わせて文字画像のイメージを表示するなど、光学文字読取装置に必須な機能として充実している。

➤ インターフェース部

文字認識処理によりコードに変換されたデータを出力する部で、回線での接続のほか、フロッピーディスクなど各種媒体に出力する。

1.3.2 光学文字読取装置の動作

光学文字読取装置の動作の概要を図 1.3.2 に従い説明する。

- ホッパに積まれた複数枚の帳票から 1 枚分離し、光電変換部に送る。光電変換部を通過した帳票は停止して、文字認識の終了を待つ。
- 光電変換部では、紙面上の濃淡を電気信号に変え 2 値化部で 2 値化し、メモリに入力する。
- 認識処理部ではこの画像を文字認識処理し、コードを出力する。
- 認識処理にて認識エラー（不読など）があった場合には、走査部でオペレータにより正しいコードに変換される。
- 帳票上の全ての文字の認識が終了すると、認識結果のコードはインターフェース部より出力、また帳票はスタッカに送出され 1 枚の帳票の処理が終了する。

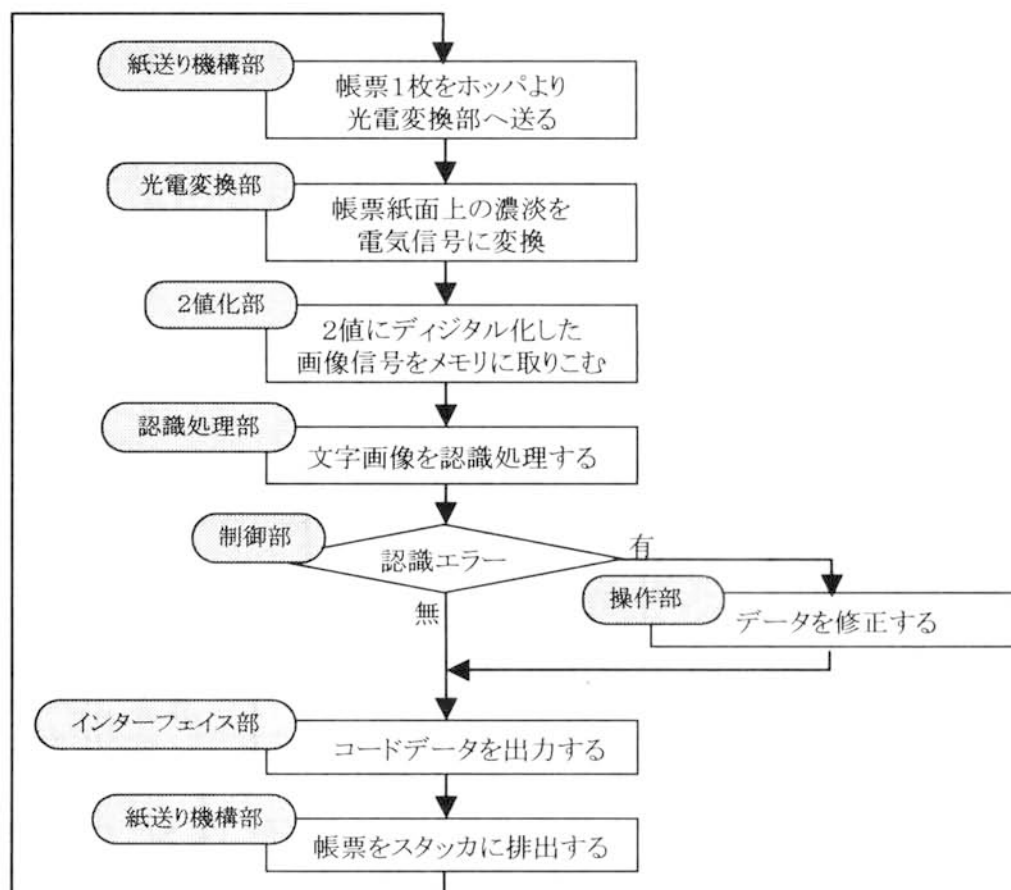
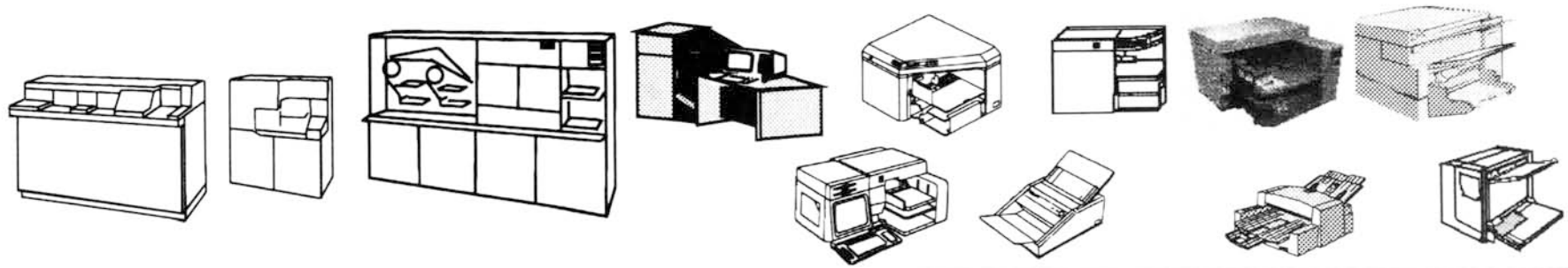


図 1.3.2 光学文字読取装置の動作

1.4 製品の経緯

図 1.4.1 に、筆者が研究開発に関与した光学文字読取装置を中心とした製品の経緯を示す。また、ここには装置各部の構成・動作方式の変遷、読取り字種の拡大、および文字認識方式の発展経緯の代表的な時期を表す。



年度		1970	75	80	85	90	95	2000
型式		H-8252	H-8257	T-550/30	T-550/60, 80 T-550/40, 50 T-550/47, 27	HT-4175, 76 5050/5060	HT-4196, 97 6050, 6070	HT-4194, 98 HT-4132 HT-4133 HT-4137 HT-4138
構成	制御部		専用プロセッサ (パイプラインハードウェア)			ワークステーション パソコン		
	認識部							
	前処理部							
	2値化部	ハードワイヤード論理						
動作方式			行走査型		全面走査型			
読取文字種	漢字		ドキュメントリーダ型					
	カタカナ							
	英字							
	数字							
	印刷文字							
認識方式	手書文字			非線型マッチング法	図形検証法	多重仮説検証法		複合仮説検証法
	印刷文字			可変閾値法				

ストロークアナリシス法
4象限特徴抽出法

図 1.4.1 製品の経緯

国産初のH-8252ドキュメントリーダーから多くの製品が市場に供された。この間、認識方式、ハードウェアおよびソフトウェアの進歩は著しく、装置の性能は飛躍的に増大した。また、装置の小型化および価格の低下は著しいものがあった。

読取り文字は印刷文字のみから、手書文字に、また字種も数字から英字、カタカナへと拡大し、現在では漢字にまで対象を広げて来た。これには、認識方式の大きな進歩の研究成果が生かされている。

これに対応して、アーキテクチャの構成も、当初はすべてハードウェア（ハードワイヤード論理）であったが、全体制御を行なう制御部および手書文字認識処理を行なう認識部が、ハードウェアに命令を強化された専用プロセッサに発展した。また、高速の装置ではプロセッサとは独立して並行動作を行なうパイプライン制御の高速認識論理が併用された。この専用プロセッサは、後述する前処理の役割の減少に伴い前処理を行なうまで処理範囲を広げた。

ワークステーションおよびパソコンが発達、低価格化するに従い、制御部から順に使用され、現在では認識処理を担っている。

ただし、多値の画像信号を2値化する2値化部では、計算処理量が多大なため、現在でもハードウェアとして残されている。

1.5 ソフトウェア化の動向

当初、全てがハードウェアで構成されていた光学文字読取装置も、素子の高性能化および

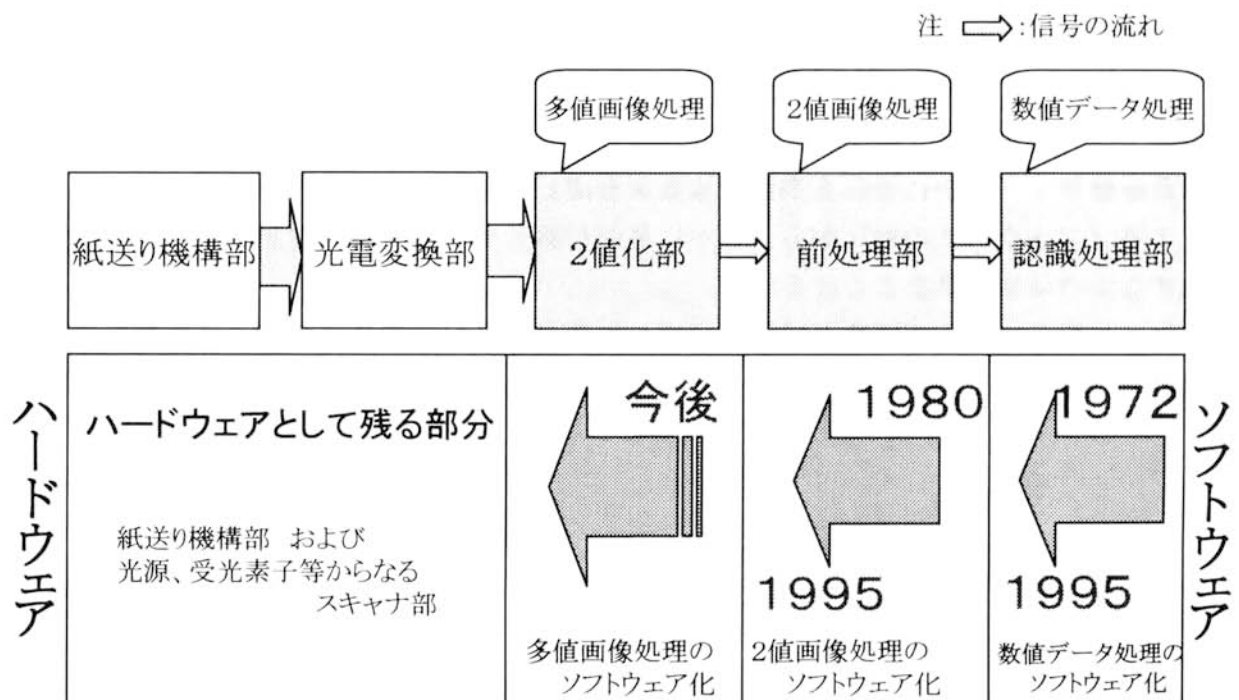


図 1.5.1 ソフトウェア化の動向

び文字認識方式の研究成果実用化のため、ハードウェアからソフトウェアへの転換が進められてきた。この転換は、認識処理が画像ではなく数値データの処理である部分からソフトウェア化され、次第に画像処理に進み、近い将来すべての画像処理に及ぶと予想される。

図 1.5.1 に動向を示す。

すなわち、印刷文字しか読み取らなかった初期の光学文字読取装置では、全てがハードウェアで構成されていたが、ICが登場して小型コンピュータによるソフトウェア処理が利用できるようになったこと、および変形の多い手書文字読取りには複雑な処理が可能なソフトウェアが必須であったことから、認識処理部のソフトウェア化が進んできた。

さらに、マイクロコンピュータの登場および発達、メモリの低価格化により 2 値画像の画像処理にもソフトウェア化が進み、前処理部も現在ではソフトウェア化されている。これにより、ハードウェアにより単純な処理しかできなかった文字切出しも複雑な処理が可能となり、利用しやすい光学文字読取装置¹²⁾になった。

今後、さらにマイクロコンピュータの性能向上に従い、多値画像の画像処理が高速で可能になるため、残された 2 値化部もソフトウェア化されると予想される。

1.6 本論文の位置付け

上述のように、大きな進歩をした文字認識については、多くの研究が発表されているが、そのほとんどは認識アルゴリズムの研究である。この研究を生かした製品が光学文字読取装置である。光学文字読取装置は、認識アルゴリズムを利用し実際の製品とするために、進歩の著しい多くの先端技術および先端部品を利用したアーキテクチャを採用して初めて役に立つ装置として成り立つ。このために、多くの努力が費やされているが、研究成果として発表されることは少ない。

本論文は、発展する文字認識アルゴリズムの研究成果を、日進月歩の先端技術および先端部品を使用し、いかに光学文字読取装置を構成し実現するか、特にその中でも、ハードウェアとソフトウェアの対比から、いかに最適な組合せにより有用な光学文字読取装置を構成するかの研究成果をまとめる。

また、本報告は、文字認識と同様な傾向の処理が必要な画像処理装置、自動認識機械にも貢献するものである。

第 2 章

アーキテクチャの要点と課題

第2章 アーキテクチャの要点と課題

本章では、光学文字読取装置に必要とされる装置の特質を明らかにし、これに主眼を置いたハードウェアとソフトウェアとの特質の対比を行なう。また、これらを踏まえて光学文字読取装置のアーキテクチャ構成の要点と課題を明らかにする。

2.1 光学文字読取装置の役割

光学文字読取装置のアーキテクチャの研究にあたり、光学文字読取装置がコンピュータの入力機器として必要とされる役割を明らかにする。

表 2.1.1 光学文字読取装置の役割

必要とされる役割	内容	評価項目	決定要素	補完手法
入力データの信頼性	文字を誤り無く、正しく入力すること	文字認識精度	文字認識方式	チェック機能 修正機能
	全ての帳票を、漏れなく入力すること	紙送り信頼性		2枚送りチェック機能
入力コスト／パフォーマンス	他の入力手段に対し、経済的に有利であること	処理速度／コスト	アーキテクチャ構成	
		ランニングコスト		
制限緩和	必要な項目が入力できること	認識対象の拡大	文字認識方式	

2.1.1 入力データの信頼性

コンピュータの入力機械として、第一に必要な機能は、入力すべきデータを漏れなく、誤り無く入力することである。光学文字読取装置の場合には、入力すべきデータは帳票に記入された文字であり、これを誤り無く認識しコードに変換することである。合わせて、複数枚の帳票を漏れなく読取ることが重要である。

● 文字認識精度

紙面上の文字は、乱雑な記入、汚れなど多くの要因により形状、濃度など認識に十分な文字品質が保たれていないことがある。このように劣化した文字品質では、認識処理において、“人”と同様に読み誤り（誤読文字）および読取り不能（不読文字）が生じる。特に、入力装置としての光学文字読取装置として重要なことは誤読文字を無くすことである。この課題に対しては、最も認識精度の良い文字認識方式の採用がまず必要となる。

文字の認識精度は、不読率および誤読率で評価される。

$$\text{不読率} = \frac{\text{不読文字数}}{\text{読取り文字数}}$$

$$\text{誤読率} = \frac{\text{誤読文字}}{\text{読取り文字数}}$$

入力データとしては、コードの内容から検出可能な不読よりもコードからは検出できない誤読のゼロ化が重要である。しかし、文字認識においては、誤読と不読は二律背反の関係にあり、誤読をゼロあるいは最小にすることが出来ないことが多い。このため、実際にはサンプル集合内で不読／誤読のバランスを設定して改良作業を行なうことが行なわれる。

例： 手書数字 誤読率：不読率＝1：10

- 補完手段により誤読、不読文字の検出修正

上記のように、現実の光学文字読取装置では誤読をゼロにすることができない。このため、光学文字読取装置では通常、誤読文字データの入力を防ぐため、補完手段として多くの機能を備えている。主なものは、データチェック機能およびオペレータによる文字データの訂正が可能な修正手段である。

データチェック機能：サムチェック、ディジットチェックほか

修正手段：即時修正、バッチ修正ほか

- 紙送り信頼性

上記文字認識精度以上に信頼性として要求されるのは、帳票を失わないことである。光学文字読取装置の紙送り機構は機種により異なるが、数十枚から千枚程度自動的に読み取る機構となっている。この時、紙送り機構が1枚ずつ分離して送ることにより、全部の帳票を確実に読み取ることが期待される。

しかしながら、帳票は、人の手を介してターンアラウンドして後、光学文字読取装置に入力されるため、温度、湿度、折れ、汚れ、ホッチキス、糊付けなどの要因により確実に1枚ずつ分離できないエラーが発生する。また、オペレータが紙送り機構に複数枚の帳票をセットする際、裏返しあるいは逆向きなどのヒューマンエラーを起こすこともある。

これらの場合、読み取る枚数が抜ける現象が発生することがある。この場合抜けた帳票の全文字がコンピュータに入力されない、すなわち帳票1枚分の大量のデータが入力されないという、誤読以上の重大なエラー（データ入力もれ、蒸発とも言われる）が発生する。

このため、光学文字読取装置では、帳票の抜け防止あるいは検出に多大の努力が払われている。

例：2枚送りチェック機能

2.1.2 入力コストパフォーマンス

実用に供される光学文字読取装置は、他の入力装置との競合にさらされる。ここで評価されるのは、入力作業のコストパフォーマンスである。光学文字読取装置におけるコストパフォーマンスは主に、処理速度／コスト（装置の価格）、および光学文字読取装置を業務に使用することにより増大する紙および印刷費用などのランニングコストである。

- 処理速度／コスト

光学文字読取装置は、文字認識能力を最大限にあげるため、最も認識精度の良い認識方式を採用する。価格と認識精度は必ずしも並行するものではないが、認識精

度の高い方式は複雑で認識論理部の価格も高くなり勝ちであることは事実である。このため、コストパフォーマンスの向上のために速度性能を向上させる必要がある。

従って、処理速度は光学文字読取装置にとって非常に重要な項目となる。この処理速度は、光学文字読取装置のアーキテクチャに大きく左右され、またコストに与える影響が大きい。特に、アーキテクチャ決定に当り本論文の主目的であるハードウェアとソフトウェアの最適構成が重要なポイントとなる。

● ランニングコスト

入力データの信頼性のため、光学文字読取装置を利用する業務では、制限条件を加え実業務で利用されている。ただし、これらの制限条件は、ランニングコストの増加および利用範囲の制限となっており光学文字読取装置の普及を妨げる要因ともなっている。

本論文の趣旨と離れるため本点の詳述は避けるが、制限条件を列挙すると以下のものがある。

- ▶ 印刷文字の字形、印刷条件（使用プリンタなど）の制限
=>高価な装置、インクリボン
- ▶ 手書文字の記入欄の形状（柵目など）の制限
=>帳票大型化
- ▶ 帳票紙質（厚さ、反射率、など）の制限
=>高価な OCR 用紙

この中で帳票の大型化、高価な紙質が必要なことから、特に帳票のコストにかかわるランニングコストの上昇がある。このため、光学文字読取装置においては、制限を緩和しランニングコストを下げる種々の努力がなされている。紙送り機構では当初の70Kg^(*)OCR紙のみから、35Kg薄紙普通紙を可能にするなどランニングコスト低減に貢献する技術進歩が図られている。

2.1.3 制限緩和

コンピュータへ入力するデータは、初期には数字データが90%以上占めていたが、現在ではカタカナから漢字まで、あらゆる文字種データの入力が必要とされている。また、実際の紙上の文字記入欄もサイズ、形状、色彩など多種多様である。しかしながら、文字認識技術では非常に限られた文字種および文字記入欄でしか対応できない。このため、認識文字種の拡大、記入欄の多様化など光学文字読取装置の制限を緩和する研究⁶⁾も多くなされた。これにより、現在の光学文字読取装置では、認識対象文字種としては漢字、文字記入欄としては、表形式などが実用化¹²⁾されている。

[まとめ]

光学文字読取装置は、データ入力装置として、“人”の主要な情報媒体である紙を入力

(*) 印刷業界では、紙の厚さを四六判サイズ(788mm×1091mm)1,000枚の重さ(連量)で表す。一般に紙が厚いほど高価である。

媒体として、記入された文字をコンピュータに入力する装置である。このため、

- (1) 入力すべき帳票上の文字を漏れなく、正しいコードに変換し（高精度認識）
- (2) 入力コストパフォーマンス向上のため、高速に入力するための処理速度を具備することが必要とされる。

2.2 光学文字読取装置のアーキテクチャの特質

光学文字読取装置に要請される役割に観点をおいて、装置アーキテクチャに必要な特質を明らかにする。

表 2.2.1 アーキテクチャに必要な特質

必要な特質	内 容	目 的	備 考
改善可能性 (柔軟性)	装置ハードウェア完成後も、認識能力向上のため改良改善ができる構成	文字認識精度の改善可能性の保持	最新の認識研究成果および実業務で得た内容の適用
高速度処理	可能なコストの範囲で最も高い処理速度が得られる構成	装置コストパフォーマンスの向上	計算アルゴリズムの開発と、ハードウェアとソフトウェアの最適化構成
低コスト	可能な仕様を実現する最も低いコストの構成		

2.2.1 改善可能性（柔軟性）

光学文字読取装置に関わらず自動認識機械の認識能力は、常に人の認識能力と比較される。しかしながら、自動認識機械の認識能力は、多くの研究成果にもかかわらず、人の認識能力に比べまだ不十分であり、これらを業務に利用する側から認識能力向上の要求が強い。

このため、自動認識機械は通常の装置と異なり、装置として完成後も改良が可能な、あるいは認識能力向上をはかることが可能な構成である必要がある。光学文字読取装置においては、文字認識精度向上のための改善可能性を考慮したアーキテクチャでなければならない。

● 必要性

➤ 最新の研究成果の適用

2.1 節で述べたように、光学文字読取装置に要請される役割では第一に最新の研究成果を取り入れ、最高の認識能力を具備することが求められる。装置の開発には1～3年という長期間を要する。この間装置が採用した文字認識方式も改良研究が進み、この成果を取り入れることが必要となる。

➤ 開発評価の実施

3.1 節で述べるが、大量サンプルによる認識精度の評価は、開発時、ハードウェア完成後に実装置を使用して行なわれる。これは、認識方式の研究はシミュレーションで行なうのが一般的であるが、シミュレーションでは、利用するコンピ

ユータの制約から大量の文字サンプルでの評価が難しいことが多いこと、および装置固有の光電変換部との整合性は実装置で評価することが必要であることなどのためである。この評価改良内容は装置に適用されなければならない。

➤ 業務の改善

さらに実業務においては、できるだけ業務の効率を向上させるため、実稼動開始後の改良も必要である。すなわち、光学文字読取装置では、利用者から誤読（認識誤り）あるいは特定の字形に対する認識精度改良要求を受け、装置の改良が行なわれることがある。これは、開発段階での大量サンプルの評価においても想定し得なかった、実業務における他のサンプルに対する認識能力の不十分さの現われであり、稼動後も改善を要することがある。

図 2.2.1 は、評価改良および光学文字読取装置出荷後の文字認識不良率の推移例を示す概念図である。このように、開発段階での改良、装置出荷後の改良により認識精度の向上が図られている。

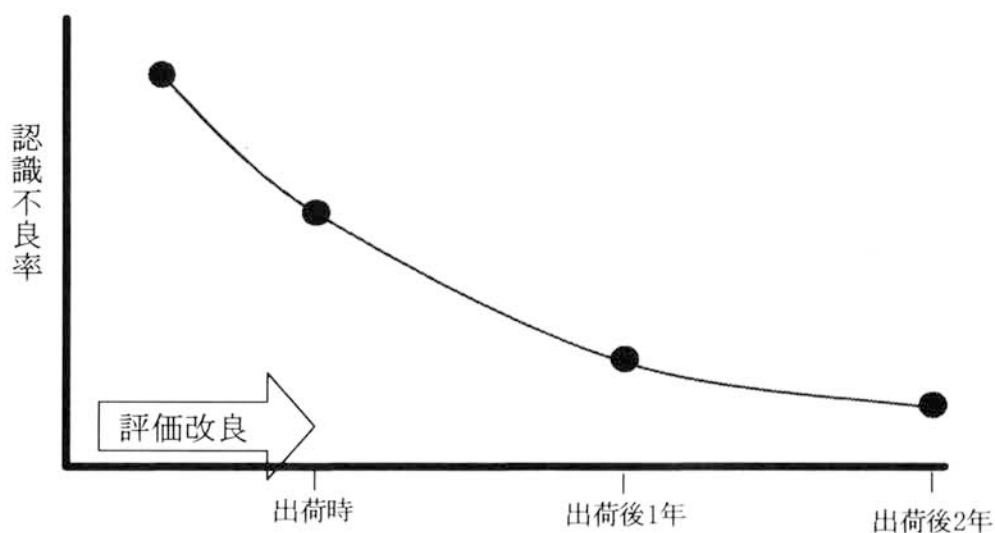


図 2.2.1 認識精度の改良

このように、装置ハードウェア完成後さらには実稼動中も改良改善が必要となるため、光学文字読取装置のアーキテクチャは、開発段階から後の文字認識能力向上が可能な構成をとっておく必要がある。従って、これらの改良は、装置を変更せず、具体的にはハードウェアを変更せず可能となるよう当初よりアーキテクチャには考慮が必要である。

2.2.2 高速度処理

2.1 節で述べたように、光学文字読取装置では処理速度が重要な項目である。処理速度は、毎分何枚帳票を入力できるかで評価されるため、実際の業務に当り、はっきりと性能差が数値的に求められる。すなわち主観的な認識能力に比べ、客観的な効果が計算できる

ため、最大のパフォーマンスとして評価される。

また、特に注意すべきことは、上記 2.2.1 項の改良によって速度が低下する可能性がある。改良によっても少なくとも仕様速度を守らなければならないため、速度に対してできるだけ余裕を持ったアーキテクチャが必要となる。

2.2.3 低コスト

コストは、開発費を除けば、ハードウェアによって左右される。

ハードウェアは、図 1.3.1 に示したように紙送り機構、光電変換部、認識部、および制御部から構成される。

このため、光学文字読取装置構成上、採用する動作方式、文字認識方式、持つべき諸機能を前提に、コストと速度を最適化することが装置アーキテクチャの研究開発上の最大の目的となる。

[まとめ]

光学文字読取装置のアーキテクチャには

- (1) ハードウェア完成後も文字認識精度向上のため、文字認識処理の改良が可能なアーキテクチャが必要であり、
- (2) 処理速度をパフォーマンスとし、良好なコストパフォーマンスを得られるアーキテクチャの実現が必要となる。

2.3 ハードウェアとソフトウェアとの対比

本節では、特に文字認識処理および光学文字読取装置として重要な 2.2 節の内容を考慮してハードウェアとソフトウェアとの対比を行なう。以下では、文字認識処理を対象としてハードウェアとソフトウェアとについて対比し、表 2.3.1 に特質をまとめる。なお、対比は第一次的な特徴の対比であり、アレンジにより特質は変化する。

2.3.1 処理の特徴

- ▶ ハードウェア処理は、並列処理が可能であり、ハードウェアの物量（例えばゲート数、IC チップ数など）を増す構成により、複数のデータおよび動作を同時に処理することができる。このため目的の速度を達成する事が可能となる。
- ▶ ソフトウェア処理は、プログラムによる逐次処理であるため、処理速度はその処理を行なうプロセッサのサイクルタイムとダイナミックステップ数による。すなわち、処理時間は、このダイナミックステップ数を数える事により評価できる。

ソフトウェアに高速処理の実現のためには、このダイナミックステップ数を減らす計算方法が研究される。一方、判定分岐を有効に利用して速度を速めることが可能であるが、処理時間が入力データの内容によりばらつくことになる。実際、入力データによっては平均的なデータより 10 倍以上の時間がかかることも少なくない。このため、認識時間に制限の有る動作方式では、十分な注意が必要

表 2.3.1 ハードウェアとソフトウェアとの特質対比

		ハードウェア	ソフトウェア
処理の特徴	処理動作	並列／並行処理	逐次処理
	速度決定要素	ハードウェア量	ダイナミックステップ数
	処理時間単位	クロック	サイクル
	処理時間	固定的に定まる	平均時間で評価
	処理時間のばらつき	小	大
処理の適性	処理内容	一律的処理	判定分岐処理
	処理単位	ビット～	Byte、word
	処理動作	シフト／論理演算	演算／判定分岐
	データ処理量	大	小
	処理時間制限	有り（固定）	無し（可変） ^(注 2)
	画像処理	○	×
	数値／判定処理	×	○
アーキテクチャへの要請	改善可能性（柔軟性）	×	○
	高速性	○	×
	コストへの影響 ^(注 1)	大	少

(注 1)開発費などの一時費用は除き、装置 1 台のコスト（ハードウェアコスト）を対象とする。

(注 2)処理するデータ内容により変化する。

である。

検討段階では、代表的なモデルパターンによりダイナミックステップ数を評価するが、開発評価時には大量データによる検証が必要である。

- ▶ 処理時間の単位は、ソフトウェアでは、1 命令の実行のサイクルとなるが、ハードウェアにおいては高速のクロックとなることが多い。
- ▶ ハードウェアではデータによる判定分岐を行なわないことが多いため、処理時間は固定的に定まりデータによる処理時間のばらつきはない(あるいは少ない)が、ソフトウェアではデータの内容により処理を変えるため、ばらつきが生じる。このため、処理時間を平均時間で評価するが、時間制限のある処理には注意が必要となる。

2.3.2 処理の適性

- ▶ ハードウェアは、複数のデータの同時並行処理が可能であるから、一律で大量の処理に向いている。特に、データ量の多い画像処理には最適で、ビット単位のシフト処理によるマスク処理には最適となる。

しかしながら、ハードウェアでは、1 スキャンのビット数や計算される数値の桁数などが固定される欠点があり柔軟性には欠ける。

- ▶ ソフトウェアは、判定分岐が必要な処理には不可欠である。処理単位はビット単位では効率が悪く、バイトやワード単位での数値演算となる。また、データ内容で分岐することにより最適な処理を行なうことができるが、処理時間はばらつく。

2.3.3 アーキテクチャへの要請項目

2.2 節で明らかにした光学文字読取装置のアーキテクチャの要請項目について対比する。

① 改善可能性（柔軟性）

- ハードウェアにおいては、装置ハードウェア完成後の改良は、装置を構成する部品等の改良（手直し、交換等）が必要になる。このため、改良には改良部品の用意とともに、改良作業に専門のハード作業要員が必要となり、作業要員の確保・派遣、確認要員派遣など時間・コストがかかる。
- ソフトウェアにおいては、装置のプログラム入れ替えおよび確認ですむため、ハードウェアより作業が軽い。最近では、リモートメンテナンス処理が普及しており、実際に作業員が装置まで行かないで改良が実施されるなど、さらに改良のコスト・時間は低減されている。

② 高速性／コスト

- ハードウェアにおいては、並列および併行処理が可能であり、一般的に必要な速度が得られる。ただし、速度はハード物量に関係し、コストおよび体積（装置の大きさ）、装置の消費電力に大きな影響を与えるため、実際には、それらを考慮して選択適用される。
- ソフトウェアにおいては、逐次処理であるため、プログラムステップ数（＝メモリ容量）に関係し、ハードウェアよりコストおよび体積（装置の大きさ）、電力等への影響は少ない。しかし、速度に与える影響は大きい。
- プログラム処理において、速度はプロセッサの能力（サイクルタイム等）と処理のダイナミックステップ数で決まる。ダイナミックステップ数を減少させて速度を向上させる方法は数多くあるが、この場合、通常メモリが膨大になることが多く、ソフトウェアにおいても速度とコストの最適化が要求されるので注意が必要である。
- このため、ソフトウェア処理においては、プロセッサの能力を出来る限り高いものにし、かつメモリ容量が十分大きいものが要求される。ソフトウェアの装置コストは、プロセッサおよびメモリ量で概略決定する。
- なお、プロセッサを複数化し、併行処理をさせることにより、速度を速める手法²⁹⁾がある。この場合、処理はソフトウェアで実行されるものの、プロセッサというハードウェアを複数化することにより速度が決定されるため、本論文ではハードウェアでの対策に分類する。

以下では、ハードウェアとソフトウェアとの特質の対比を明確にするため、相関値計算処理につき速度および特徴の具体例を示す。

なお、報告に当り、プロセッサとしては、以下を前提とする。

- 1word 16 bit、
- 実行命令は加減算、AND、1ビットシフト、条件分岐（ゼロ、シフトオーバーフロー）命令程度、

- 各ステップは1サイクルで実行するシンプルなものとする。
- また、ダイナミックステップ数の評価は以下のモデルパターンで行なう。
- 未知パターン **X** と標準パターン **P** は、図 2.3.3 に示す縦 32 × 横 20 の 2 値画像
 - 相関値計算は、未知パターン **X** と標準パターン **P** との重ね合せ処理後、一致した画素をカウントする。

(a) すべての処理をプログラムで行なう

図 2.3.1(a) にプログラムで相関計算を行なう場合のフローチャートを示す。1 画素の抽出をシフト命令で行ない、このシフト命令で発生したオーバーフロービットで“1”か“0”かの判定を行ない、“1”の時に相関値レジスタ **S** に“1”を加算する。1 画素処理の毎に 4 ~ 5 ステップ必要となるため、1 カテゴリの相関計算のためのダイナミックステップ数は、式(2.3.1)となる。

$$\text{ダイナミックステップ数} = 4W + 5B \quad (2.3.1)$$

ここで、**W** は白画素数、**B** は黒画素数で $(32 \times 20 - W)$ である。モデルパターンでの 1 カテゴリのダイナミックステップ数は、約 3,000 ステップで、この内 92% (2,664 ステップ) が画素の処理である。これは、1 ビットの処理が不得意なこと、ループ終了判定に 2 ステップ要することなどの影響で、ソフトウェアで画像処理を行なう場合に共通する特徴的な現象である。

(b) 計算アルゴリズムの改良

計算アルゴリズムの改良を行ないソフトウェアで最大限の速度を得る例を図 2.3.1(b) に示す。改良のポイントは、(a) で 92% を占める画素の処理の短縮であり、ここでは 2 つの削減手法を実施している。

(i) 繰返し処理内ステップ数の削減

1 画素処理のダイナミックステップ数を 3 ステップに削減する計算法に改良する。これは、画像処理において 1 画素の処理のステップ数を減らすことが速度向上に多大に寄与するためである。(a) の例においては、1 ステップ削減することにより、640 ダイナミックステップ数 = 22% の削減効果がある。

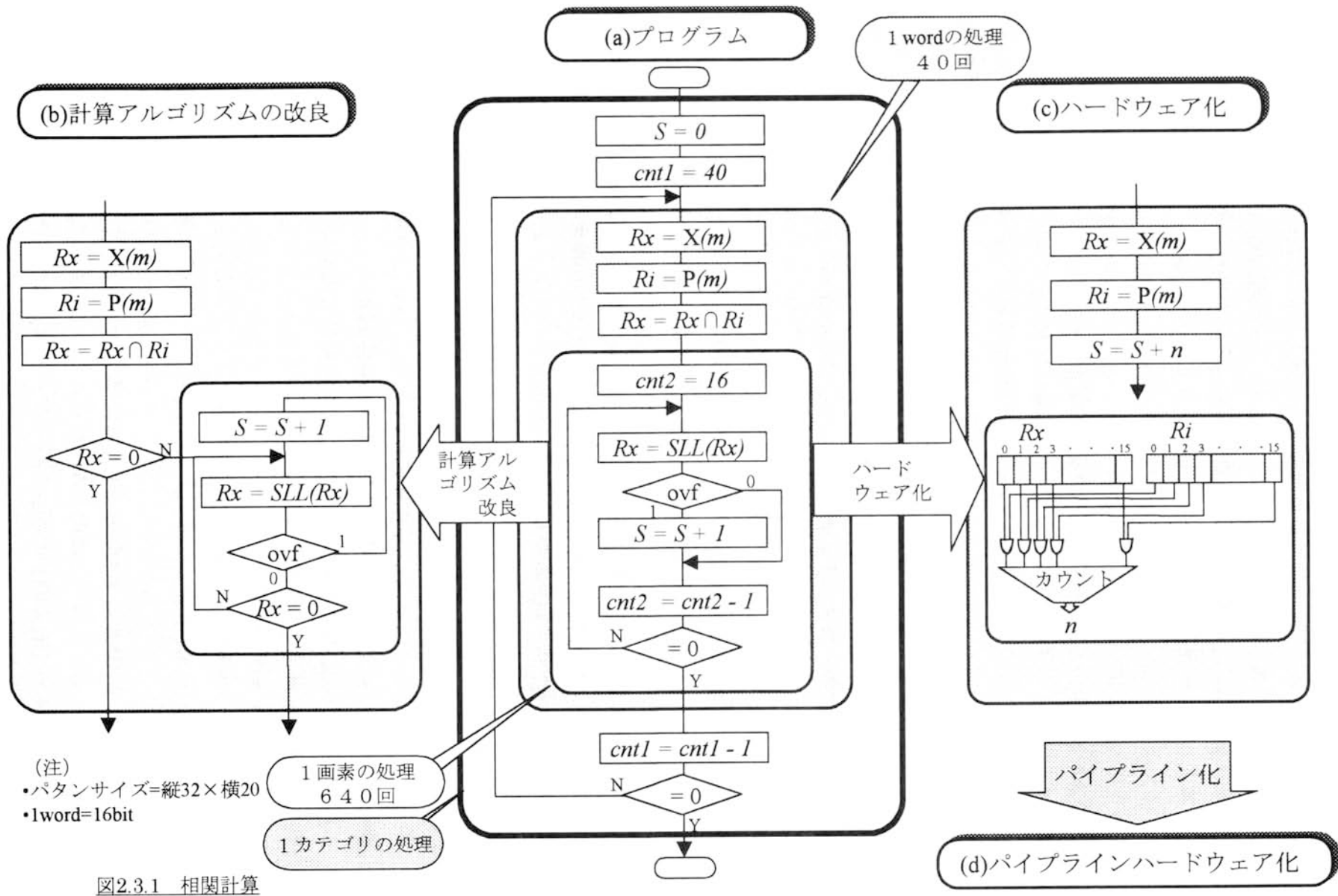
具体的な方法は、図 2.3.1(b) に示すように命令の順を変更し、必然的に必要な分岐命令を有効に配置することなどである。

(ii) 繰返し処理回数の削減

(a) において **lword** のループを **cnt2** で制御しているのに対し、(b) では、**Rx** の内容が空すなわち = 0 と成ることに着目し、**Rx = 0** になった以降は画素処理は不要であることを利用する。

これにより、AND 後の **Rx** が空の場合には残りループをしない。すなわち、**lword** すべて白画素の場合には、1 画素処理を行なわないで済み、また、**lword** 内の右端部の白画素に対しても 1 画素処理を行なわないで済むため、ダイナミックステップ数が大幅に削減される。

この 2 つの手法の適用により、モデルパターンに対し、約 500 ステップとなった。



この内51%（252ステップ）が画素の処理であり、(a)に対し、約1/6のダイナミックステップ数となり、また、1画素処理の比率も低減している。

このように、数ステップのループから、骨を削るがごとき努力により、1ステップでも削ることが非常に効果を発揮することが特徴である。

注意すべきことは、ここで採用される計算アルゴリズムはトリッキーな方法も含むため、プログラミングで避けなければならない永久ループの危険性を含むことである。しかしながら、1ステップといえども多大の効果が得られるので危険を犯す必要があり、通常的安全性を最優先とするプログラミングとは異なるのが画像処理ソフトウェアといえる。

また、(a)および(b)で特徴的なことは、画像データの内容すなわち、“1”と“0”の比率あるいは分布により、処理時間が異なることである。(b)を例にとると、全“白”パターンでは160ステップ、全“黒”パターンでは2,080ステップとなり、13倍となる。このため、処理速度見積に当っては、評価のためのモデルパタンの選定には注意が必要である。

また、処理時間に制限がある場合には最悪パターンでの計算をすることになる。一方、後述のハードウェア論理においては、このような画像データの内容に左右されることが少ないのが一般的である。

(c) ハードウェア追加による命令強化

(a)および(b)において、数ステップの1画素処理が全体の過半を占めているのは、プロセッサにビット処理および繰返し処理に有効な命令が不足しているためである。この欠点を除くためには、プロセッサにビット処理およびループを強化した命令を追加し、画像処理機能を強化する必要がある。

このような命令を追加するためには、プロセッサの演算部に、対応するハードウェアを追加する。

図2.3.1(c)に相関計算で有効なハードウェア追加の例を示す。これは、(a)および(b)で50～90%を占める1画像処理を大幅に短縮するためのハードウェア例である。すなわち、1wordの“1”の数をカウントするハードウェア論理を追加し、1画素処理を一挙に短縮する手法である。

R_x および R_i レジスタの対応する各ビット毎にANDゲートを設け、その出力をカウントする論理を設ける。これにより、プログラムでの1画素毎の演算処理は不要となり、 R_x および R_i レジスタへのパターンデータの読出し、および相関値レジスタへのカウント値の加算処理の3ステップに削減される。

これにより、モデルパターンに対し、約200ステップとなった。(a)に対し、約1/15のダイナミックステップ数である。なお、この場合は(b)と異なり、画像データの内容によるばらつきがないので、最大処理時間の制限がある場合にも有利となる。

(d) パイプライン処理

処理時間を最も削減する方法が、パイプラインハードウェア化である。パイプライン化したときの動作を、図 2.3.2 に示す。(イ)は逐次処理（シリアル処理）で処理時間は、前処理、相関処理など各処理の合計時間となる。

これに対し、相関計算をパイプライン化した場合のタイムチャートを(ロ)に示す。相関計算は、相関論理部において前処理と独立して実行される。すなわち、第 1 文字の相関計算は第 1 文字の前処理が終了した時点から開始される。同時に、第 2 文字の前処理が実行される。相関論理部において第 1 文字の相関計算が終了すると、相関論理部は、第 2 文字の相関計算を実行する。また、同時に、第 3 文字の前処理が開始される。このように、1 つの文字の処理は逐次実行されるが、同時に次の文字の処理も実行されるため、平均的に処理時間は短縮されることになる。このため、この方式は、多数の文字を連続して認識する場合には、高速処理の効果が得られる。しかし、少数の文字の場合には、高速の効果は少ない。フィールド単位で認識することの多い光学文字読取装置においては、これは 1 フィールド内の文字が多いか少ないかに該当する。通常 1 フィールドの文字数は、10～30 文字であるためパイプライン処理は有効である。

パイプライン処理の場合は、プログラムのダイナミックステップ数は、必要な情報のイニシャライズと起動および終了割込み処理であり、非常に少なくなる。処理時間は、プログラムのダイナミックステップ数ではなく、パイプラインハードウェア

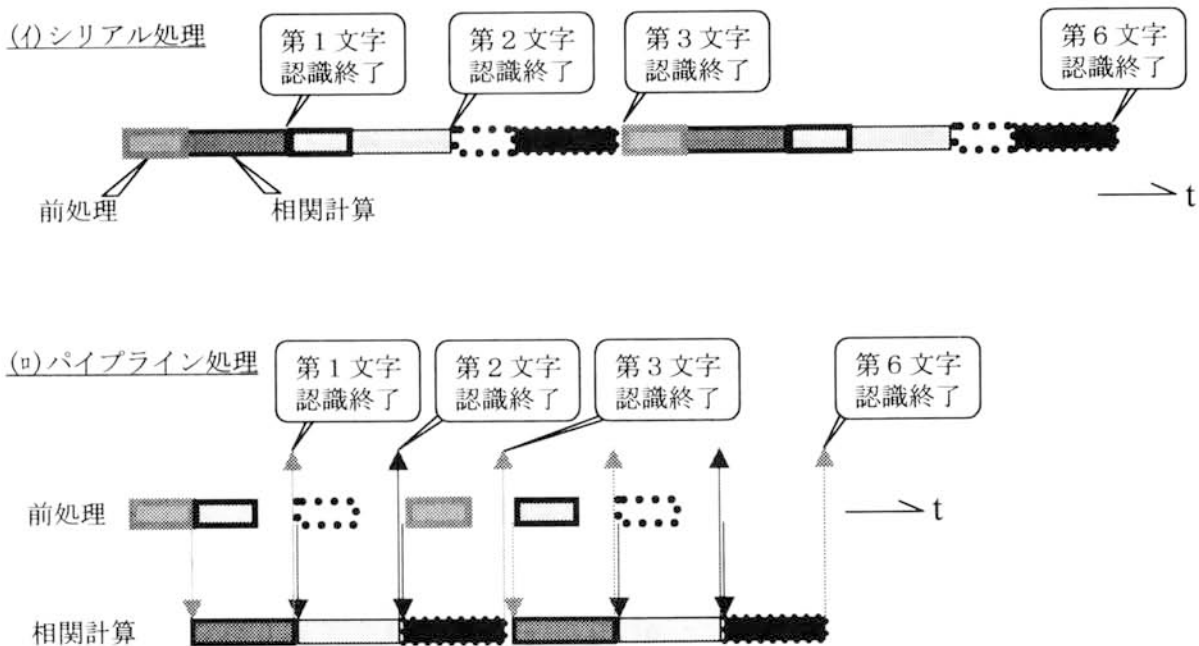


図 2.3.2 パイプライン処理のタイムチャート

アの計算時間で定まることとなり、処理時間短縮はハードウェア処理速度の向上という課題となる

なお、パイプライン動作のためには、独立して動作するための制御部が必要になる。これには、シーケンサが行なう。シーケンサは簡単なプログラム（ファームウェア）で動作する構成で、本論文ではハードウェアに含める。

図 2.3.3 に各ケースのダイナミックステップ数を示す。

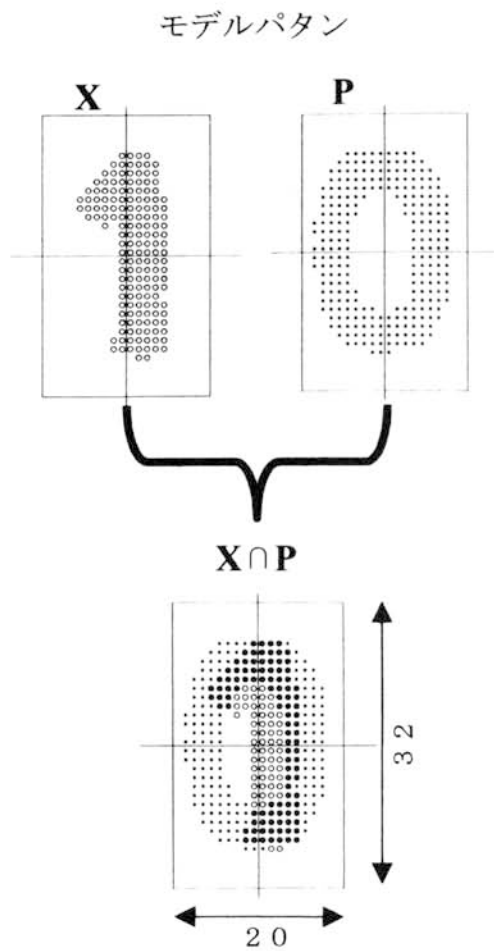
以上によりソフトウェア処理よりハードウェア処理が速度面では有利なこと、ソフトウェア処理では入力パターンにより大幅な時間のばらつきがあるが、ハードウェアではないこと、また計算アルゴリズムの改良は重要であることがわかる。

なお、本項では計算アルゴリズムはソフトウェアに有効な例を示したが、ハードウェア処理にも非常に重要である。

【まとめ】

ハードウェアとソフトウェアはその特質が異なる。

- (1) 装置ハードウェア完成後の認識能力向上に必要な改善可能性（柔軟性）においては、ソフトウェアが適している。
- (2) 速度面ではハードウェアが勝るが、コスト面では不利となる。このため、ハードウェア化においては、速度／コストの効果を評価する必要がある。



- パターンサイズ=縦32×横20
- 1 word = 16 bit

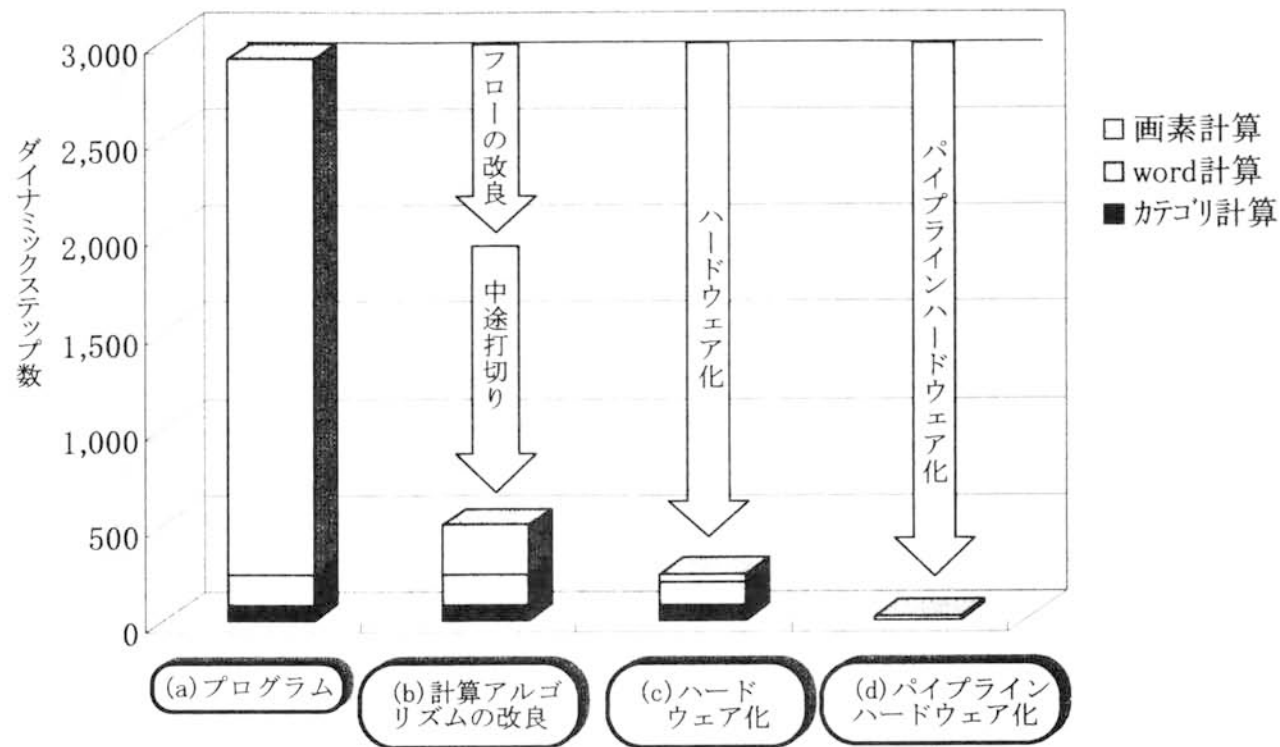


図2.3.3 関連計算ダイナミックステップ数

2.4 ハードウェアとソフトウェアとの最適化

本節では、処理速度向上のため、ハードウェアとソフトウェアとの最適化が有効な処理を各認識方式につき明らかにする。

2.4.1 文字認識方式とその特性

光学文字読取装置のアーキテクチャ開発は、採用される文字認識方式に大きな影響を受ける。以下では、各認識方式の特性をアーキテクチャ開発の観点から明らかにし、ハードウェアとソフトウェアとの最適化の該当箇所を明らかにする。

観点としては、ハードウェアとソフトウェアの最適化が速度向上の効果を発揮する処理はどのような特性を持つかに置く。

表 4.2.1 代表的な認識方式

		印刷文字	手書文字
文字の特徴	文字の大きさ	○ 一定	× 不定
	文字線の位置	○ 位置一定	× 位置不定
	線の繋がり	× 線欠け多い	○ 線繋がっている
	文字線の形状	× 線欠け多い	○ 形状に特徴（線に囲まれた白部分の形状）
文字認識手法		相関法 可変閾値法	構造解析法 四象限特徴抽出方式 非線型マッチング法 図形検証法
処理内容	前処理	<input type="checkbox"/> 画像処理	<input type="checkbox"/> 画像処理 <input type="checkbox"/> 2次元1次元変換
	特徴抽出	<input type="checkbox"/> 画像処理	☆ 数値計算・処理
	照合	<input type="checkbox"/> <input checked="" type="checkbox"/> 画像処理	☆ <input checked="" type="checkbox"/> 数値計算・処理
	判定	類似度判定	オートマトン／距離判定
標準パターン形式		固定長	可変長
標準パターン作成手法		統計的手法	個別作成

- (注) ○ : 認識に利用できる特徴
 × : 認識に利用できない特徴
 : 入力画像／データの内容に依存しない処理
 ☆ : 入力画像／データの内容により、計算方法が異なる処理
 : 繰返し処理／計算量が多い処理

文字認識手法としては、非常に多数の手法が提案されているが、大きく相関法と構造解析法に分かれる。また、認識対象文字である印刷文字あるいは手書文字（特に数字、英字、カタカナなど単純な字形）に分けられる。

これらは、プリンタなどの機械で印字される印刷文字と、“人”が筆記具により一本ずつ文字線を記入する手書文字との特徴の違いがあるためである。

印刷文字は、大きさが一定でかつ文字線の位置が同じである反面、各線の繋がりには保証されない。このため、文字線の繋がりではなく、文字線の位置が認識するための特徴となる。このことから、相関法が一般的に用いられる。なお、多種フォントの場合には大きさ、文字線の位置が異なる場合があるが、大きさの正規化を行ないサイズを統一するなどの前処理を加えている。

一方、手書文字は、人が記入するため、大きさ、文字線の位置は文字毎にばらばらであるが、文字線の形に特徴がある。このため、文字線あるいは文字線に囲まれた白地の形状を分析する手法である構造解析法が一般的に用いられる。

なお、手書き文字においても漢字の場合は、文字線の位置を粗く使用する方式⁵⁾もある。これは、漢字は文字線（画数）が多く数字やカタカナほど変化しないこと、構造解析法では照合に用いられる標準パターの作成を個別に行なうが、漢字ではカテゴリが数千種と多く人手対応ができないこと、などによるものと考えられる。

文字認識手法も、この特徴を利用するため、印刷文字と手書文字では異なる手法が用いられる。このため、光学文字読取装置においても、それぞれに対応した2通りのハードウェアまたはソフトウェアをもつことが必要とされる。

(1) 印刷文字認識

印刷文字認識手法の代表的な例として可変閾値法を説明する。

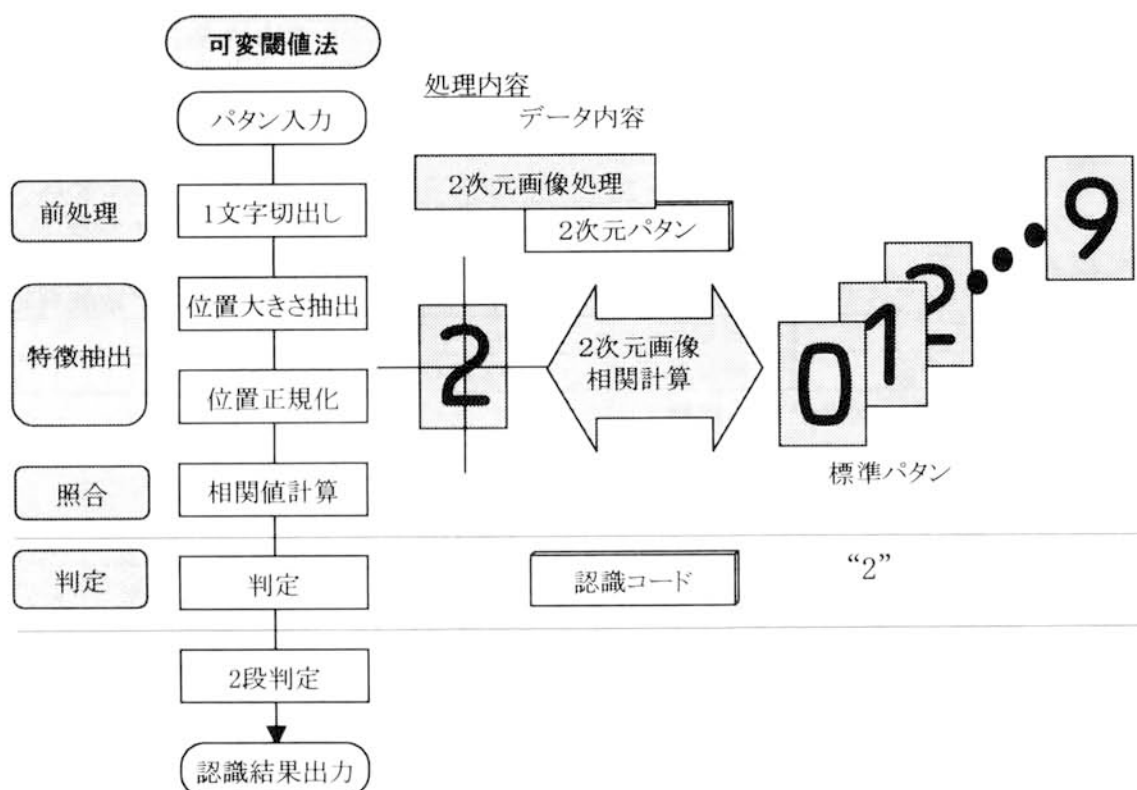


図 2.4.1 可変閾値法の認識フローチャート

➤ 可変閾値法

図 2.4.1 に可変閾値法の認識フローチャートに示すように、1 文字画像の入力から相関計算までの各処理の扱うデータはすべて 2 次元画像であり、各処理も 2 次元画像の投影、移動、相関計算という 2 次元画像処理である。このため、各画素に対応した繰返し処理が多いのが特徴である。したがって、ハードウェア処理による高速化が効果を発揮する認識方式である。

なお、2 段判定は対判定による多段認識方式⁴⁾を採用している。ただし、具体的な処理は、相関計算処理の組合せであるため、上記高速化に伴い効果が得られる。

(2) 手書文字

図 2.4.2 に示すように、構造解析法による文字認識方式は発展している。発展の段階で特徴的なことは前処理の役割が減少していることである。

ここで、前処理とは、入力画像、データに依存しない一律な処理または可換変換処理とする。すなわち、定まったマスクにより一律に 2 次元画像を変換する処理、および可換変換である 2 次元画像の 1 次元チェインコード化処理などを前処理とする。

なお、前処理部の役割が減っている理由は、プロセッサ、メモリが高価であった時代には、できるだけ後段（特徴抽出、照合）のデータ量を少なくし、プログラムおよ

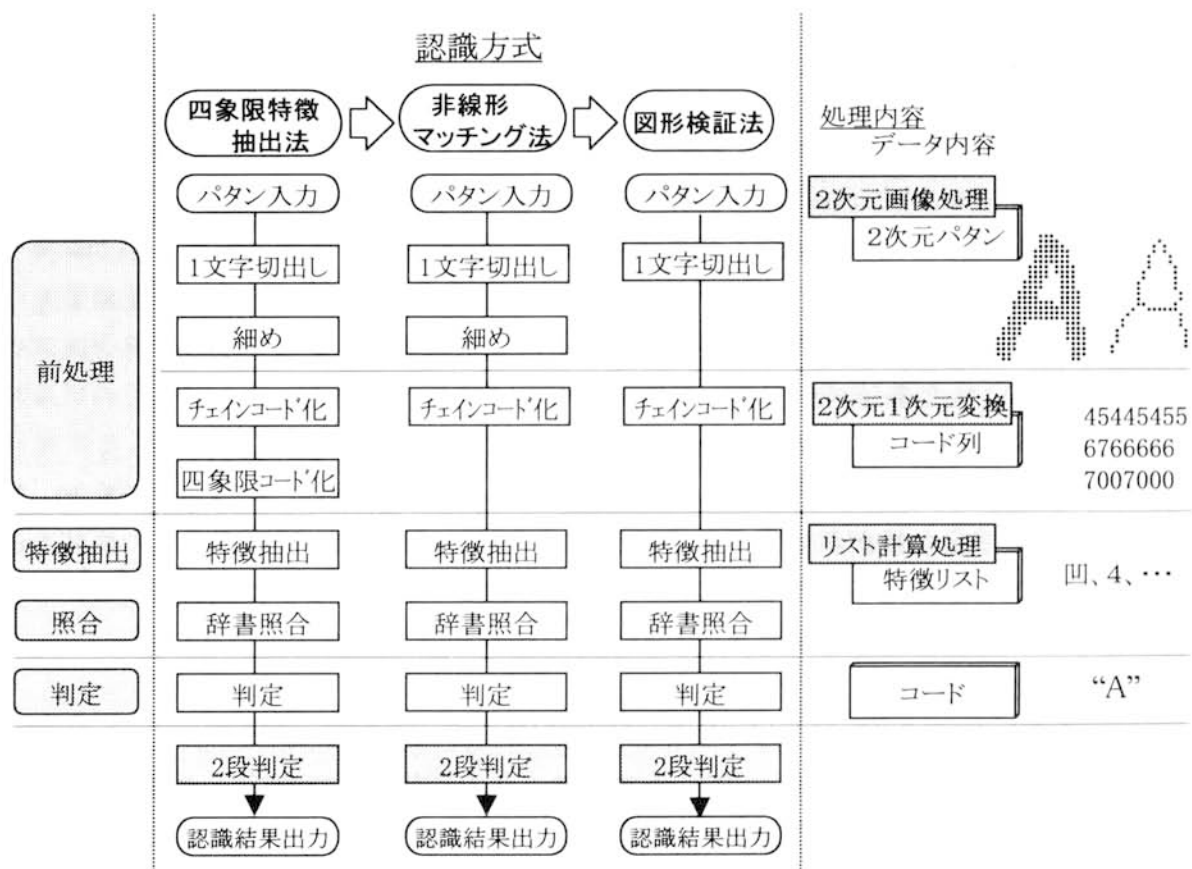


図 2.4.2 手書文字認識フローチャート

びメモリを最小にしてコスト節減を図る必要があったことによる。ただし、一律的な前処理による情報削減は、判定時に情報が不足することになり、認識精度に一定の限界を設定することになる。このため、プロセッサ、メモリが低価格化、高性能化するに従い、前処理による情報欠落をなくし判定時の特徴を増す方向に発展している。

各認識方式の概要を述べる。

◆ 四象限特徴抽出法³⁾

前処理では、細め処理後1次元チェインコードに変換し、さらに四象限コード列にまとめる。この四象限コード列からクラス分けを行ない、標準パターンとオートマトンによる照合を行ない一致したコードを出力する。

この認識方式の特徴は、2次元文字画像を前処理にて四象限コード列に変換することにより、情報量を大幅に削減できたため、特徴抽出および照合処理の計算量が少なくて済んだことである。これにより、手書数字を読取る小形で低価格な光学文字読取装置が実現された。

しかしながら、四象限コード列では、英文字の認識には特徴が不足し、適用できないといった限界があった。

◆ 非線型マッチング法¹¹⁾

四象限コード列への変換を行わず、1次元チェインコードを直接標準パターンとダイナミックプログラミングにより照合する。得られた距離により、判定出力する。

この方式は、四象限コード列への変換で欠落した文字線の特徴を利用するため、1次元チェインコードから直接特徴抽出および照合処理を行なった。このため、特徴抽出および照合処理の計算量は大幅に増大したが、手書英字およびカタカナの認識が可能となった。この方式により、手書英字およびカタカナを読取る光学文字読取装置が実現された。計算量が増大したため、第5章でのべる照合論理を開発し、大型高速の光学文字読取装置でまず実用化された。さらに、この照合論理はLSI化され、小形の光学文字読取装置にも適用された。

しかしながら、1次元チェインコードは一律の細め処理で作成されるため、詳細な文字線の特徴が欠落する。このため、乱雑に記入された手書文字の誤読を避けることが困難という限界があり、読取対象は丁寧に記入されることを前提とした文字に制限していた。

◆ 図形検証法

細め処理を行わず、2値化文字画像の文字線縁を1次元チェインコードに変換し、線分に分割、特徴量計算などを行ない、標準パターンと照合する。得られた距離により、判定出力する。

この方式により、細め処理のため丁寧な文字に限定していた認識対象文字を広げられたため、手書文字の記入制限が大幅に緩和された。すなわち、丁寧に記入

されていない手書文字についても、誤読を少なくすることができるようになった。

認識処理は、複雑増大化したがる、部品等の進歩により小形の装置でもソフトウェアが利用できるようになったため、実用化が可能となった。これにより、光学文字読取装置の利用業務が一般の伝票・契約申込書などに大幅に拡大された。

なお、2段判定は印刷文字認識と同様に、対判定による多段認識方式⁴⁾を採用している。ただし、具体的な処理は、それぞれの認識方式の特徴抽出で得た特徴を使用しており、対判定のため処理量は少ない。

構造解析法では可変閾値法と異なり、前処理において2次元画像を1次元チェインコードに変換した後は数値計算処理となる。特徴抽出処理は複雑な判定分岐があり、ハードウェアには不向きである。前処理の役割が減少しているため前処理のハードウェア化による処理速度向上の効果も減少している。ただし、照合処理では多数の標準パターンとの照合計算であり、繰返し処理が多くハードウェア化が高速化に効果がある。

以上のように、文字認識処理においては前段階ではハードウェア化で効果がある2次元画像処理、後段階ではソフトウェアに向いている数値演算、判定処理となる。また、各認識方式によりその切替段階が異なる。

また、いずれにおいても照合処理においては、標準パターンとの照合による繰返し処理が多いため、これらがハードウェアとソフトウェア化による高速化に効果がある処理となる。

2.4.2 最適化のアプローチ

光学文字読取装置において重要な速度仕様達成のためには、アーキテクチャの構成においてハードウェアとソフトウェアとの最適化が重要となる。速度仕様達成には前節に述べたように種々の方法がある。アーキテクチャ検討段階にてどのようにアプローチすべきかを述べる。

アプローチの概念を図2.4.3に示す。

- 光学文字読取装置の開発に当り、どのような仕様の装置とするかが決定される。これを受けて、経験から概略のアーキテクチャの方針がたてられる。すなわち、低速ながら安価な製品の開発であればソフトウェアでの構成を考え、速度が不足な場合はハードウェアにて速度達成を図る。また、高速を追求する製品の場合は、パイプライン構成を前提に検討するなどである。ただし、高速/低速あるいは低価格/高価格などの区別はその時点の技術レベルにより大きく左右される性質のものであり、本論文の趣旨から離れるため詳述は省くが、綿密な各種調査と経験により決定される。
- 低価格を追求する場合は、まずプログラムでの実現を図る。速度に最も影響する繰返し処理（1画素の処理および照合計算処理）を見積り、

この時間から全体の処理時間を評価する。不足の場合は、計算アルゴリズムの改良を行ない、ダイナミックステップ数削減をおこなう。この検討を繰り返し、速度向上を図る。さらに、速度向上のためにはハードウェア化を図る。すなわち、繰り返し処理にハードウェアを追加する。速度が不足の場合は、ハードウェアの範囲の拡大、並列処理の増強などの検討を繰り返す。

- 高速を目的とする光学文字読取装置の場合は、パイプライン構成を考える。どの処理をパイプライン化するかを検討する。パイプライン処理は、1つの場合もあるが、複数の処理を独立にパイプライン化する場合も多い。

パイプラインハードウェアの検討も、繰り返し処理の部分の構成を決定することにより、おおよその処理時間を得られる。これに加えて処理の範囲拡大、並列処理強化などにより速度向上をはかり、最後にパイプライン部の制御をおこなうシーケンサ部の検討行なう。

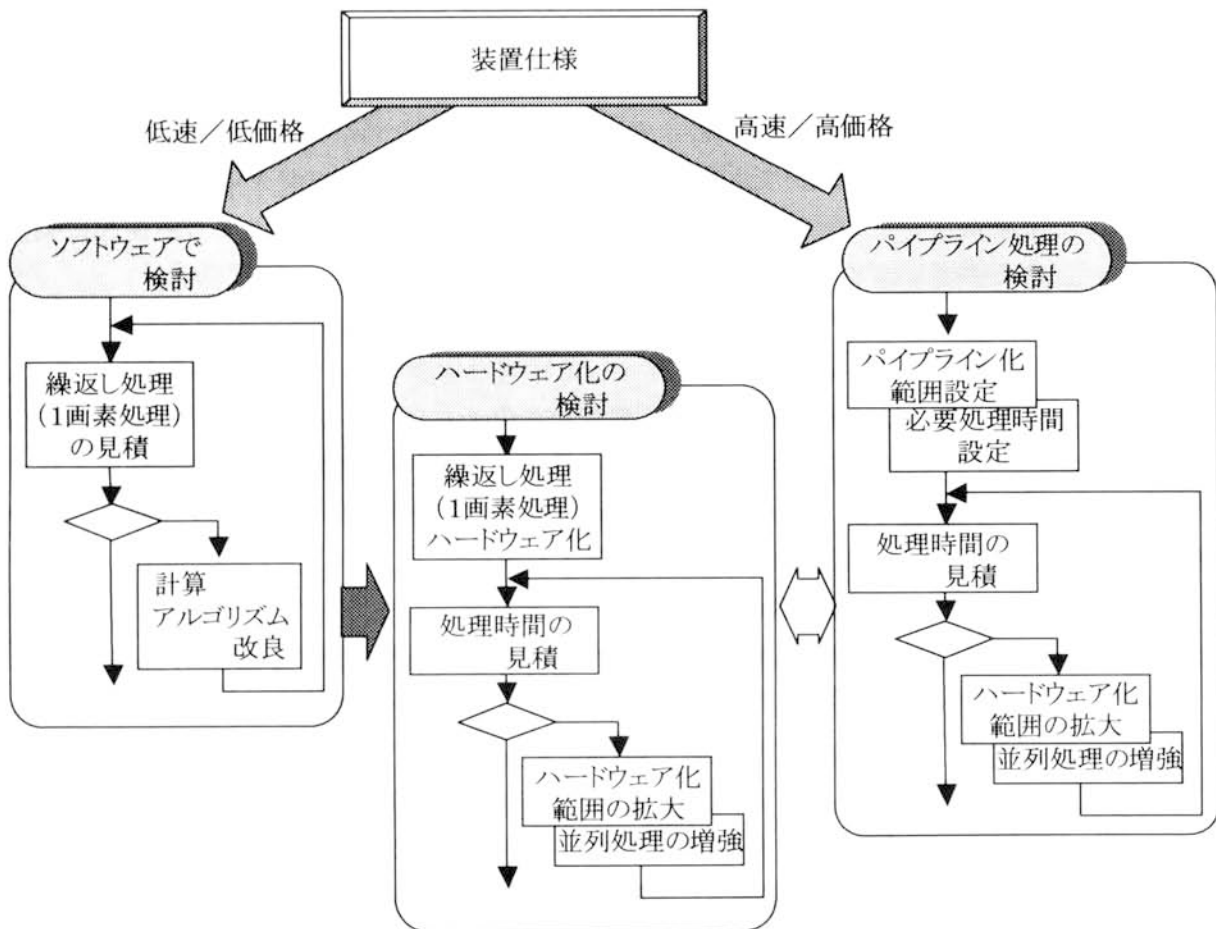


図 2.4.3 アプローチの概念

[まとめ]

複雑な判定分岐からなる処理はソフトウェアとし、必要な速度を得るために、

- (1) ソフトウェアでの速度見積により不足の場合、ハードウェア化する。ハードウェア化は、一律的な処理からなる画像処理および繰り返し処理の多い照合処理を検討する。
- (2) さらに高速化するには、ハードウェア範囲の拡大、一部複数化による並列計算、およびパイプライン化による並行処理により、速度とコストのバランスしたソフトウェアとハードウェアの最適化したアーキテクチャを求める。

2.5 第2章のむすび

図 2.5.1 に光学文字読取装置アーキテクチャの要点と課題を図示する。

光学文字読取装置は、文字データの入力のための自動認識機械として入力データの信頼性すなわち正しく、漏れなく文字を認識して入力することが重要な役割である。

合わせて、他の入力手段との比較から、速度とコストによるコストパフォーマンスの追求

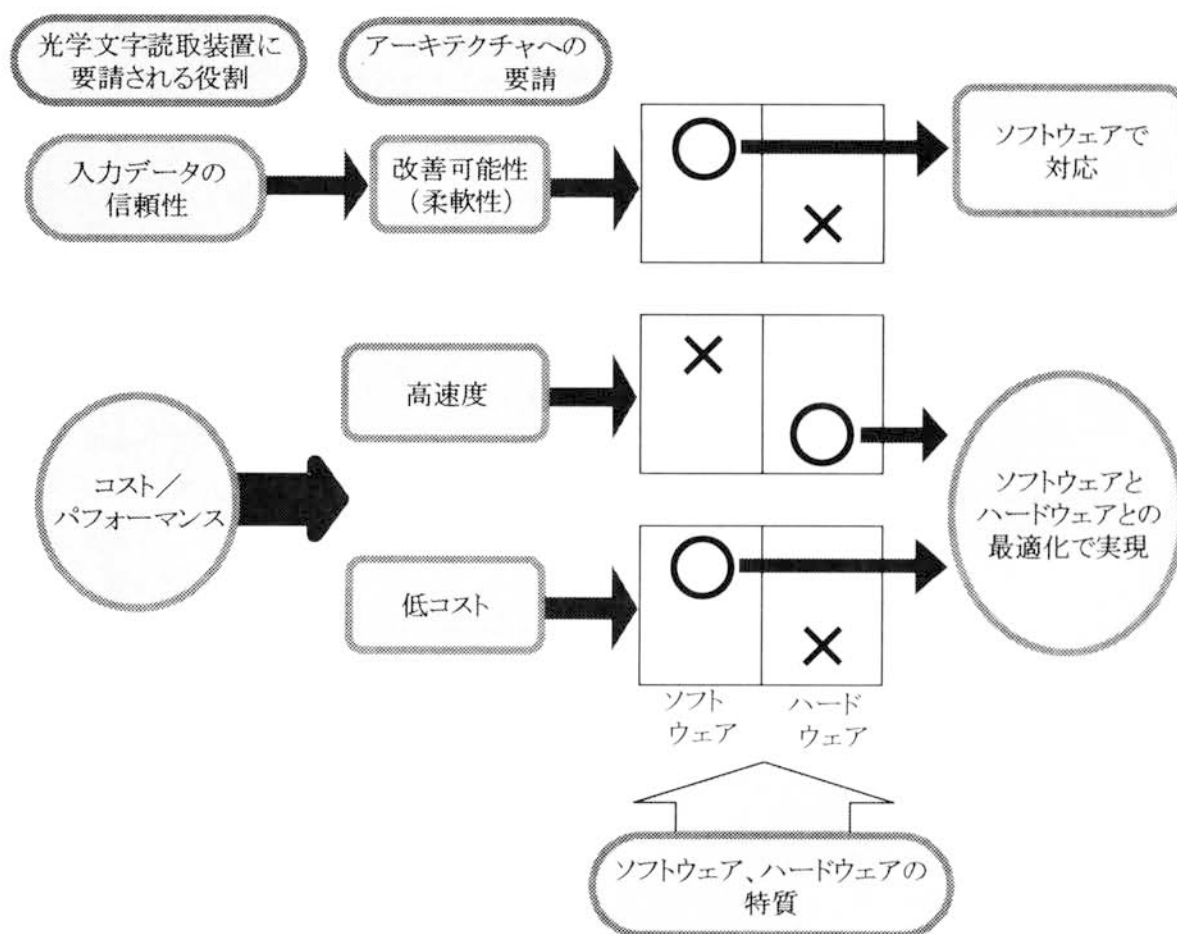


図 2.5.1 光学文字読取装置アーキテクチャの要点と課題

が必要である。

このため光学文字読取装置としては、高いデータ信頼性を持つために認識能力の最も高い文字認識方式を採用するとともに、アーキテクチャとしては、装置ハードウェア完成後もできるだけ改善可能性(柔軟性)を備えなければならない。

さらに、入力のコストパフォーマンスを向上させるために、高速性と低コストを追求しなければならないという性格も持つ。

また、アーキテクチャを構成するソフトウェアとハードウェアの特質を考慮し、改善可能性(柔軟性)には、改良にハード部品の交換が必要なハードウェアではなく、プログラムの交換で改良可能なソフトウェア対応とし、高速化のためには、並列並行処理が可能なハードウェアの採用が必要となる。

このため、光学文字読取装置のアーキテクチャには、改善可能性(柔軟性)のため、ソフトウェアの能力を生かし、また、コストパフォーマンス向上のため、ソフトウェアとハードウェアとの最適化を図らなければならない。

第 3 章

アーキテクチャの決定手法

④ デバッグフェーズ

- 単体デバッグ
- 連動デバッグ

詳細設計・試作機製作後、各部の単体デバッグを行なう。
 単体でそれぞれ確認されたハードウェア、ソフトウェアは統合され、連動したデバッグを行ない全体の動作を確認する。

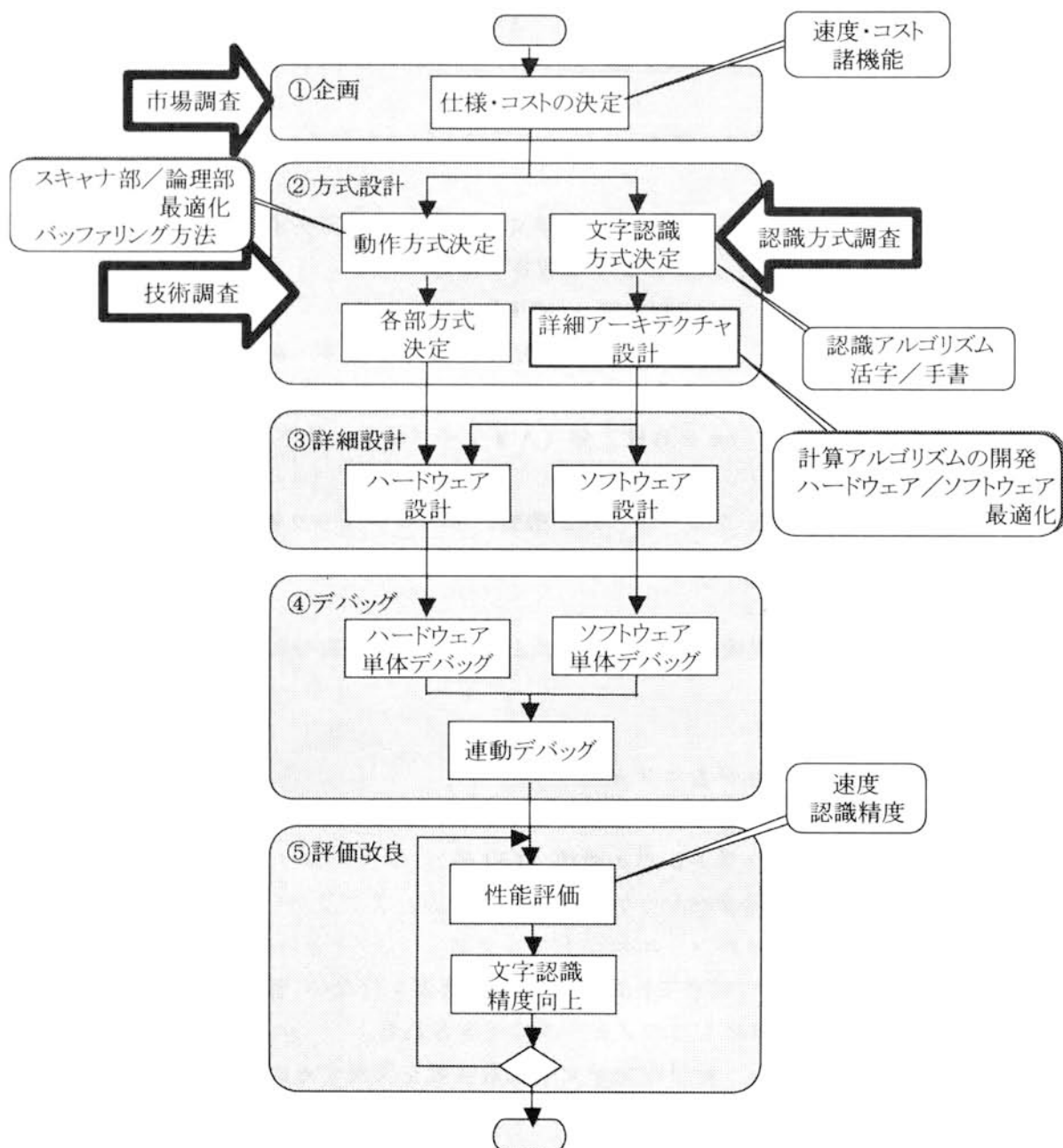


図 3.1.1 開発手順

⑤ 評価改良フェーズ

光学文字読取装置の開発過程において、評価改良フェーズは一般の装置とは異なる重要性をもつ。一般の装置においても、評価改良は重要であるが、その主眼は主に仕様で定められた性能が実現できているかの評価が中心となる。

光学文字読取装置においても、仕様に定められた機能性能が実現できているかの評価および改良は行なわれるが、これに加えて光学文字読取装置では、基本的性能である文字認識精度の評価改良が行なわれる。

文字認識方式の研究開発は、通常シミュレーションで行なわれるため、大量のサンプルでの評価改良は困難である。また、文字認識精度は得られる文字画像（通常2値）の品質に大きな影響を受ける。文字画像の品質は、光電変換部の特性に大きく左右されるため、光学文字読取装置ハードウェア完成後、整合性の評価が必要となる。また、計算アルゴリズムの影響の評価、装置開発中に得られた文字認識研究成果の適用も必要となる。

このため、これらの総合した評価として、このフェーズでは大量のサンプルによる文字認識精度の評価および改良が行なわれる。改良に当たっては、前章で述べたように、ハードウェアの変更ではなくソフトウェアの変更で行なう方針とする。ただし、まれに大幅な改良が必要な場合、ハードウェアの変更に及ぶ例もある。これは、②方式設計フェーズでの見通しの悪さの結果と言える。

また、ソフトウェアでの対策では、処理速度を低下させないことも重要な評価項目である。

3.2 方式設計

本節では、光学文字読取装置の開発で最も重要な位置を占める方式設計フェーズにつき述べる。

方式設計フェーズでは、図 3.2.1 に示すように2つの方式を決定する。1つは、光学文字読取装置の基礎となる文字認識方式の決定であり、他の1つは、装置全体のハードウェア構成・動作を決定する動作方式の決定である。また、前者は、文字認識能力に関わり、後者は処理速度およびコストに関わる。なお、認識方式は、実際には処理速度および装置コストに大きな影響を与えるが、光学文字読取装置の最も重要な性能である文字認識能力をその時点で最高のものにするとの方針を取る場合、処理速度および装置コストは以後の設計段階で解決しなければならない。

動作方式により速度の決定要因および論理部の役割が決定する。また、ハードウェア主体かソフトウェア主体かも明らかになる。この動作方式は、アーキテクチャ設計の実現方法を大きく左右する要因であるため、その内容を以下に述べる。

3.2.1 動作方式

第2章において、光学文字読取装置のアーキテクチャの課題はコストパフォーマンス追求のためハードウェアとソフトウェアの最適化を図ることが必要であることを述べた。

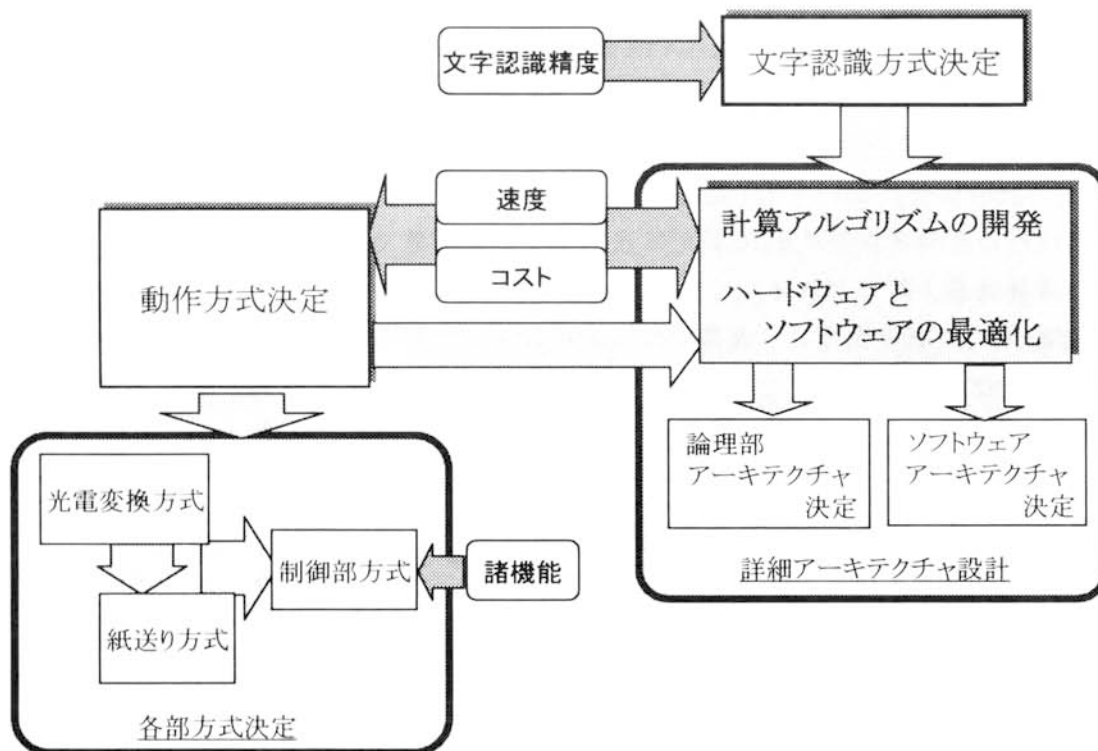


図 3.2.1 方式決定手順

動作方式の課題もコストパフォーマンス向上のため、高速性と低コストの追求であり、動作方式の決定にもアーキテクチャ決定と同様のアプローチが必要である。

動作方式の決定当り、最適化を図る部位はスキャナ部と論理部である。また、動作方式は光学文字読取装置の構成・動作を定めるため、アーキテクチャに先立ち決定される。この決定を受けて、認識部の持つべき特性、すなわちハードウェア主体かソフトウェア主体かが定まる。

以下では、光学文字読取装置の処理速度を決定する要因、動作方式およびアーキテクチャに重要な位置を占めるメモリの役割、および具体的な動作方式について述べる。

① 処理速度

光学文字読取装置の1帳票当りの処理時間Tは、下記の式で表される。

$$T=f(T_f, T_s, T_r, T_c) \quad (3.2.1)$$

T : 帳票1枚当りの処理時間

T_f : 帳票1枚当り紙送りに要する時間

T_s : 帳票1枚内の読取りエリアを走査する時間

T_r : 帳票1枚内の総文字認識時間

T_c : 帳票1枚内のデータチェック/出力時間

(i) T_f : 紙送り時間

T_f は、文字を全く読取らず、単に帳票を送るだけの時間である。図 3.2.2 に紙送り機構の概略を示す。

光学文字読取装置においては、帳票を1枚ずつホッパより給紙し、光電変換部を経由してスタッカに格納する。これに要する時間 T_f は、帳票の長さ L 、帳票間のギャップ G および紙送り速度 V_f で決定する。 V_f が一定速度の場合は、 T_f は、式(3.2.2)となる。なお、紙送り機構によっては V_f は一定速度ではない場合もある。

$$T_f = \frac{L+G}{V_f} \quad (3.2.2)$$

ここで、 T_f : 紙送り時間
 L : 帳票の長さ
 G : 紙送り間隔
 V_f : 紙送り速度

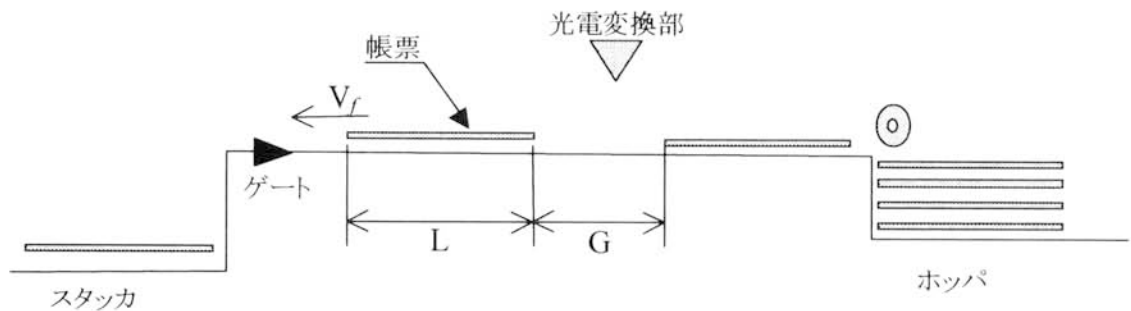


図 3.2.2 紙送り機構の概略図

(ii) T_s : 走査時間

T_s は、帳票上の読取りエリアを光電変換部が走査する時間であり、ビデオ信号の取りこみ時間とその準備時間（走査前および後処理など）の和である。

ラスタースキャンでは式(3.2.3)で表される。

$$T_s = T_{sv} + T_{sc} \quad (3.2.3)$$

T_{sv} : ビデオ信号入力時間

$$T_{sv} = \frac{S}{pq} \times t_b \quad (3.2.4)$$

T_{sc} : 準備時間

S : 走査エリアの面積
 (> 読取りエリア)

p : 縦サンプリングピッチ

q : 横サンプリングピッチ

t_b : 1メッシュの入力時間

ラスタースキャンは、最も一般的な

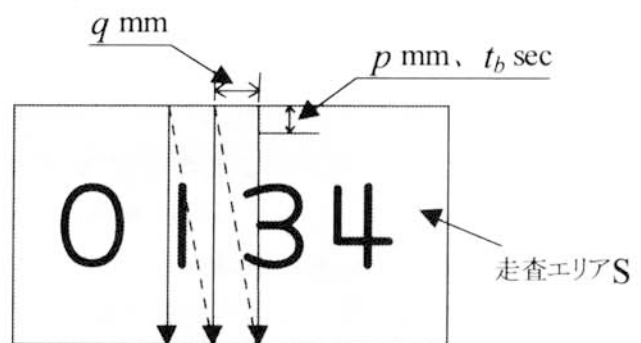


図 3.2.3 ラスタ走査

走査方法で、図 3.2.3 に示すように走査エリア S 全体をメッシュに分けて走査する。このため、ビデオ入力時間 T_{sv} は、走査エリアの面積および縦横の走査サンプリングピッチと 1 メッシュ（画素）の入力時間 t_b で決定する。なお、準備時間 T_{sc} は動作方式により異なるが、紙送り機構のスタート/ストップ時間などが含まれることもある。

(iii) T_r : 文字認識時間

T_r は、メモリに入力された文字の画像情報から、前処理および認識処理を行なう時間である。 T_r は下記の式で表される。

$$T_r = \sum t_L + \sum t_F + \sum t_R \quad (3.2.5)$$

t_L : 行切出し時間

t_F : フィールド切出し時間

t_R : 文字認識時間

文字の認識は、2 値化された画像から行(t_L)についてフィールド(t_F)の切出しを行なう。 t_R には、複数の文字が含まれるフィールド画像から、1 文字の切出し、前処理、特徴抽出および照合などの文字認識処理の実行時間からなる。

(iv) T_c : データチェック/出力時間

読取った文字データの妥当性をチェックする時間で、通常は少ない値（数十 msec）であるため、検討時には無視される。

② メモリの位置付け

光学文字読取装置は第 1 章で述べたように世代と共にハードウェアからソフトウェアへの転換が進んできた。これを主導している不可欠な要因に、メモリ（RAM）の大幅な低価格化、大容量化がある。

光学文字読取装置においてメモリは、プログラムおよび辞書データを格納すると共に認識すべき文字の画像を一時格納するバッファメモリとして使用され、アーキテクチャ構成に非常に大きな位置を占める。

メモリが非常に高価であった初期には、スキャナ部（光電変換部および紙送り機構）においてバッファメモリが少なくすむ動作方式が採用されていた。メモリが低価格化し、スキャナ部の対メモリコストが逆転するに従い、バッファメモリを増し、スキャナ部をシンプルな動作・構造として全体のコストの低減を図る動作方式が採用されてきた。

また、認識精度向上のため認識の研究は、前処理等の一律処理による情報欠落を回避し、入力される原パターンそのものから特徴を抽出する方式に発展させてきている。これにより、認識各段階で必要な処理が複雑増大化し、ソフトウェアのウェイトが増すに従い、プログラムのためメモリ容量の増大が要請されてきた。また、同時にソフトウェア処理の特性である認識時間のばらつきを吸収し、効率の良いアーキテクチャを実現するため、バッファメモリの大容量化も必要とされた。

このようにメモリは、光学文字読取装置に最も重要なキーデバイスであり、実用化研究の方向を定めてきた。

実際の光学文字読取装置に使われるメモリの容量はコストパフォーマンスを考慮して制限されるが、メモリの低価格化に従い、メモリ容量は増大してきた。すなわち、光学文字読取装置のメモリ容量はメモリ低価格化に逆比例して許されるだけのメモリ容量を増加させてきたと言える。

③ 動作方式

光学文字読取装置の処理時間 T の構成は、動作方式により決定する。
代表的な動作方式には、表 3.2.1 に示すものが実用に供された。

表 3.2.1 動作方式

動作方式		長所	短所	走査単位 入力用バッファメモ リサイズ例
ドキュメントリーダ型		最高速度を得る	認識速度に 高速が必要	1行/枚 1～40 k byte
ページリーダ	ランダム走査型	走査時間最小	光電変換部の 制御が複雑	1フィールド 5～40 k byte
	行単位走査型	比較的安価な 装置用	低速	1行 40 k byte
	全面走査型	書式の柔軟性が 大きい	バッファメモ リ大	1枚 1～4Mbyte

(ア) ドキュメントリーダ型^(*)

最高速度を得る動作方式で、読取る帳票は印刷文字1行とマークといった大量に発生する小形のターンアラウンド帳票を対象とする。電力ガス業界の検針票、各種入金票などの集中入力用の光学文字読取装置に採用される。毎分1,000枚以上の入力が可能で、1時間で数万枚を入力処理できる。

処理時間 T は T_f のみとなるようにアーキテクチャを構成する。文字の走査は帳票の走行と同期して行なうため、 T_s は T_f 内に含まれる。文字の認識も T_s 内に含まれるように構成するため、文字の認識速度は走査速度より高速にしなければならない。

このため、 $T_f > T_s, T_r, T_c$ となるようにアーキテクチャを設計する。

$$T \approx T_f$$

T_r は最大時間が制限されるため、ハードウェアが主体の構成とする。また認識論理には、この制限時間内に認識が終了するよう余裕を持った構成が必要となり、コストはかかるが、他に変わりがたい処理速度が得られる。

(イ) ページリーダ

ページリーダは、認識する行が多いなど、複雑な書式の帳票の読取を目的とした光学文

^(*) ドキュメントリーダは近年普及してきた印刷文書 OCR ではなく、OCR 業界での専門用語であり、その定義は本文に述べる通りである。

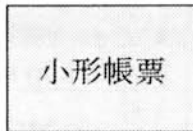
字読取装置である。

ページリーダーにおいては、多数の文字を読取るために文字認識を主眼にした動作方式であり、処理時間 T は主に T_r で決定するようにアーキテクチャを構成する。これは特にソフトウェアで認識処理されるため、処理時間が一定でない文字認識に効果的な構成であり、手書文字を対象とする光学文字読取装置などに用いられる。

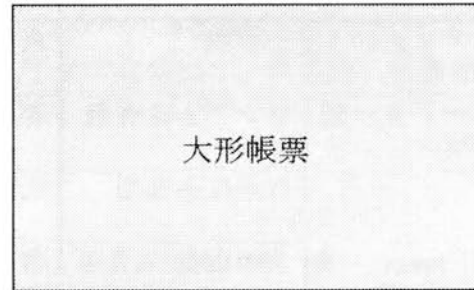
装置はコストを抑え、無駄の無いアーキテクチャをとるため、読取る帳票の形式によって速度が異なる。図 3.2.4 にページリーダーの定性的な速度の特性を示す。

ページリーダーのアーキテクチャの特徴は、 $T_s < T_r$ である。すなわち文字の画像の入力が認識処理より速いため、文字画像の格納のために大容量のバッファメモリが必要なことである。

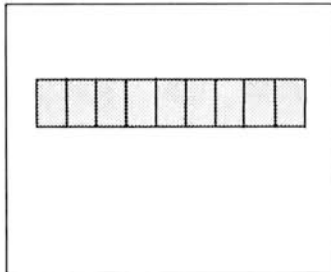
帳票の大きさ



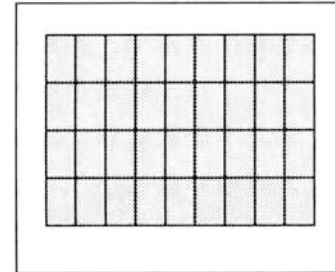
速い > 遅い



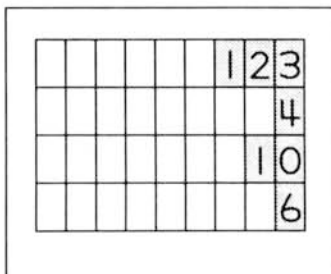
読取り文字数



速い > 遅い



記入文字数



速い > 遅い

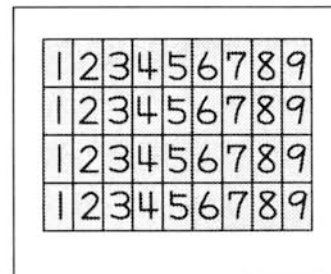


図 3.2.4 ページリーダーの速度特性

処理時間は、文字数が多いときには認識時間で決定する。

$$T \approx T_r$$

認識処理はソフトウェア主体で構成され、バッファメモリにより処理時間のばらつきは吸収されるため、処理速度は平均時間で評価される。このため、評価はモデルパタンなどで行なわれ、この短縮にハードウェアとの最適化が研究される。

(a) ランダム走査型

バッファメモリを最小の1フィールド分のみ用意し、かつ複雑なフォーマットの帳票を読取るために採用された動作方式である。

$$T = T_f + f'(T_s, T_r)$$

ここで、 $f'()$ は走査と認識のタイミングによる関数。

紙送り機構は帳票を光電変換部に停止させる。光電変換部は、読取るべきフィールドのみ走査し、文字認識部に渡す。光電変換にはフライングスポットスキャナなどが用いられ、複雑な動きが可能で、高速なページリーダーに採用された。

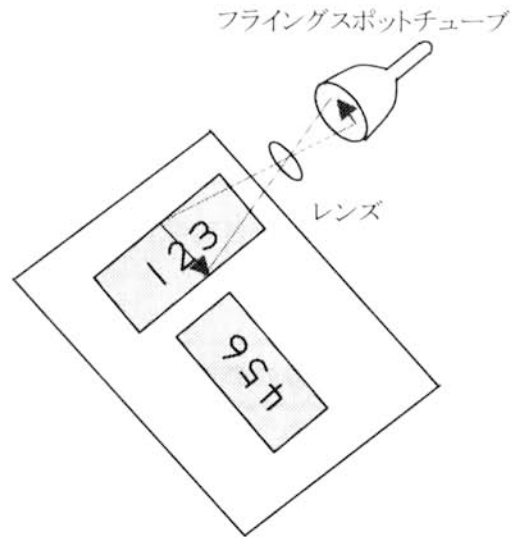


図 3.2.5 ランダム走査型

(b) 行走査型

安価な手書文字を対象とする光学文字読取装置に採用された動作方式で、1行単位に走査される。光電変換部には、レーザースキャナやCCDセンサが用いられた。

紙送りの動作は、行毎に帳票を停止させる。走査と文字認識の終了を待ち、終了後次の行まで帳票を進ませる。

$$T = T_f + f'(T_s, T_r)$$

ここで、 $f'()$ は走査と認識のタイミングによる関数。

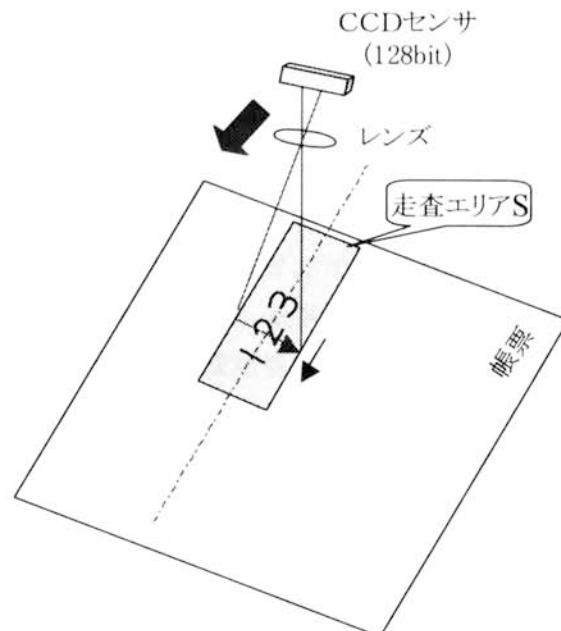


図 3.4.6 行走査型

(c) 全面走査型

メモリが安価となりバッファメモリとして十分に使用できる現在では、帳票全面のイメージを走査し、全画像を

メモリに取りこむ動作方式が主流となった。紙送り機構の動作は一定速度で帳票を静止しているCCDセンサの下を走行させ、画像の入力が行なわれる。通過後、帳票を停止させ、文字認識終了を待つ。文字認識終了後、帳票をスタッカに送る。

シンプルな動きであるため、紙送り機構および光電変換部は低コストを追求される。

$$T = T_f + f''(T_s, T_r)$$

ここで、 $f''()$ 走査と認識のタイミングによる関数。

3.2.2 各部方式の決定

動作方式を踏まえて、実現技術調査を行ない各部の方式を決定する。ここでは、光学文字読取装置の根幹を成す文字認識方式と認識性能・速度性能に大きな影響を与える光電変換方式が中心課題である。

(i) 光電変換部

光電変換部は、認識性能、速度性能に大きく影響するため、速度性能に重点をおき、実現技術から光電変換素子、照明素子を決定する。通常のイメージスキャナと異なる光学文字読取装置の光電変換部の特徴としては、ドロップアウトカラー^(*)、および黒背景などがある。

(ii) 紙送り機構

光電変換部と対になって動作するため、光電変換方式採用を受けて決定される。特に速度要因およびコストが考慮される。実業務で問題となるランニングコスト低減のため、紙質の制限緩和に多大の努力が費やされており、35kgの薄紙、普通紙、再生紙への紙質範囲の拡大などの成果をあげている。また、認識精度にも影響が大きい。特に文字の切出し位置精度を決定する走査精度は重要な項目となる。合わせて、操作性、安全性、信頼度なども考慮される。

3.2.3 認識方式の決定

光学文字読取装置の主目的の文字認識をなす最重要技術であり、コスト、速度などにも多大の影響を与えるが、まずはコスト、速度を除外して最も文字認識能力の高い方式を採用する。また、認識方式は、読み取り対象によって異なることが一般的である。

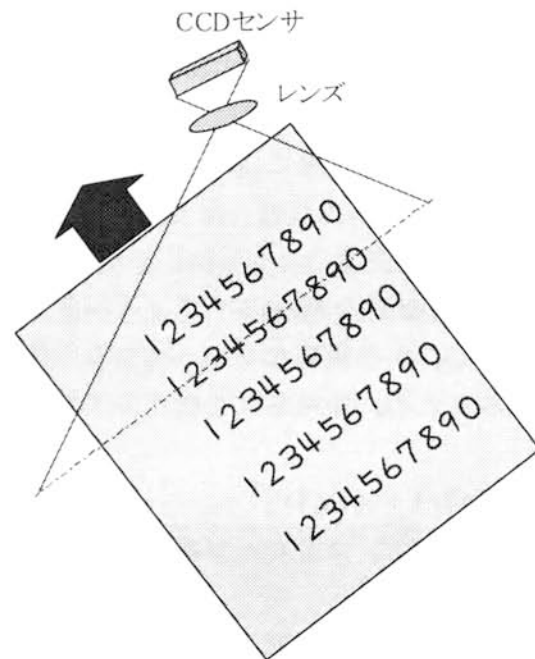


図 3.4.7 全面走査型

(*) 人には見えるが、スキャナで入力した画像には現れない印刷色。

例：印刷文字：相関法

手書文字：構造解析法

3.2.4 詳細アーキテクチャ設計

採用された文字認識方式を光学文字読取装置に実現するアーキテクチャを設計する。光学文字読取装置実現の為に非常に重要な位置を占め、この結果により光学文字読取装置開発の成否が決定するフェーズである。

図 3.4.8 に手順を示すが、採用された文字認識方式を改悪することなく、仕様フェーズで決定された速度／コストを実現するため、

最新の技術を適用した多くの案の比較検討や検討の繰返しなどにより、ハードウェアおよびソフトウェアの最適化を図る必要がある。

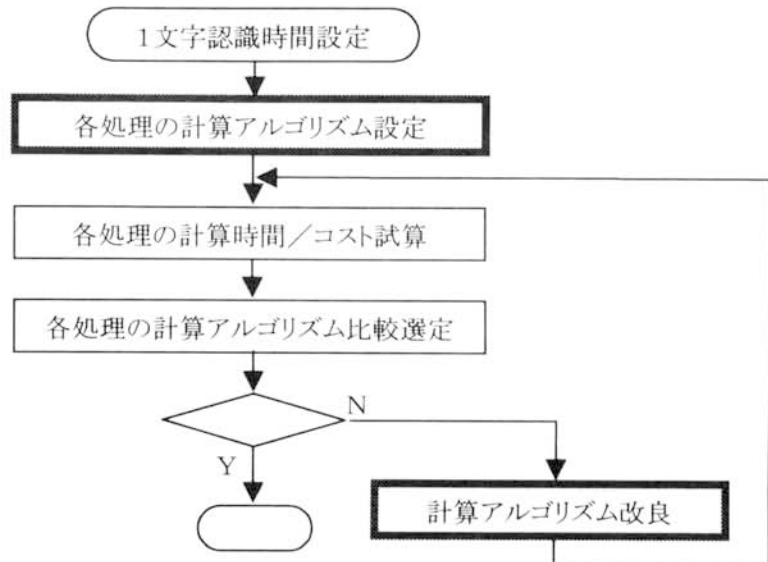


図 3.4.8 詳細アーキテクチャの設計

① 計算アルゴリズムの開発

計算アルゴリズムの研究は、採用された認識方式の実用化に非常に重要な課題となる。計算アルゴリズムは、文字の認識処理をどのような計算方法で実現するかを決定する。すなわち、採用された認識方式の原理は採用するが、具体的な計算方法は作り直す必要があるためである。

認識方式の研究は認識能力の向上を目的としているため、実用化の見地すなわち速度やハード規模などは考えられていないことが多い。

また、研究は通常シミュレーションで行なわれ、研究の過程では多くの改良が加わえられている。この結果、通常シミュレーションで開発したプログラムは、複雑な構造・構成そして研究データ採取用のルーチンが入っていることが多い。また、大型プロセッサ、高級言語の利用などで研究が行なわれていることなど、小規模な装置である光学文字読取装置にはそのまま適用できないことが多い。

このため、採用した認識方式を実用化の観点から見直す必要がある。

具体的には、認識原理はそのままこれと等価な適用可能な計算手段（計算アルゴリズム）を開発する必要がある。この開発には処理内容が等価なことを前提とするが、解決できないときには変更する場合もある。この場合はシミュレーションにもどり確認する。計

算アルゴリズムは、ハードウェアとソフトウェア両者に共通して効果を発揮する例もあるが異なることが多い。

動作方式の決定により、認識処理がハードウェア主体かソフトウェア主体かがおおよそ決定される。これに応じて計算アルゴリズムはハードウェアまたはソフトウェアの特性を考慮して開発される。その手法は、繰り返し処理に注目し、その繰り返し数の削減および繰り返し処理の計算を軽減することである。これはハードウェア論理においては、クロック数の削減、ソフトウェア処理においてはダイナミックステップ数の削減を図ることである。

計算アルゴリズムの効果の例を図 3.4.9 にて説明する。

ここでは 8 bit からなる a, b の乗算を、乗算命令のないシンプルなプロセッサで行なうと

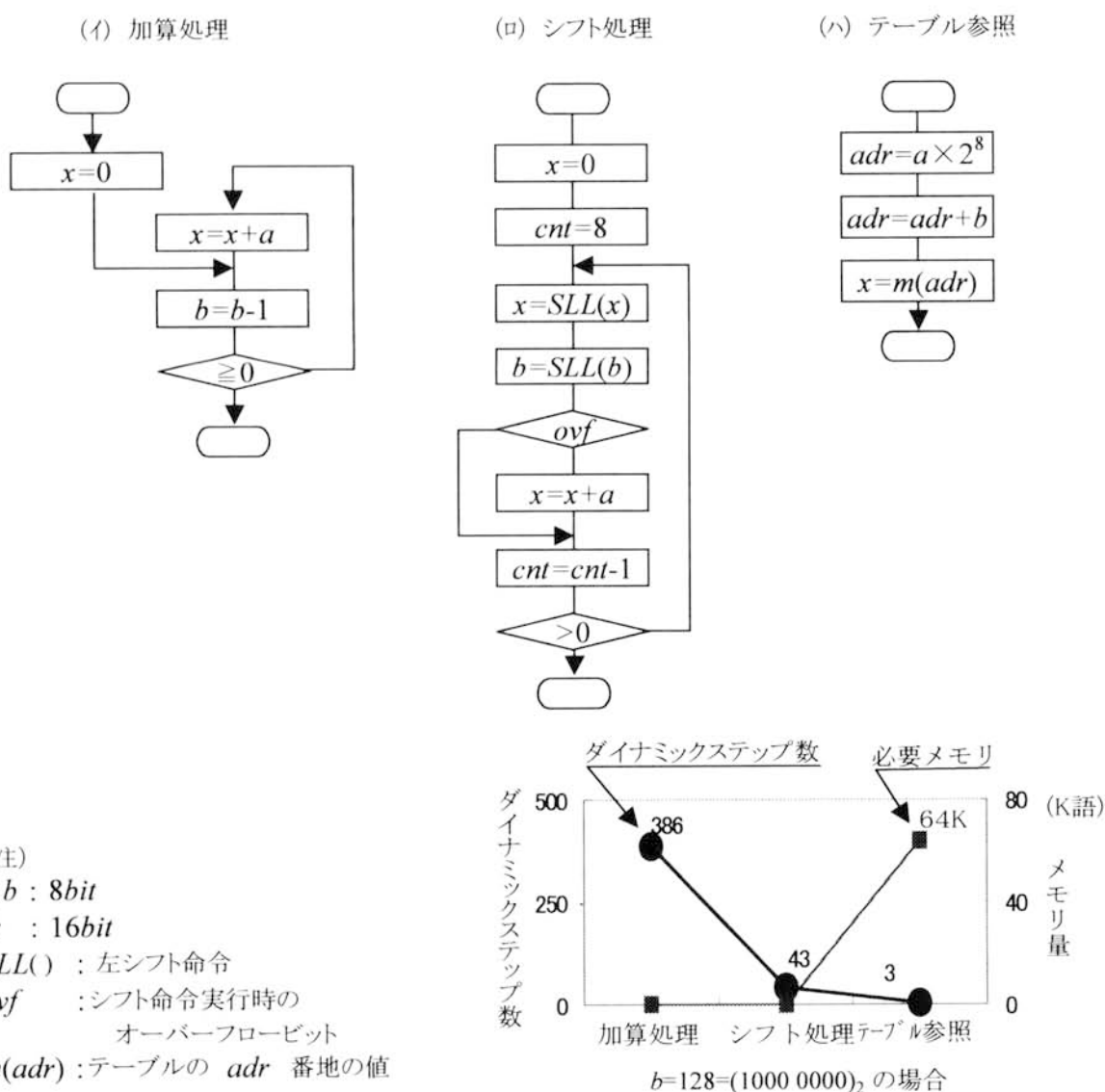


図 3.4.9 計算アルゴリズムの効果

する。加算のみで行なう場合、シフト処理で行なう場合は図 3.4.9(イ)および(ロ)に示すフローで実施される。また図 3.4.9(ハ)テーブル参照法においては 64 Kword のテーブルは必要であるが、高速に計算できる。繰返し回数は、(イ)加算処理では b 回(0~255)、(ロ)シフト処理では 8 回、(ハ)テーブル参照では 1 回であり、繰返し計算処理は、(イ)加算処理では 3 ステップ、(ロ)シフト処理では 5~6 ステップ、(ハ)テーブル参照処理では 1 ステップとなる。 $b=128$ の例を図 3.4.9 のグラフに示すが、このように、処理速度(計算時間)には大きな差が生じる。

このように、メモリなどのハードウェアの資源と、処理速度の最適化を図り最大の効果を得るために、計算アルゴリズムの研究は非常に重要な位置を占める。

② 速度/コスト見積

計算アルゴリズムを受けて速度を見積もる。ハードウェアの場合は、クロック数により速度を見積り、速度向上のためにはクロック数削減の論理改良、並列処理の拡充をおこなう。また、必要な場合にはパイプライン化も検討する。

ソフトウェアの場合は、ハードウェア論理の追加により繰返し処理を削減可能な命令の追加などを行なう。例えば、図 3.4.9 のシフト処理で専用の `cnt` 論理を追加し、"`cnt=cnt-1`" と "`test Jump`" を 1 ステップで実行する命令を追加する。これにより、繰返し処理のステップ数は 1 ステップ削減され処理時間が 1/6 短縮される。認識処理全体では数万ステップの削減に繋がることも多く効果が大きい。

なお、これらの対策はコストを評価しながら行なう必要がある。また、実際の開発においては①、②は同時にまた繰返し実行される。

[まとめ]

光学文字読取装置の開発手順のうち方式決定フェーズは、開発の成否を左右する重要な段階である。

ここでは、最新の技術に基づき動作方式を決定し、合わせて、最も認識精度の良い認識方式を採用する。

動作方式により判明する速度決定要因を考慮し、ハードウェア主体かソフトウェア主体かを考慮しながら、それらに応じた計算アルゴリズムの研究および処理速度・コストの試算を繰返し、最適な組合せを実現するアーキテクチャを決定する。

特に、精度を考慮して採用された文字認識方式を実用化するため、繰返し処理の回数削減、処理短縮化のため計算アルゴリズムの開発が重要かつ効果的である。

第 4 章

高速相関器

第4章 高速相関器

本章では、高速ドキュメントリーダーおよび高速ページリーダーとして開発された光学文字読取装置における印刷文字認識処理の中で、最も処理時間を要する相関値計算処理のハードウェアの事例を報告する。

4.1 印刷文字認識仕様

企画フェーズで定められた、ドキュメントリーダー^(*)の仕様を表 4.1.1 に示す。

表 4.1.1 印刷文字読取り仕様

仕様項目	仕様	内容
①読取り文字種	12種	数字10種、記号2種
②マルチフォント数	2フォント	
③動作方式	ドキュメントリーダー型	
③認識速度	毎秒1,200字	パイプライン処理
④認識方式	可変閾値法	類似度計算、判定、2段判定
	9位置相関法	
	マルチスライス認識方式	2レベル
	マルチフォント読取方式	2フォント
⑤標準パターンサイズ	32ビット×20スキャン/ カテゴリ	
⑥メモリ構成	32bit/word、 480ns/サイクル	

- 読取り文字種は、数字10種、記号2種の12種を2つのフォントで同時に認識するマルチフォント認識。また、2レベルのマルチスライス認識方式とするため、相関値の計算時間は単一フォントの48種に相当する。
- 認識速度は、毎秒1,200字とした。すなわち1文字の認識時間は830 μ secであり、ドキュメントリーダーであるため、紙送り速度から決定される最大認識時間である。
- 認識方式は可変閾値法とした。特に、文字認識精度重視のため、マルチスライス認識方式を、また相関計算には、9位置相関法を採用することとした。
- 標準パターン形式は、採用された光電変換部の走査線密度および印刷文字のサイズから縦32ビット横20スキャンとした。
- 標準パターンを格納するメモリは、32bit/wordとし、読出し時間は480nsである。

(*) p. 43 の脚注参照。

4.2 可変閾値法の理論式と計算アルゴリズム

詳細アーキテクチャを設計するに当り、認識アルゴリズムの理論式をハードウェアに適した計算アルゴリズムに変更する。

4.2.1 可変閾値法の理論式

可変閾値法は加重相関法を発展させた認識方式で、判定部に特徴がある。

➤ 類似度計算

第 i 標準パターン P_i と未知パターン X (認識すべき文字のパターン) の類似度計算式は式 (4.2.1) で与えられる。

$$R_i = \frac{(P_i, X)}{\sqrt{|P_i|} \cdot \sqrt{|X|}} \quad (4.2.1)$$

ただし、 (P_i, X) は標準パターン P_i と未知パターン X との内積であり、 $\sqrt{|P_i|}$ と $\sqrt{|X|}$ はノルムである。

➤ 第 1 および第 2 候補の選出

第 1 候補 m 、第 2 候補 n は、最大および次大の類似度を得るカテゴリである。

$$R_m = \underset{i}{\text{Max}} (R_i) \quad (4.2.2)$$

$$R_n = \underset{i}{\text{Next}} (R_i) \quad (4.2.3)$$

ただし、 Max は最大値を、 Next は次大値を与えるものとする。

➤ 判定

式 (4.2.4) および式 (4.2.5) を満たすとき、未知文字はカテゴリ m に属するものと判定される。

$$R_m > \varepsilon_m \quad (4.2.4)$$

$$R_m - R_n > \delta_m \quad (4.2.5)$$

ここで、

$$\varepsilon_m = k \cdot |P_m|$$

$$\delta_m = k' \cdot (|P_m| - |P_n|)$$

ε_m と δ_m は判定定数で、第 1 候補のカテゴリ m により決められている閾値である。これは、可変閾値法¹⁸⁾の特長で、認識精度向上に効果がある。また、 k は 0.8、 k' は、0.1 程度の値である。

4.2.2 可変閾値法の計算アルゴリズム

上記で定義された可変閾値法の理論式を光学文字読取装置に適用するに当り、ハードウェア回路やシンプルなプロセッサでは不得意な除算やノルム（平方計算）をなくした計算アルゴリズムを開発した。

① 重ね合せの計算

関連計算は、2 値化画像の対応する各画素の AND 処理後の黒画素（“1”）の数の合計値とする。

$$S_i = (P_i, X) = \sum_{x,y} P_{ixy} \cap X_{xy} \quad (4.2.6)$$

② 類似度計算式のアレンジ

類似度の計算は、カテゴリにより決定する相関係数 t_i により計算する。

$$T_i = t_i \cdot S_i \quad (4.2.7)$$

$$\text{ここで、} t_i = \frac{K}{\sqrt{|P_i|} \cdot \sqrt{|X|}} \doteq \frac{K}{|P_i|}$$

計算アルゴリズムでは、 $P_i \doteq X$ と近似することにより、事前に係数として計算しておくようにした。

ここで、 K は T_i を整数にするための定数である。これは、 R_i は $0 < R_i < 1$ であるが、ハードウェアとしては 1 以上の整数が望ましいことから一定の係数を乗じており、例えば T_i を $2^{22} \times R_i$ と成るように選ぶ。

③ 第 1 および第 2 候補の選出

式(4.2.2)、(4.2.3)と同様。

$$T_m = \underset{i}{\text{Max}} (T_i) \quad (4.2.8)$$

$$T_n = \underset{i}{\text{Next}} (T_i) \quad (4.2.9)$$

④ 判定

$$T_m > \alpha_m \quad (4.2.10)$$

$$T_m - T_n > \beta_m \quad (4.2.11)$$

$$\text{ここで、} \alpha_m = K \cdot k \cdot |P_m|, \beta_m = K \cdot k' \cdot (|P_m| - |P_n|)$$

以上のように、計算アルゴリズムでは、事前に標準パタンのカテゴリ毎に相関係数 t_i および判定定数 α_m 、 β_m を計算しておくために、認識処理時には、“1” ビットの加算、および乗算のみで計算が行なわれ、除算および平方演算は無くすることができる。

● 計算アルゴリズムと理論式の等価性

この計算アルゴリズムと理論式の等価性は $P_m = X$ が成り立つ仮定で保たれる。

これは、式(4.2.10)、(4.2.11)の判定でアクセプトと判定される場合の未知パターンは、最

大類似度を得るカテゴリ m に十分類似していると仮定することである。この仮定は、シミュレーションで確認を行った。

4.2.3 追加アルゴリズム

可変閾値法を実際の光学文字読取装置に適用し、良好な認識精度を得るためには、理論では現れない処理を行なう必要が経験上得られている。このため、実用化に当り以下のアルゴリズムを追加した。

① 9位置相関法 ^{4),25)}

理論では、未知パターンは、正しく位置の正規化が行なわれていることを前提としている。しかしながら、実際の入力パターンは、悪い条件での印刷、汚れなどによる印字品質の劣化があるため、正しい位置に正規化できないことが多い。

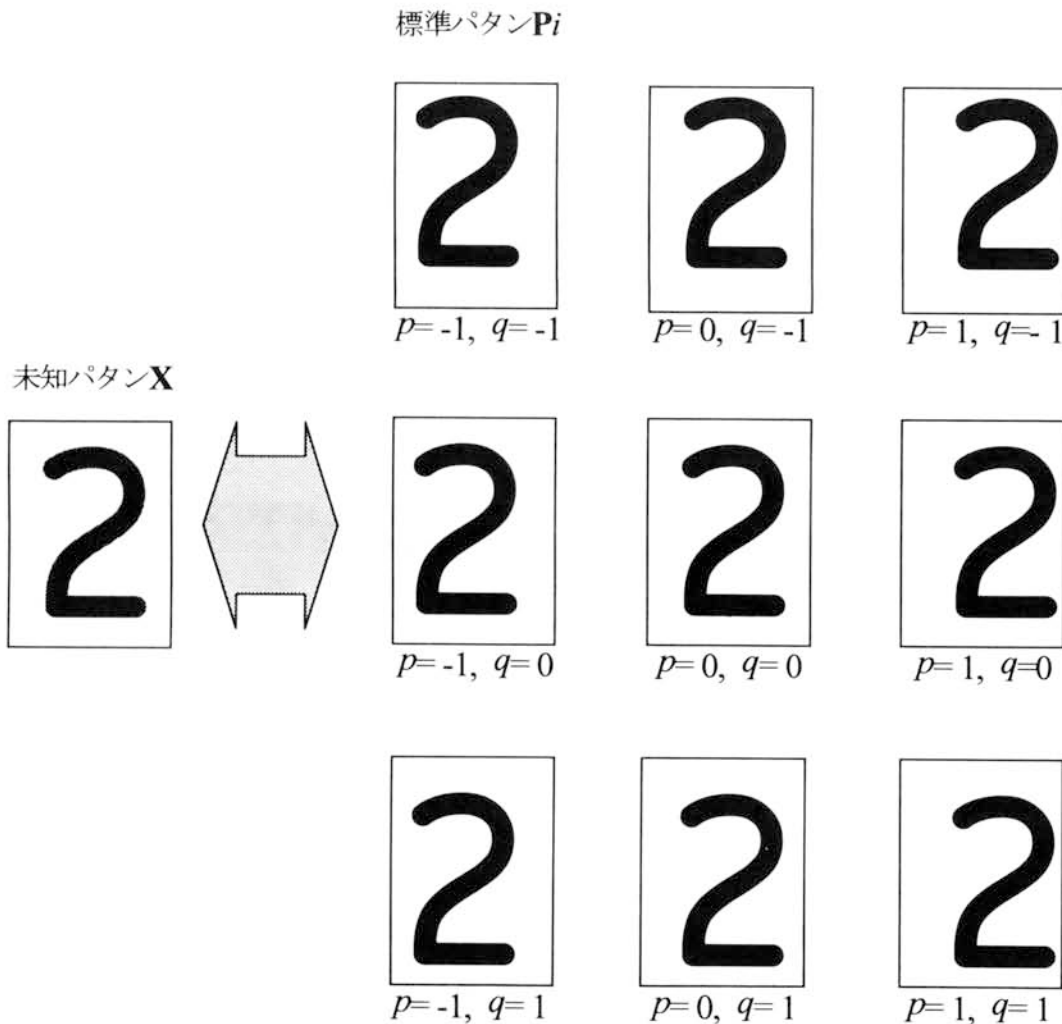


図 4.2.1 9位置相関法

このような劣化した文字についての位置正規化に対し、種々の研究もなされているが、劣化の原因が多岐にわたるため、全てのケースに対応する手法は得にくい。逆に、ある現象の文字品質劣化に対応した対策を実施した場合、他の想定していない劣化に対しては、その後の処理に悪影響が生じ、結果的に認識精度が悪化することが経験上得られている。

9位置相関法は、位置正規化は最もシンプルな縦横の外形寸法を基準とし、相関計算時、重ね合せ位置を左右上下にずらすことにより、位置正規化の狂いを補償しようとする方法である。具体的には、下記の計算を追加する。

$$S_i = \text{Max}_{p,q}(S_{ipq})$$

$$S_{ipq} = \sum_{x,y} P_{ix'y'} \cap X_{xy}$$

ここで、 $x' = x + p$, $y' = y + q$, $p = 1, 0, 1$, $q = 1, 0, 1$ とする。

9位置相関法の採用により、相関計算処理は、1位置に比べ9倍の処理が必要になる。これは、処理時間あるいはハード物量に大きな影響を与えるが、前述のごとく最高の文字認識能力を得るために採用された。

② マルチスライス認識方式

印刷文字は印刷時の文字品質に認識率が大きく影響される。これを防ぐ手法の1つがマ

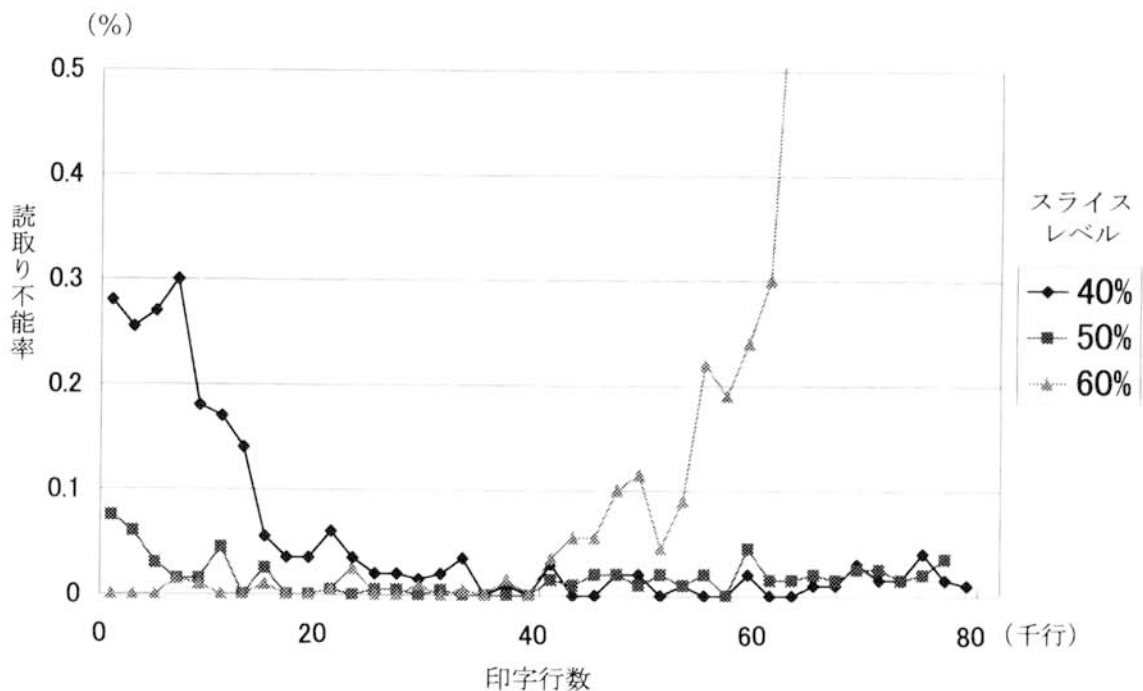


図 4.2.2 単一スライスでの認識結果

ルチスライス認識方式である。

図 4.2.2 にインパクトプリンタで印刷された文字データでの単一スライス認識での読取り不能率の例を示す。スライスレベルにより認識率に大きな差が生じている。現在でも多部紙の印刷に使われているインパクトプリンタでは、インクリボンが使用されるため、リボンのインクの消耗により印字の濃度が大きく変化する。すなわち、印字行数の少ないときはインクリボンのインクは十分であるため文字が濃く印字され、印字行数が多くなるに従い文字は薄く印字される。2 値化に当り一定のスライスレベルでは、前者は文字線が太くなり、後者では文字線が細くあるいは欠落する。

文字認識率はこの文字線の太さに影響される。図 4.2.2 において、スライスレベル 60% では文字線が細くなるため、印字行数の少ないときは良好な認識率を得るが、印字行数が多くなると文字線が細く、また、欠けるため急激に認識率が悪化する。逆にスライスレベル 40% では印字行数が多いときにも文字線の欠けが少ないため良好であるが、印字行数の少ない部分は、文字線が太すぎ悪化する。スライスレベル 50% は全印字行数におおむね良好な認識率を得るが、実稼動状態でこの状態を常時安定して維持することはマージンが少なく困難である。

このため、固定のスライスレベルでは、印字行数の少ないときと多いときの両者に対し安定したマージンのある認識率を得ることが難しかった。

マルチスライス認識方式は、これを補うための認識手法で、2 値化処理にて 2 通りの異なるスライスレベルの 2 値化画像を入力し、この 2 通りの 2 値化画像につきそれぞれ認識処理を行って 2 通りの認識結果を得る。この 2 通り認識結果からの正当な結果を選び最終結果とすることにより、安定した良好な認識率を達成する方式である。

③ マルチフォント読取方式

異なるプリンタで印字された帳票を読取る業務では、各々のプリンタのフォントが異なるため、このような業務に用いる認識方式がマルチフォント読取方式である。これは、読取るフィールドの印刷文字が複数フォントの内の単一のフォントで印字されることを利用する。この読取り方式は、文字単位ではそれぞれ単一のフォントとして認識する。

従ってフィールド全体の認識が終了した時点では、フォント毎の認識結果列が揃う。これら各フォントの認識結果列から、正当な結果列を選び最終結果とする。複数の結果列から正当な結果列を選ぶ方法としては、それぞれの結果列中の不読文字の少ない結果列を選ぶ方法²⁶⁾をとった。

4.3 詳細アーキテクチャ設計

4.3.1 可変閾値法の認識フローチャート

可変閾値法は、図 4.3.1 に示されるフローチャートにて、認識処理される。

光電変換部にて文字の画像がメモリに入力され、1 文字の画像が切出される。

この画像は、縦／横の投影パターンから文字の中心位置を計算される。

この位置が、縦 32 × 横 20 のワークエリアの中心にくるように文字のパターンが移動さ

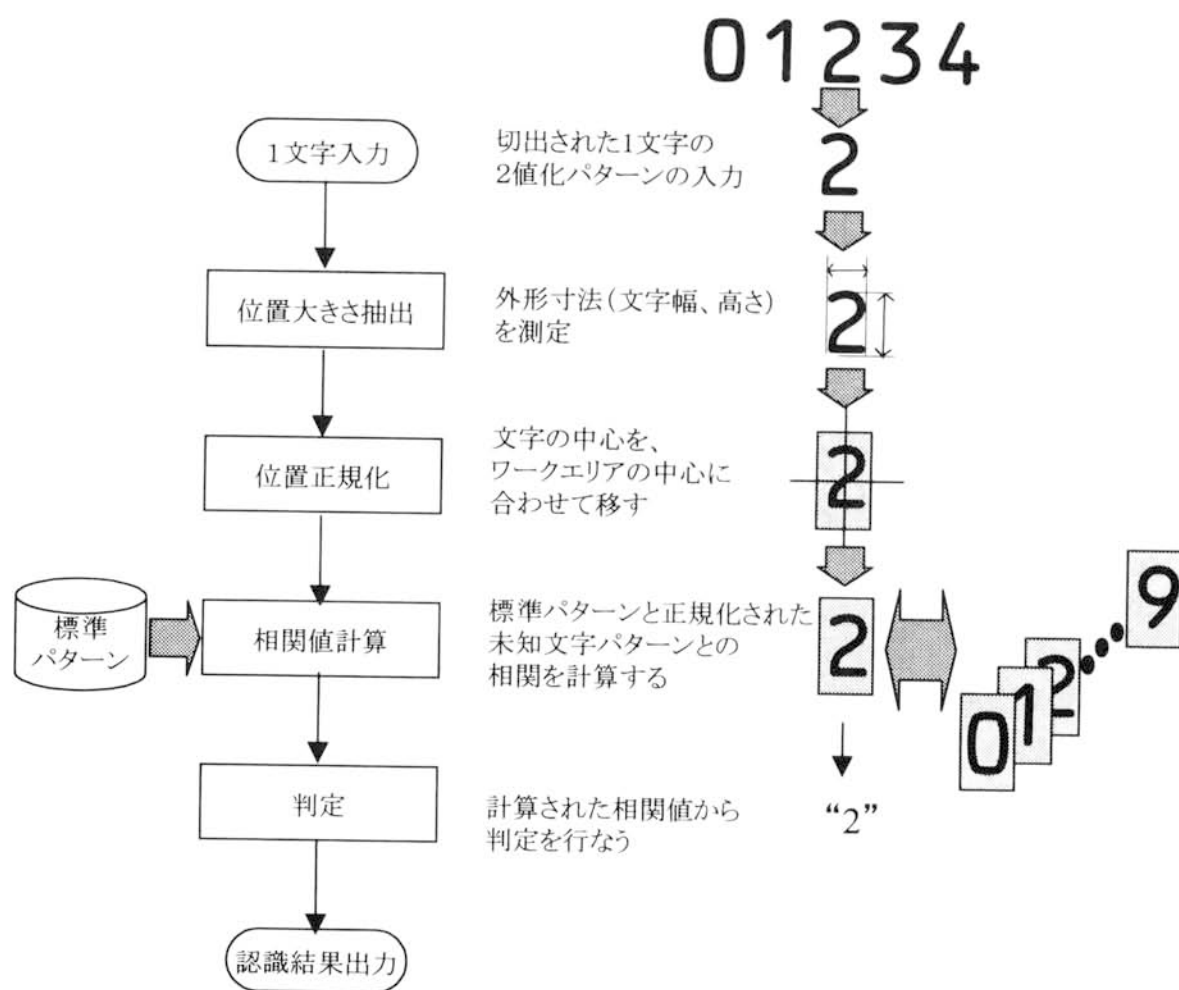


図 4.3.1 可変閾値法のフローチャート

れる。

この位置正規化画像に対し、標準パターンとの類似度が計算され、最大類似度および次大類似度が求められる。この類似度から判定処理により認識結果が出力される。

4.3.2 相関部のフローチャート

相関計算は、図 4.3.2 に示されるように計算される。

正規化された未知パターンは、カテゴリ毎に、9 位置での相関値を計算され、その最大値に対し、相関係数を乗算されて、類似度が求められる。

全カテゴリの類似度計算後、最大および次大類似度が計算され、判定処理に渡される。

4.3.3 詳細アーキテクチャの決定

アーキテクチャの検討は、表 4.1.1 の仕様、認識速度毎秒 1,200 字を 9 位置相関で実現することから行なう。最も処理時間がかかるのは相関計算処理、特に重ね合せ処理時の標準パターンおよび未知パターンのメモリからの読出し時間であることから、ここに注目し、

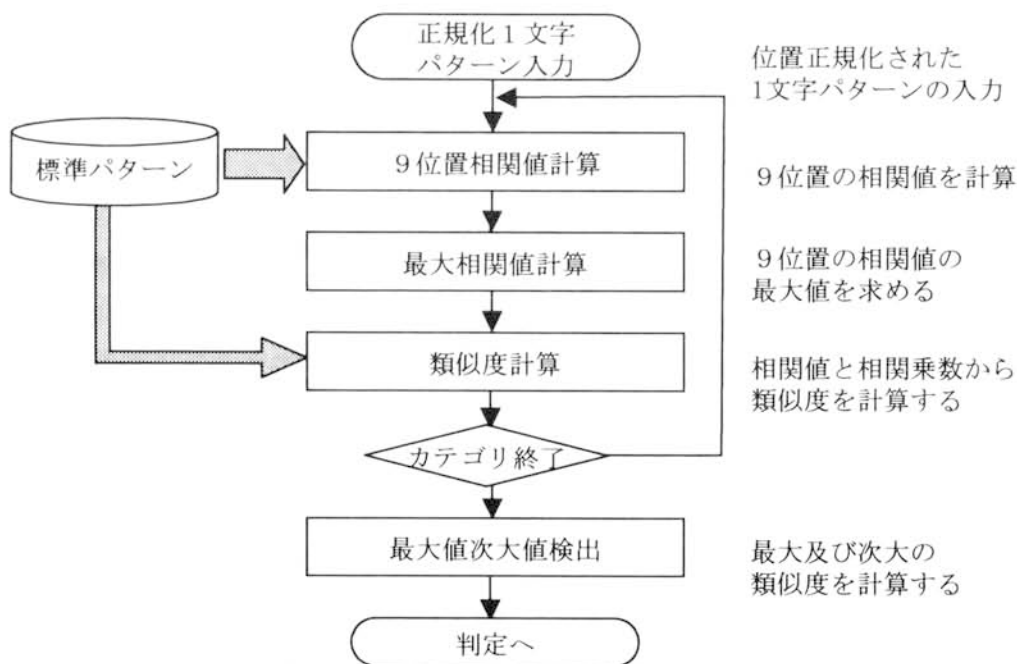


図 4.3.2 相関計算

アーキテクチャの検討を行なう。

図 4.3.3 に、3 方式の構成を示す。

(A) 相関計算のみをハードウェア化

これは、図 2.3.1(c)に相当するもので、プロセッサに相関計算命令を付加した例に当る。図 4.3.4 のタイムチャートから、メモリから標準パターンおよび未知パターンを 1word ずつ読み出すため、メモリの読出しアクセス回数は、

$$20W \times 2 \text{ パタン} \times 9 \text{ 位置} \times 48 \text{ 標準パターン} = 17,280 \text{ サイクル}$$

全アクセス時間は、

$$17,280 \text{ サイクル} \times 480 \text{ ns} = 8,294,400 \text{ ns}$$

これにより、毎秒 120 字以下の認識速度しか実現できないことがわかる。

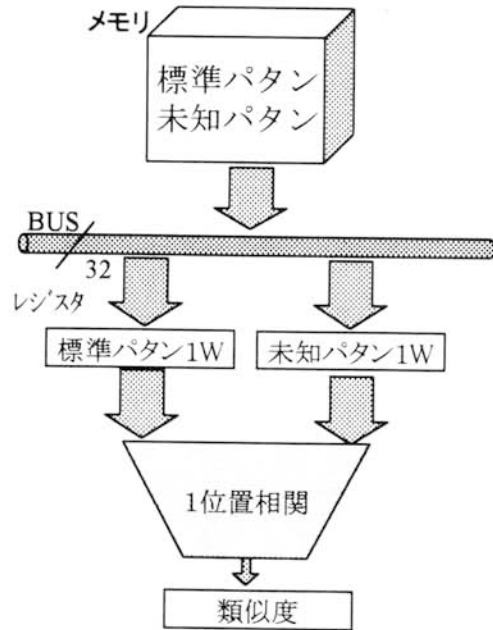
(注) 48 標準パターン = 12 カテゴリ × 2 フォント × 2 スライス

(B) 未知パターンをキャッシュメモリに格納

(A) では、48 標準パターンとの重ね合せ毎に、標準パターンと未知パターンを読み出す必要があるが、未知パターンは全てのカテゴリに対しスライスレベル毎に共通であることから、メモリとは独立したキャッシュメモリを用意し、メモリの読出し時間を削減する。すなわち、メモリからの標準パターン読出しと、キャッシュメモリからの未知パターン読出しを並行に同時に読み出す。これにより、メモリのアクセス時間は、

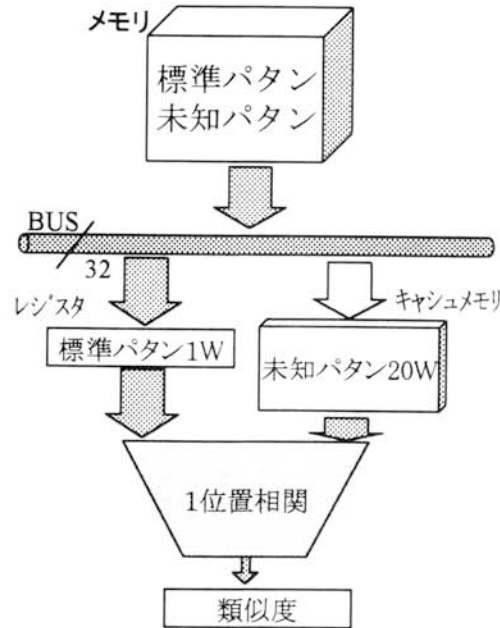
$$20W \times 9 \text{ 位置} \times 48 \text{ 標準パターン} = 8,640 \text{ サイクル}$$

(A) 相関計算のみをハードウェアにした場合



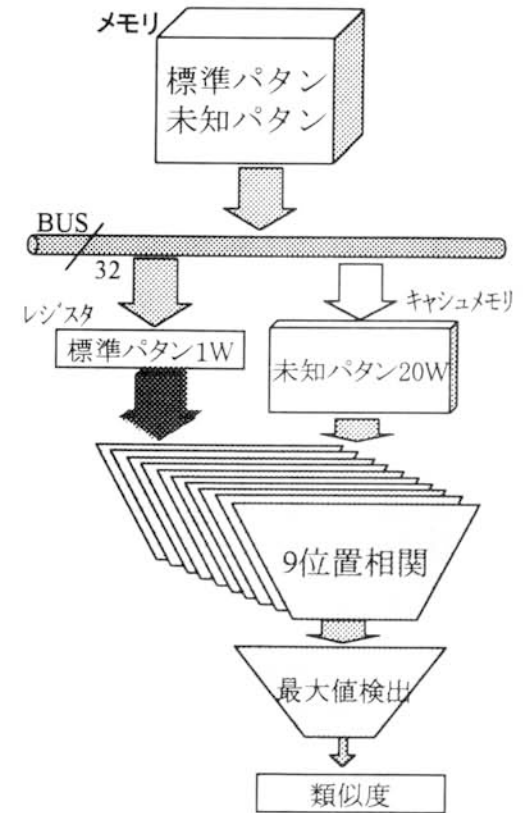
メモリアクセス回数
 $20W \times 2 \text{パタン} \times 9 \text{位置} \times 48 \text{カテゴリ}$
 $= 17,280 \text{サイクル}$
 $< 120 \text{字/sec}$

(B) 未知パタンはキャッシュメモリより読出し



メモリアクセス回数
 $20W \times 9 \text{位置} \times 48 \text{カテゴリ}$
 $= 8,640 \text{サイクル}$
 $< 241 \text{字/sec}$

(C) 9位置相関をハードウェアにした場合



メモリアクセス回数
 $20W \times 48 \text{カテゴリ}$
 $= 960 \text{サイクル}$
 $< 2,170 \text{字/sec}$

図4.3.3 相関計算とメモリ構成

全アクセス時間は

$$8,640 \text{ サイクル} \times 480 \text{ ns} = 4,174,200 \text{ ns}$$

毎秒 241 字と速度は向上したが、まだ仕様には不足である。

なお、キャッシュメモリへは、相関計算に先立ち、未知パターンをメモリより 1 回読出し格納する動作が増加する。

(C) 9 位置相関をハードウェア化

メモリアクセス時間を削減するため、ハードウェアを増強する。(B) では、48 種のカテゴリそれぞれに 9 位置相関を行なうため、標準パターンをそれぞれ 9 回メモリより読み出している。標準パターンを 1 回の読出しだけにするため、(C) では、相関計算論理を 9 回路用意し、1 回の標準パターンの読出しで 9 位置の重ね合せ計算を並列して計算する。これにより、メモリアクセス時間は、

$$20W \times 48 \text{ 標準パターン} = 960 \text{ サイクル}$$

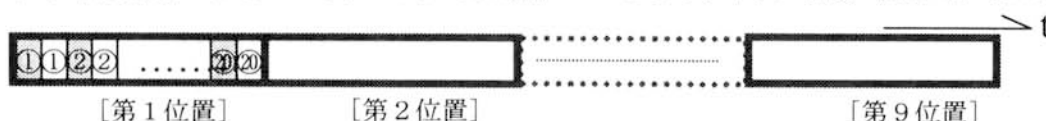
全アクセス時間は

$$960 \text{ サイクル} \times 480 \text{ ns} = 460,800 \text{ ns}$$

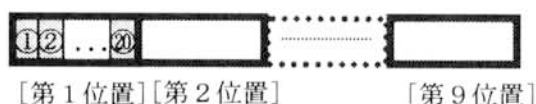
毎秒 2,170 字となる。これは、標準パターンを読み出すだけのアクセス時間だけであるが、他の時間を考慮してもパイプライン化することにより、相関計算は仕様を満たすと試算された。

なお、(C) は、未知文字 1 字に対し、48 標準パターンを 1 回読み出す最短時間である。したがって、さらなる速度向上には 1 度の標準パターン読出しに対し、複数の未知文字との相関計算の実施、あるいは複数の標準パターンを並行して読み出すなどのハードウェア増強が考えられる。また、1word のビット数を増す、例えば 64 ビットとすることにより倍の

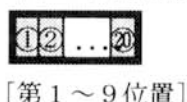
(A) 相関計算のみをハードウェアにした場合 ⇒ 360 サイクル必要 (20W × 2^{パター} × 9 位置)



(B) 未知パターンはキャッシュメモリより読出し ⇒ 180 サイクル必要 (20W × 9 位置)



(C) 9 位置相関をハードウェアにした場合 ⇒ 20 サイクル必要 (20W)



注：①：標準パターン第 1 word 読出しサイクル

②：未知パターン第 1 word 読出しサイクル

図 4.3.4 相関計算タイムチャート

速度が得られるなど、ハードウェアを増強することにより必要な速度が得られる。ただし、コスト面では不利となるため、仕様速度が得られる最小のハードウェア構成が最適なアーキテクチャとなる。

① 印刷文字認識動作

パイプライン処理による印刷文字のタイムチャートを図 4.3.5 に示す。

- ▶ 光電変換部は、1文字の切出し入力を行なう。
- ▶ 前処理部は、入力された文字パタンの投影処理および引続き正規化処理を行なう。
- ▶ 相関計算部は、正規化された未知パタンの相関計算を行なう。

各パイプラインハードウェアはプロセッサにより起動指令を受け、処理終了時、プロセッサに割り込みにより終了を報告する。プロセッサはこれを受け、次の処理を起動させる。各動作は、プロセッサの制御のもと並行して実施され、判定出力は、 $830\mu\text{s}$ 間隔で出力される。

ただし、第1文字の出力は、光電変換部の起動より、数ms後となる。このため、パイプライン処理では、少ない文字、例えば1文字のみを認識する場合数ms 必要なため毎秒1,200字の性能は得られないが、通常印刷文字の場合1フィールドに10～30文字あるため、フィールド単位での文字認識速度は概略毎秒1,200字となる。

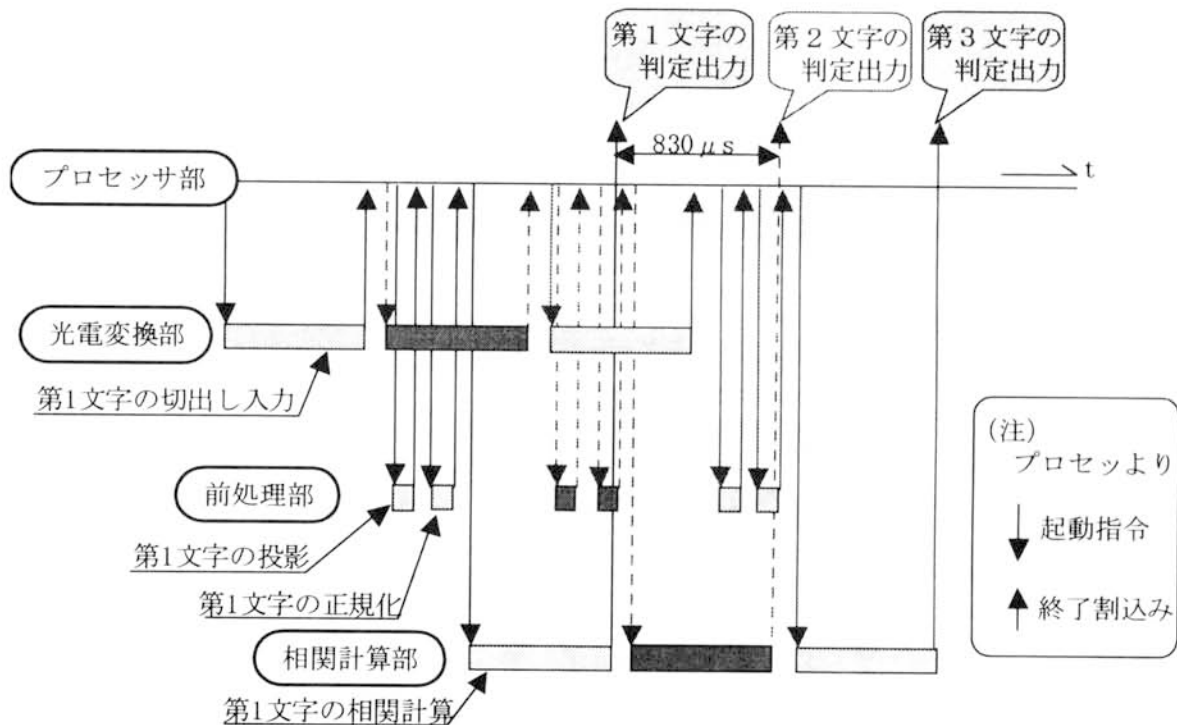


図 4.3.5 印刷文字認識のタイムチャート

② 相関器アーキテクチャ

高速相関器の機能は、図 4.3.2 に示す相関計算のフローチャートをハードウェアにて実現したもので、図 4.3.6 のように、構成される。

➤ 9位置相関計算部

9位置相関計算部は、9個の相関計算からなり、キャッシュメモリに格納された1個の未知パターンと標準パタンの9位置の相関が1度に計算される。

➤ 最大値検出部¹⁹⁾

9位置相関計算部で計算された9個の相関値の最大値が検出される。検出はシフト処理で計算される。

➤ 類似度計算部

最大値検出部で計算された相関値は、相関定数を乗ぜられ類似度が求められる。乗算は、加算とシフト処理で計算される。

➤ 最大値次大値格納/検出部¹⁹⁾

計算された、類似度は最大値次大値格納/検出部に格納される。ここには、3個の類似度が格納される構成となっており、新たに計算された類似度は、格納されている3個の類似度の中の最小値と置きかえられる。この動作は、最大値検出部と同様の論理でシフト処理により計算される。

➤ シーケンサ

シーケンサは、プロセッサの指令を受け相関器の動作を制御し、終了時にはプロセッサに割り込みにより終了報告を行なう。ROMとラッチ等で構成され簡単なプログラムで動作し、各部へのクロック供給、メモリとのやり取りを行なう。

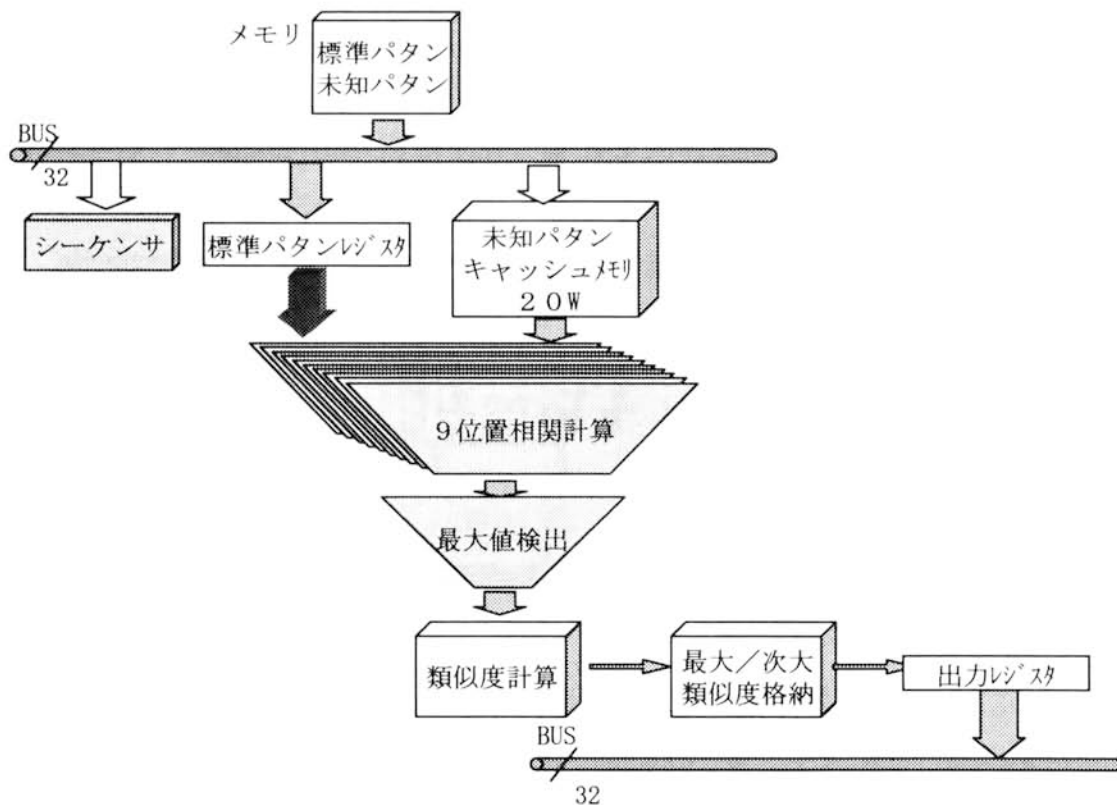


図 4.3.6 相関器の構成

図 4.3.7 に相関器の動作タイムチャートを示す。

起動指令を受けた相関器は、標準パターン P_1 から順次 P_n まで相関計算される。最大値検出は相関計算より 1 パタン遅れて計算される。同様に、類似度計算、最大値次大値格納も順次遅れて計算される。 P_n の計算が終了すると終了割込みに終了を報告し、相関器は停止する。このように相関器の内部もパイプライン動作で計算される。

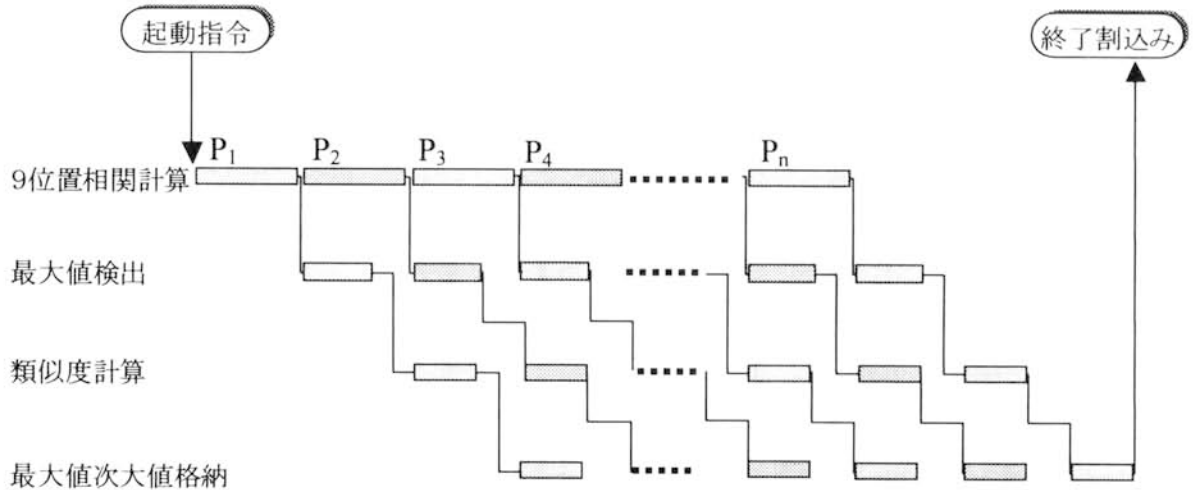


図 4.3.7 相関器の動作タイムチャート

③ 各部詳細

高速計算可能でハードウェア向きに改良された計算アルゴリズムがどのような論理

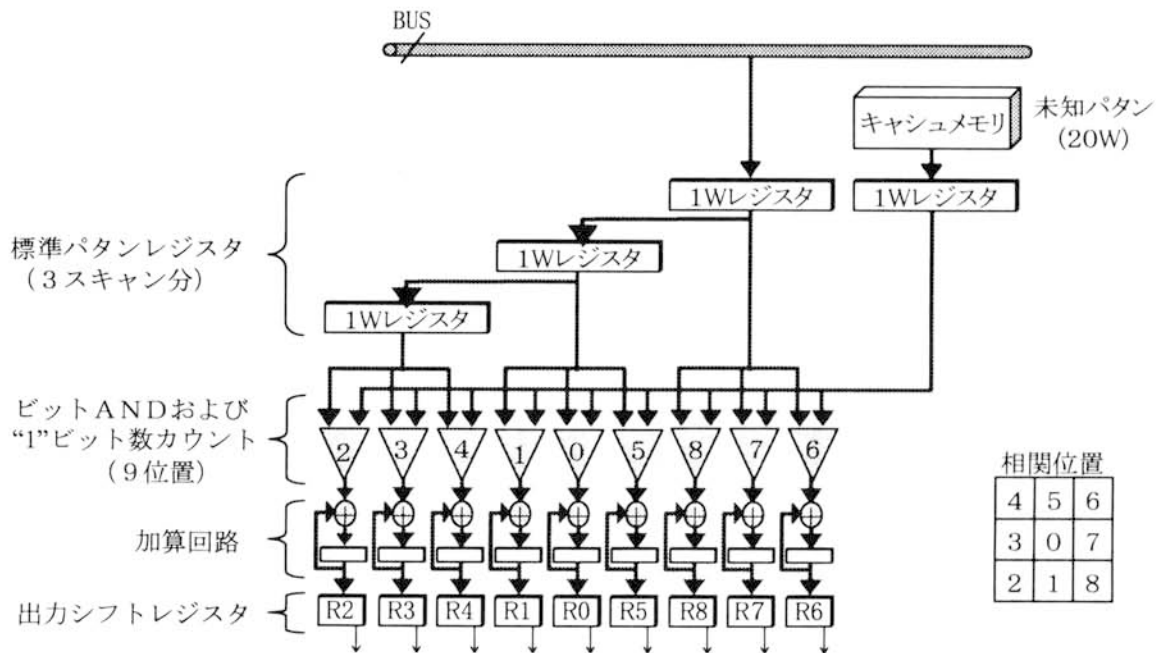


図 4.3.8 9 位置相関計算部

回路により実現されたかを示し、計算アルゴリズムの効果を述べる。

➤ 9 位置相関計算部

図 4.3.8 に 9 位置相関計算部のブロック図を示す。相関計算は未知パターンと標準パタンの対応する画素の AND がとられ、その結果が数えられる。相関は 1 word 32 bit が 1 クロックで計算され、これを 20 回加算して 1 つの標準パターンとの相関計算を終える。このクロックは、標準パターン読出しのメモリサイクルに同期させる。

➤ 最大値検出回路¹⁹⁾

図 4.3.9 に最大値検出回路を示す。9 個の相関値は、シフトレジスタより上位ビットから出力され最大値が出力レジスタに上位ビットから順次セットされる。

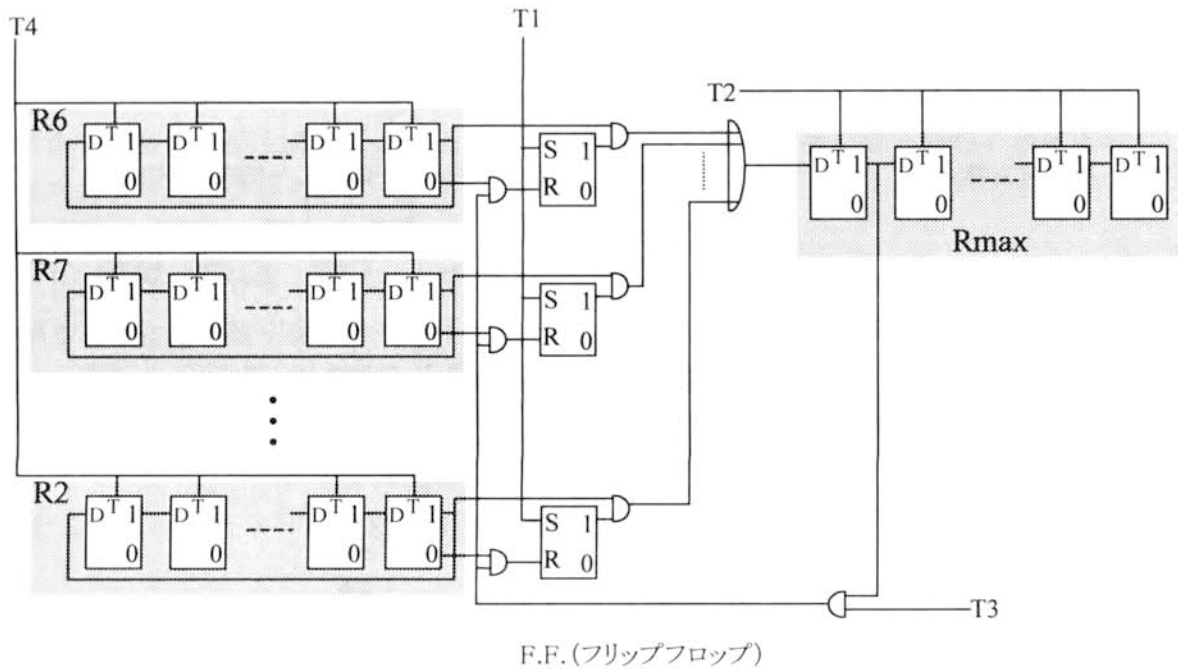


図 4.3.9 最大値検出回路

図 4.3.10 に、4 ビットで構成された R6,R7,R2 の最大値検出動作を示す。

最大値検出回路は、4 クロック T1,T2,T3,T4 で動作する。

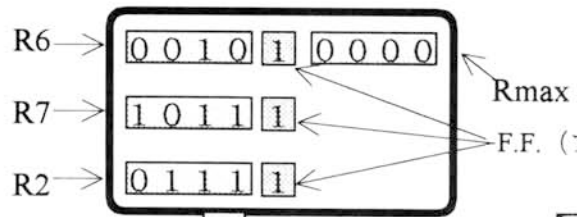
T1 はイニシャライズのクロックで R6,R7,R2 に対応した F.F. (フリップフロップ) を on にする。

T2,T3,T4 は最大値 1 ビットを計算するためのクロックであり、レジスタ Rn のビット数回繰返される。

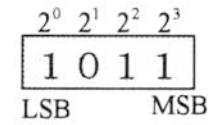
T2 では、Rmax を 1 ビットシフトするとともに、F.F. が on である対応する Rn の MSB (最上位ビット) の OR を Rmax の LSB (最下位ビット) にセットする。

T3 では、MSB が “0” である F.F. がリセットされる。T4 では、Rn を 1 ビットシフトする。

これを 4 回繰返すことにより、Rmax には Rn の最大値がセットされる。



(注) レジスタ Rn、Rmax は、右がMSB (最上位ビット) 左がLSB (最下位ビット) とする



Rmax
F.F. (フリップフロップ)

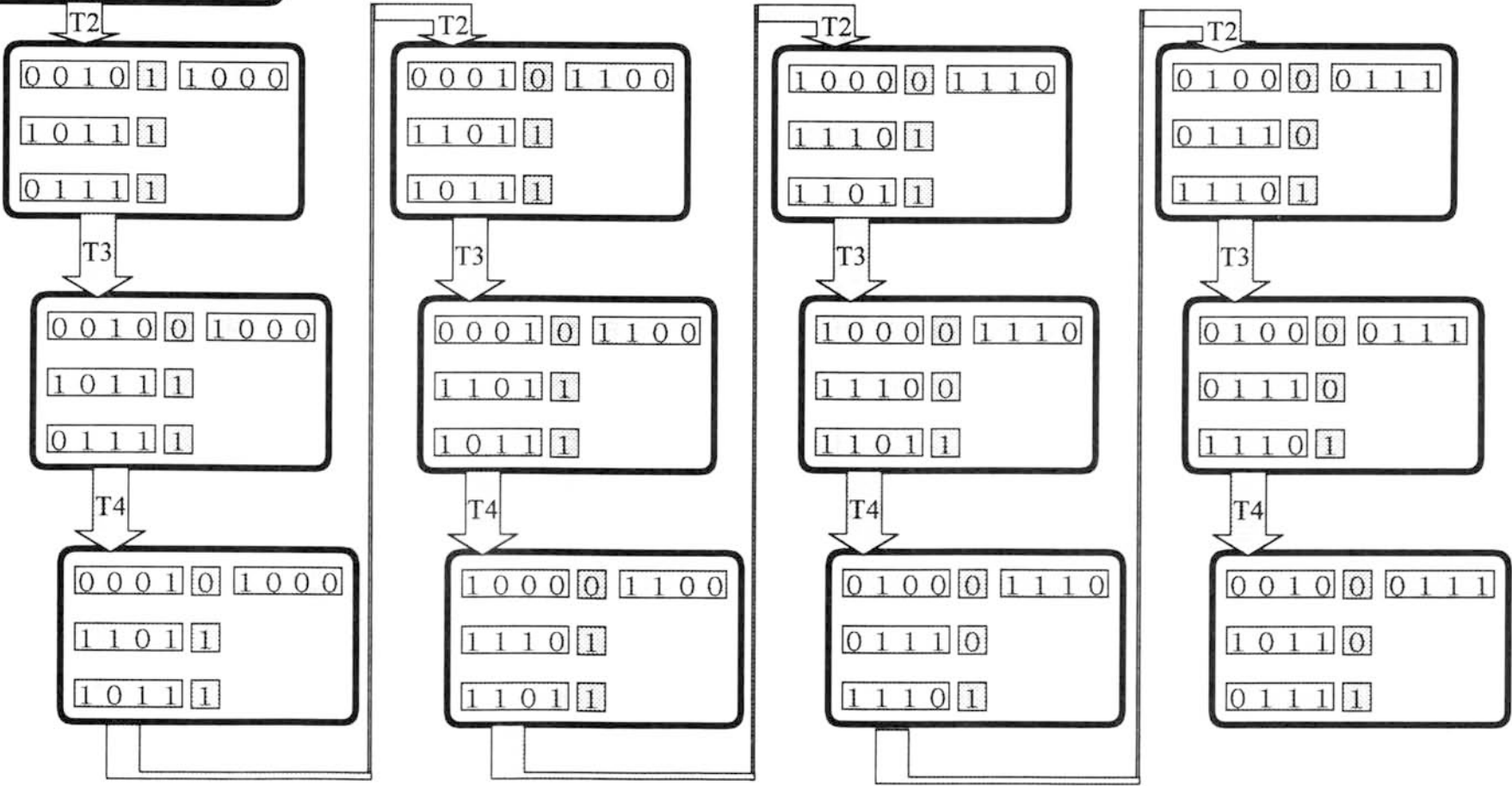


図4.3.10 最大値検出回路の動作

➤ 乗算計算

類似度の乗算計算も、図 4.3.11 に示す回路によりシフト動作で行なわれる。これは、図 3.4.9 に示すシフト処理による演算をハードウェアにて実現したものである。

なお、最大値検出回路および乗算計算回路に用いるクロックは、相関計算のクロックより速いクロックを使用し、相関計算より速く計算が終了するようにする。

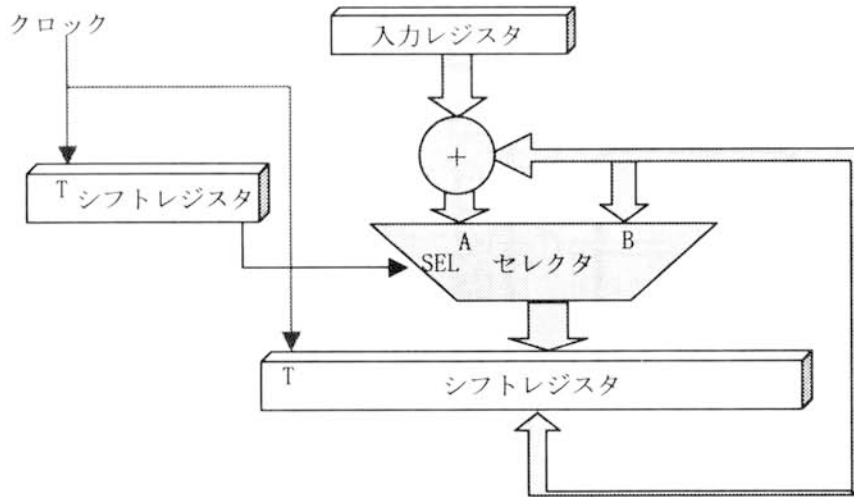


図 4.3.11 乗算計算回路

このように、全ての計算を相関計算に合わせて、シフト処理で行なうように構成した。また、相関計算よりも少ない時間で行なうことにより、図 4.3.7 に示すように相関計算時間 (=標準パタンのメモリアクセス時間) 以下で動作するよう構成した。これにより、相関計算処理は、メモリアクセス時間を限度とする認識処理時間が得られた。

【まとめ】

1,200字/秒の認識時間に制限のあるドキュメントリーダにおける、可変閾値法の相関処理は、速度制限がありかつ高速処理であることからハードウェアとし、ハードウェア化実現のため、

- 理論式に含まれている平方演算、除算を除き小数を整数化する計算アルゴリズムとし、
- ハードウェア化に当っては、
 - 9個の相関計算論理による並列処理、
 - 9個の相関値の最大値検出、
 - 類似度を求める乗算、
 - 最大類似度、次大類似度の検出、
 - をシフト処理にて実現、

全体をパイプライン処理による並行動作として、仕様認識速度を達成した。

第 5 章

非線形マッチング法における 照合論理

第5章 非線形マッチング法における照合論理

手書文字認識においては複雑な処理が多いため、ソフトウェア主体で構成されることが多く、ハードウェア化による速度向上には困難な点がある。

非線形マッチング法¹¹⁾においては、ダイナミックプログラミングにより距離を計算している。この処理は非常に計算量が多いため、実用化の課題であった。

以下では、ダイナミックプログラミングの理論式から計算アルゴリズムの開発およびこれを実現したハードウェアにつき述べる。

5.1 ダイナミックプログラミングの理論式

非線形マッチング法では、距離を計算するために、以下のようにダイナミックプログラミングを用いている。

C_s を標準パタン、 C_x を未知パタンとを、それぞれ一次元コード列で表す。

$$C_s = s_1, s_2, \dots, s_i, \dots, s_m \quad (m: \text{正の整数})$$

$$C_x = x_1, x_2, \dots, x_j, \dots, x_n \quad (n: \text{正の整数})$$

ただし、 m は C_s の一次元コード列の長さを、 n は C_x の一次元コード列の長さを表す。また、一次元コード s_i, x_j は、図 5.1.1 に示す 0~7 の方向コードとする。

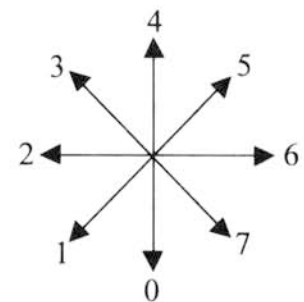


図 5.1.1 方向コード

ダイナミックプログラミングにより C_s と C_x の距離 D は、次のように表される。

$$g(1,1) = d(s_1, x_1) \quad (5.1.1)$$

$$g(i, j) = d(s_i, x_j) + \text{Min}\{g(i-1, j) + \sigma, g(i, j-1) + \sigma, [g(i-1, j-1)]\} \quad (5.1.2)$$

ただし、 $[]$ はガウス記号、 $d()$ は方向距離。

$$D(C_s, C_x) = g(m, n) \quad (5.1.3)$$

ただし、 i, j は $1 < i < m, 1 < j < n$ 内の

$$j \leq \frac{n-1}{m-1}i + \frac{\delta}{2\sqrt{2}} \frac{m+n-2}{m-1}, \quad j \geq \frac{n-1}{m-1}i - \frac{\delta}{2\sqrt{2}} \frac{m+n-2}{m-1} \quad (5.1.4)$$

を満足する値とする。なお、 δ は計算領域の幅を示す。

また、 σ は 1 つのコードが他方の 2 個以上のコードに対応づけられた場合、距離が増えるように加算される定数で 0.9 などとされる。

5.2 ダイナミックプログラミングの再現

5.1 節に述べた方式は1970年代に開発したものであり、筆者が実装置において使用されたハードウェアとして実現したものである。1999年のICDARにおいて、文字認識の世界的権威である Prof.Srihari により、類似したアイデアが発表されており、このアイデアの優秀性と再評価の可能性が示唆される。

5.3 計算アルゴリズム

5.3.1 シミュレーションプログラムの問題点

非線型マッチング法は、大型コンピュータによるシミュレーションで開発された。このときの計算方法は、図 5.3.1 の破線間の○印で示した全ての $g(i, j)$ について、式(5.1.2)の計算を、図 5.3.1(a)で示す矢印で示す順番で逐次行なっていた。このため、計算量が多大で処理速度が遅く、光学文字読取装置に組込むことが出来ないことは自明であった。

5.3.2 計算アルゴリズムの開発

計算アルゴリズムの開発にあたっては、多くの繰返し計算が必要なことから、ハードウェア化を前提に行なった。

ハードウェア化に当っては、複雑な数値演算処理を除き、並列演算の利用を考慮した計算アルゴリズムを開発、下記を実現した。

- ◆ 対角線追跡論理を開発、これにより式(5.1.4)の演算を不要化し、制御を単純化した。
- ◆ ガウス計算を単純化した。
- ◆ $g(i, j)$ の計算回路を複数持ち、並列計算を行なうことにより、最小の計算処理サイクルを実現した。

① 対角線追跡論理²¹⁾の開発

研究の過程で作成されたシミュレーションプログラムでは、全ての距離計算を逐次行なっている。これを最小サイクルで行なうため図 5.1.2(b)に示す対角線追跡論理を開発した。これは、 $g(1,1)$ から $g(m, n)$ まで δ が最小のときたどる道筋を得る計算方法である。

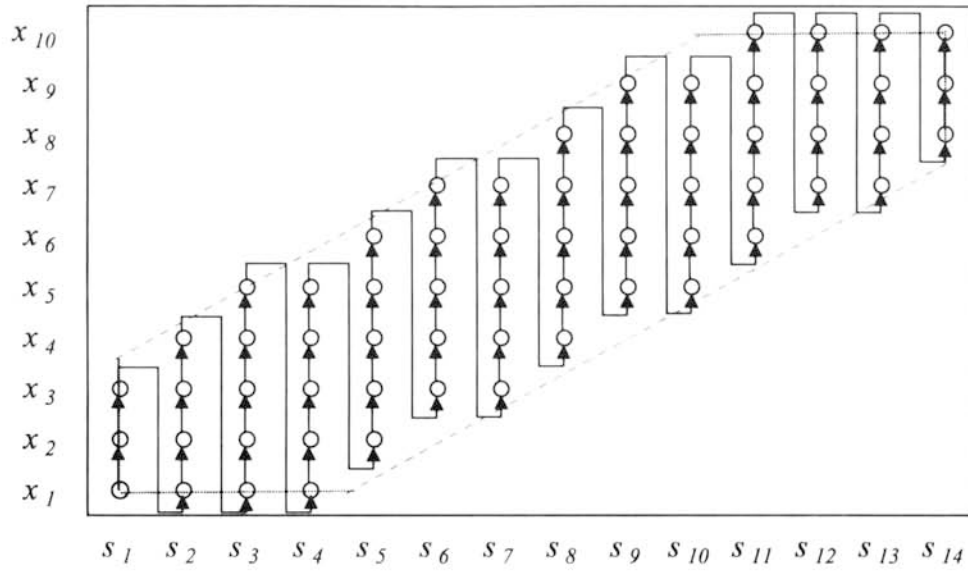
図 5.3.1(b)に示す→がこの道筋で、●が $g(i, j)$ 計算点である。図 5.3.1(b)から、計算点●が対角線に最も近傍の点であることがわかる。この道筋をたどる論理を図 5.3.2 に示す。

この対角線追跡論理により、シミュレーションでは、複雑な式(5.1.4)の計算で行なっていた δ 内外の判定が、計算不要に成り、制御が簡単になった。

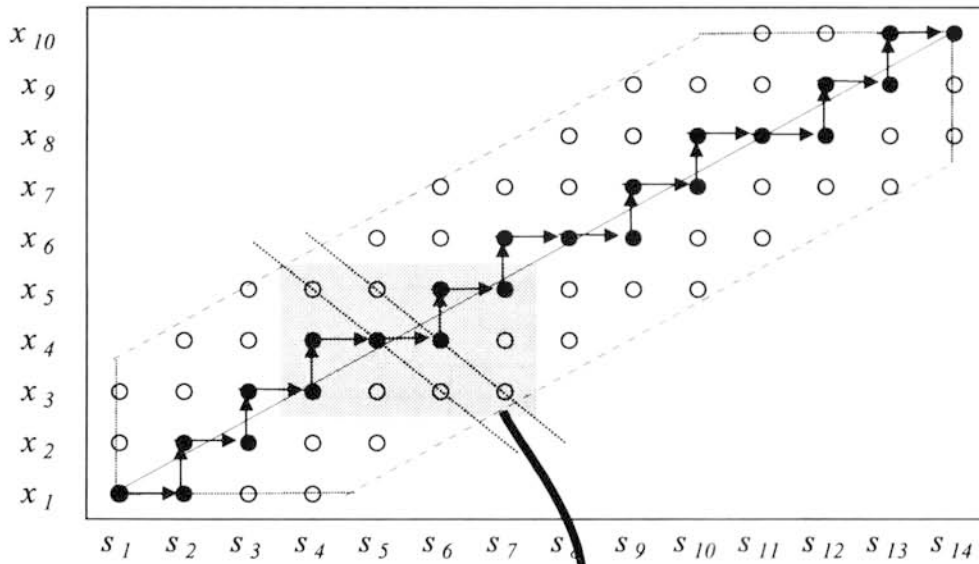
また、図 5.3.1(c)に示すように、右下がりの 45° に並ぶ B' , D' , F' は各々独立して距離の計算が可能である。これを利用して複数の距離計算回路を持ち並列計算の構造が可能となり計算サイクルの削減を実現した。

② 距離計算の単純化

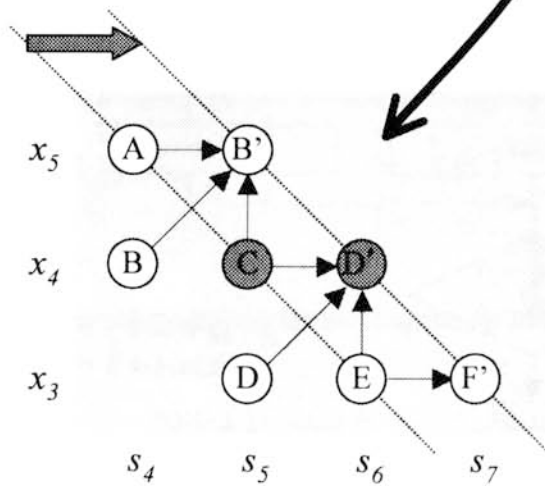
式(5.1.2)を図 5.3.1 の各点について計算するに当り、 $g(i, j)$ の構成を下記と定めた。



(a) シミュレーション



(b) 計算アルゴリズム



(c) 距離計算

図5.3.1 距離計算方法

$$g(i, j) \quad (10 \text{ ビット}) = \infty \quad (1 \text{ ビット}) \\ + \text{ 距離 (8 ビット (0~255))} \\ + \sigma \quad (1 \text{ ビット})$$

ここで、

- ◆ ∞ (1ビット) は、 δ 領域外および距離が 255 を超えたことを示す
- ◆ 距離 (8ビット) は、方向距離 $d(s_i, x_j)$ が 0~4 であることから、距離が 255 を超えたときには、対象外のカテゴリと判断できるとのシミュレーション結果から定めた。
- ◆ σ (1ビット) は、シミュレーションでは、0.9 であり、これは一つのコードに他のコード 2~10 個が対応するとき 1~9 の値をとることから、この範囲では等価な値をとるよう下記の論理により 1ビットとした。

式 (5.1.2) の Min 計算で

$g(i-1, j)$ または $g(i, j-1)$ が採用されたときで、
 その σ が “1” のとき、距離 = 距離 + 1 および $\sigma = “1”$
 その σ が “0” のとき、 $\sigma = “1”$
 $g(i-1, j-1)$ が採用されたとき、 $\sigma = “0”$ とする。

この計算アルゴリズムにより σ および ∞ の計算が単純化された。またガウス計算が不要となり整数演算が可能となった。

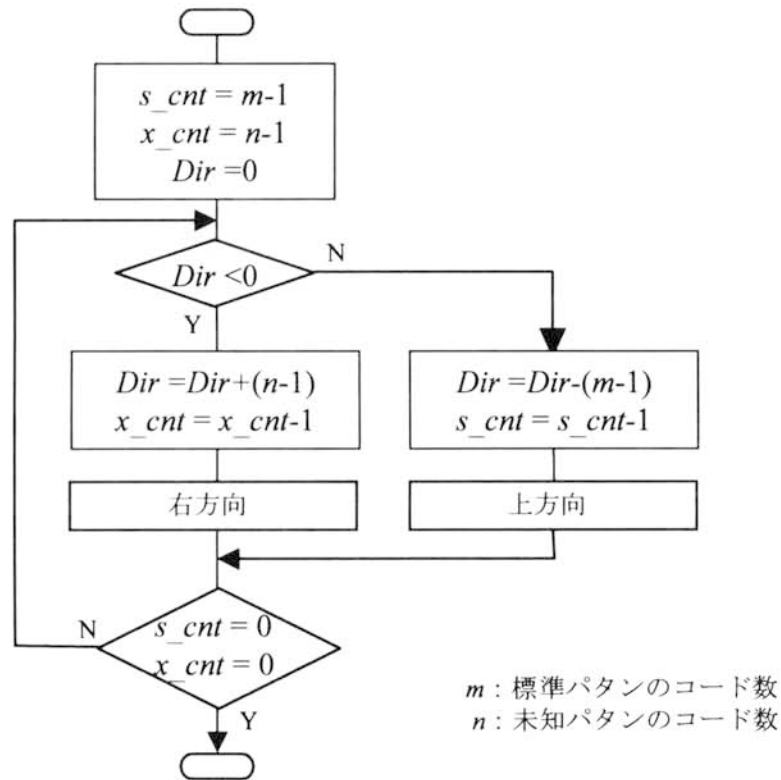


図 5.3.2 対角線追跡論理

以上の計算アルゴリズムは、現実的な範囲で理論式と等価である。

5.4 ハードウェアの構成

図 5.4.1 にハードウェア論理構成²²⁾を示す。この構成の特徴は、距離計算回路を、図 5.3.2(c)に示すA、C、Eのような右下がりの45°に並べ並列に距離を計算する。これを対角線追跡論理で決定される回数実施し、 $D(Cs, Cx)$ を計算する。

5.4.1 構成

◆ 距離格納レジスタ G_reg は、距離計算回路の出力 $g(i, j)$ を格納するレジスタで、図

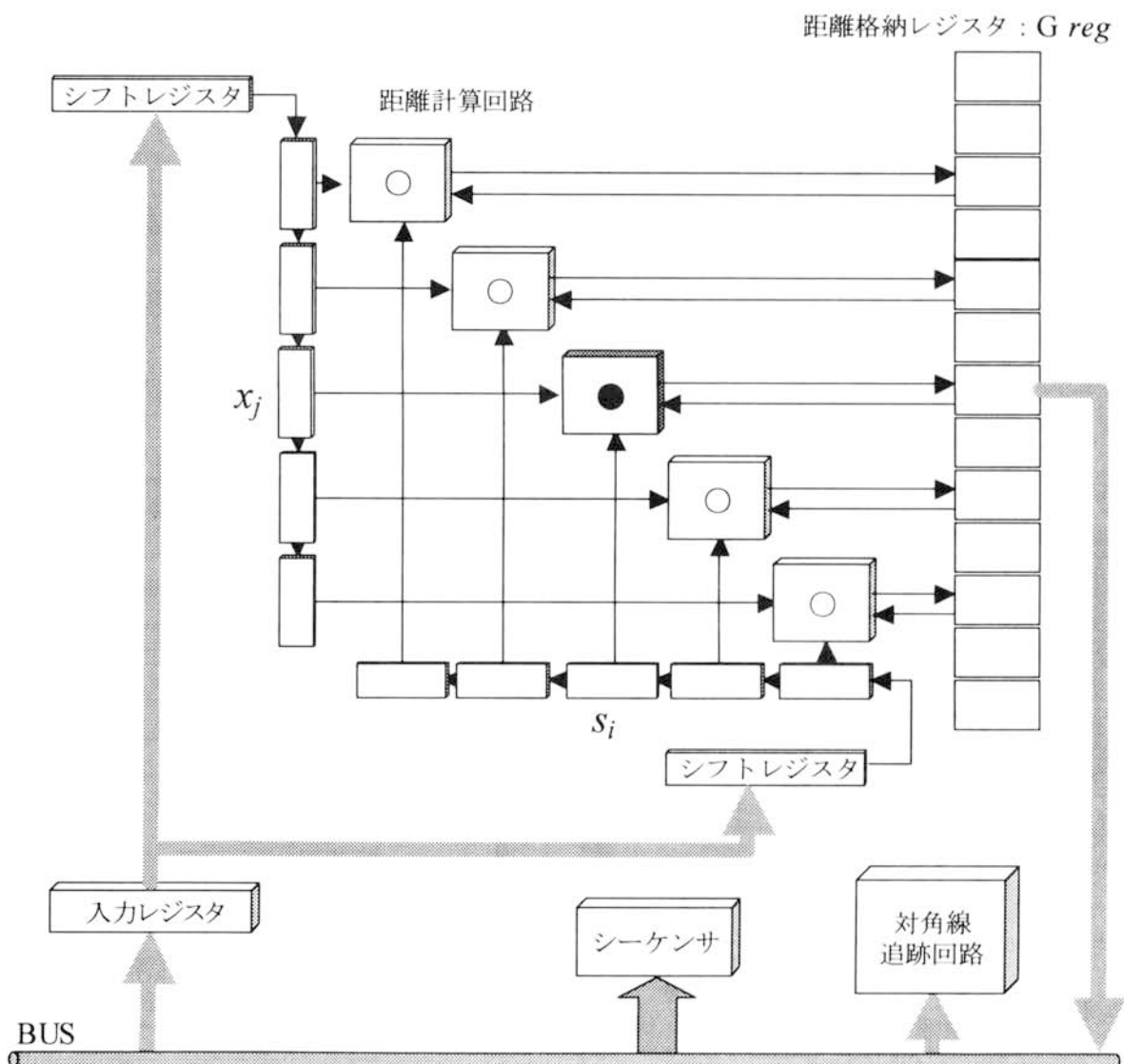


図 5.4.1 非線形マッチング法の照合部の構成

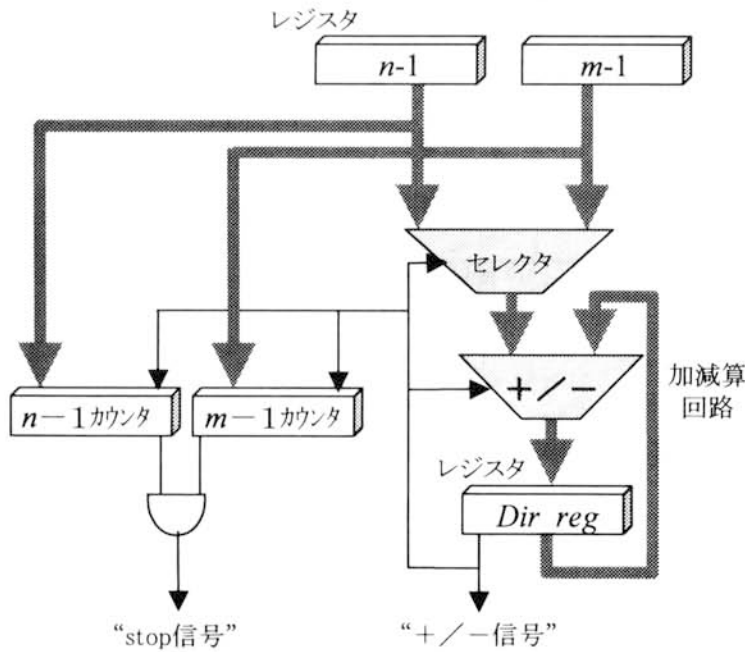
5.3.1 の●および○にあたる。各レジスタは、10ビットで構成され、ワークとして、上下2レジスタ追加されている。

- ◆ 距離計算回路は、 δ によって決定される数より構成され、それぞれ対応する s_i, x_j および G_reg と結合されている。図 5.4.1 においては、距離計算回路が 5 個の例を示す。
- ◆ 対角線追跡回路は、対角線を追跡し進むべき右または上方向を指示し、また計算の終了を指示する。
- ◆ シーケンサは、各論理のタイミングパルスを発生し、またパイプライン動作を可能とする。

① 対角線追跡回路

方向を指示する“+/-信号”および終了を指示する“stop信号”を計算する論理部である。

図 5.4.2 に、論理構成および図 5.3.1(b)に対応する $n=10, m=14$ の場合の動作を示す。方向を示す Dir_reg が正か負かで照合論理部のすべての動作が決定する。すなわち、対角線追跡が右方向か上方向かの計算が行なわれ、また対角線追跡回路では、対応するカウンタはデクリメントされ、 Dir_reg には、 $(n-1)$ または $(m-1)$ が加算あるいは減算される。



$n-1=9, m-1=13$ の場合

Dir_reg	移動方向	$n-1$ カウンタ	$m-1$ カウンタ
0		9	13
9	右		12
-4	上	8	
5	右		11
-8	上	7	
1	右		10
-12	上	6	
-3	右		9
6	右		8
-7	上	5	
2	右		7
-11	上	4	
-2	右		6
7	右		5
-6	上	3	
3	右		4
-10	上	2	
-1	右		3
8	右		2
-5	上	1	
4	右		1
-9	上	0	
0	右		0

図 5.4.2 対角線追跡回路

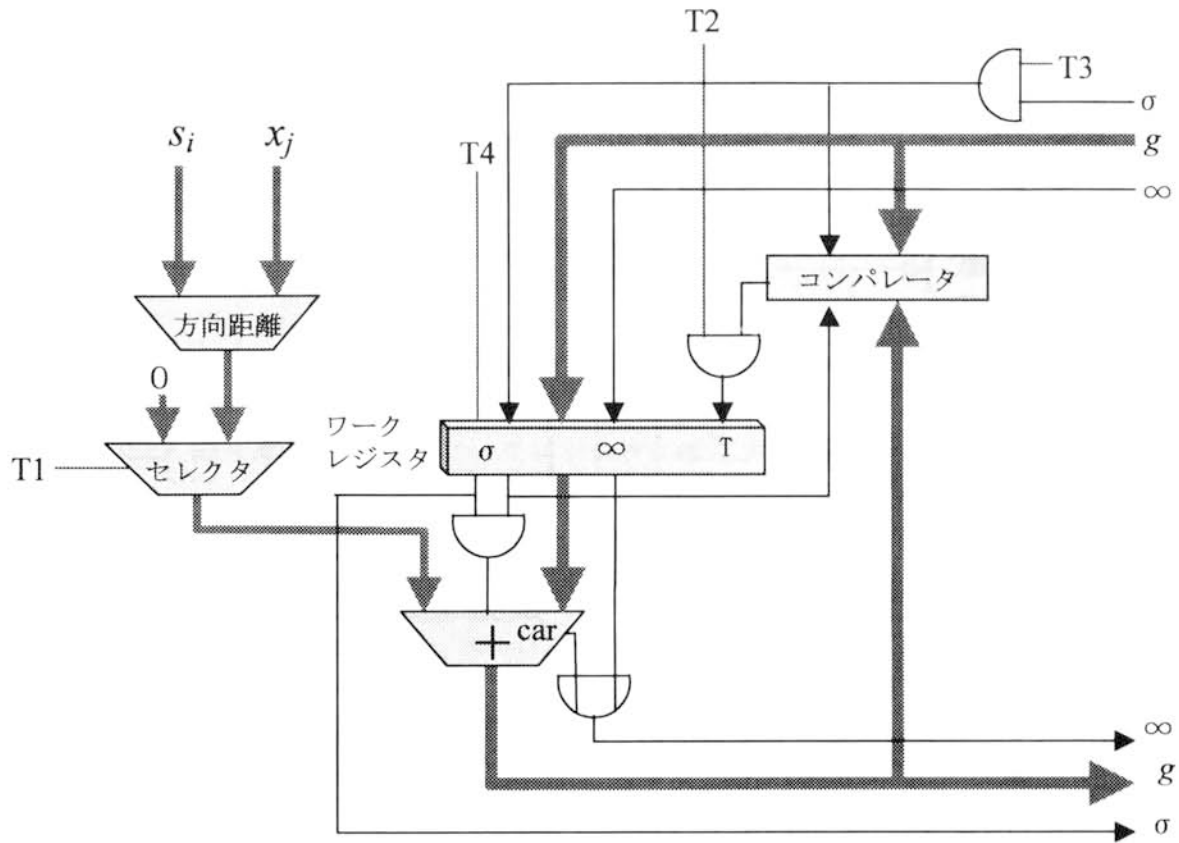


図 5.4.3 距離 $g(s_i, x_j)$ 計算回路

2つのカウンタが0となったときに“stop 信号”がでて照合演算は終了する。

② 距離計算回路

式(5.1.2)を計算する1つの距離計算回路の構成を図5.4.3に、1サイクルのタイミングチャートを図5.4.4に示す。

距離計算回路は距離 g の計算のためのワークレジスタ、Min 値を指示するコンパレータ、方向距離計算²⁷⁾、加算器などからなり、T1~T4 のタイミング信号で計算が行なわれる。また、T5 と Greg モード信号は、Greg のシフトおよびロー

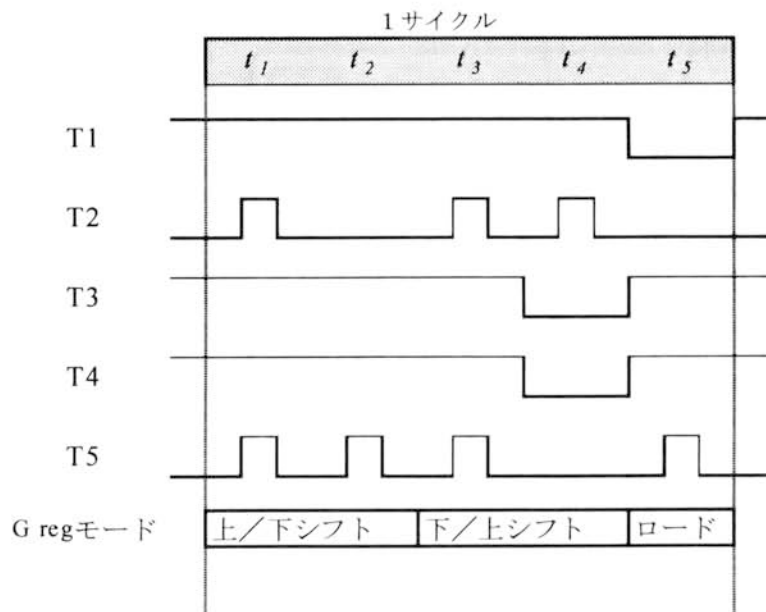


図 5.4.4 1サイクルのタイムチャート

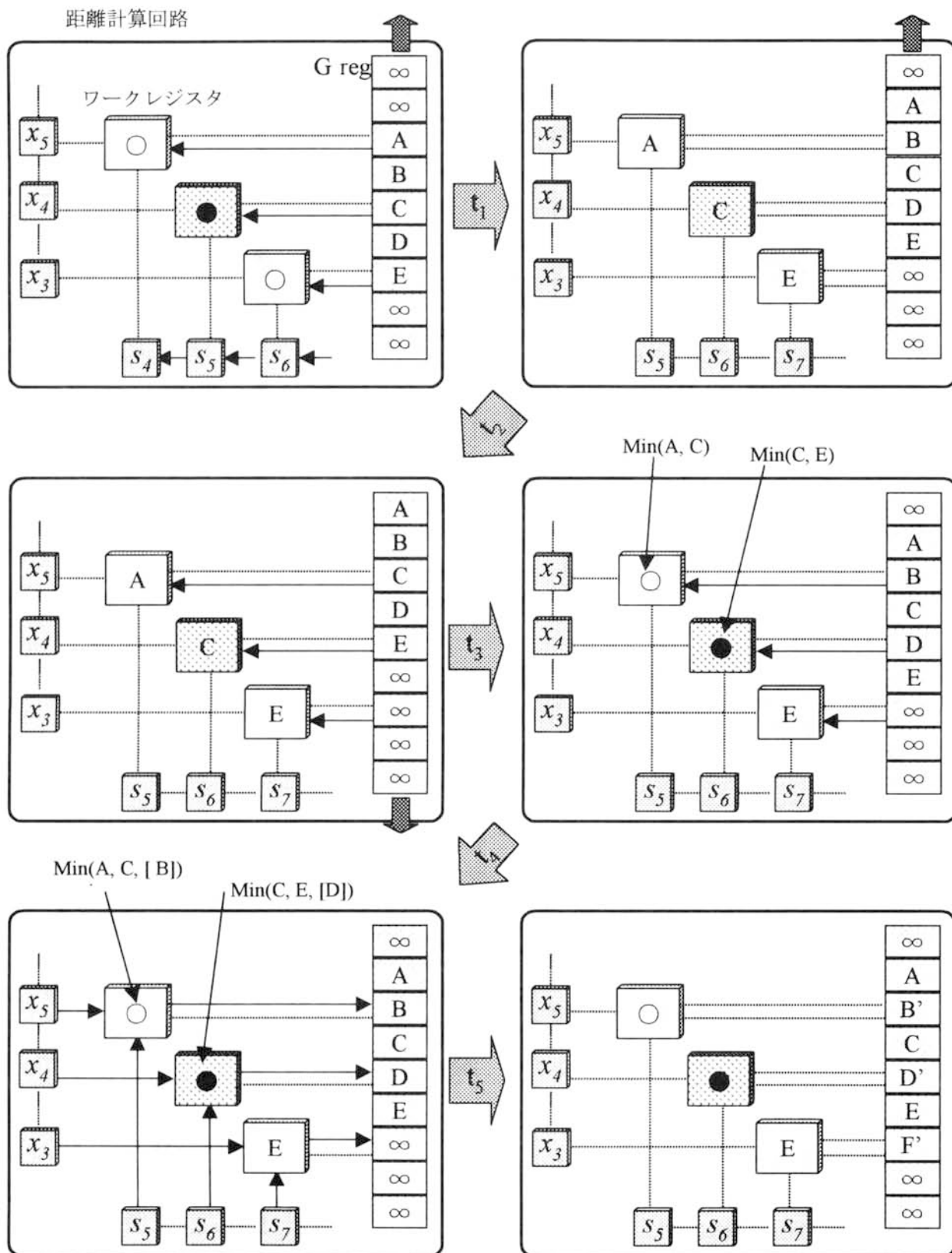


図5.45 1サイクルの動作

$$B' = \text{Min}(A, C, [B]) + d(s_5, x_5)$$

$$D' = \text{Min}(C, E, [D]) + d(s_6, x_4)$$

$$F' = \text{Min}(E, \infty, \infty) + d(s_7, x_3)$$

ドセットを制御する。

距離の計算は $t1 \sim t5$ の 5 クロックで行なわれ、これを 1 サイクルとする。

5.4.2 動作

図 5.3.1(c)に対応する距離計算動作を、図 5.4.5 に示す。構成は 3 個の距離計算回路で、A,B,C,D,E の各距離から右方向に移動計算し、 B' 、 C 、 D' 、 E 、 F' に移る 1 サイクルを示す。

- ▶ 初期状態は、 s_i として s_4, s_5, s_6 、 x_j として x_3, x_4, x_5 、 $G\text{ reg}$ には距離 A,B,C,D, E が格納されている。
- ▶ $t1$ では、各距離計算回路のワークレジスタに対応する $G\text{ reg}$ の値 A,C,E がセットされ、 $G\text{ reg}$ は上方にシフトする。また、 s_i もシフトし、 s_5, s_6, s_7 となる。
- ▶ $t2$ では、 $G\text{ reg}$ がさらに上方にシフトする。
- ▶ $t3$ では、ワークレジスタと $G\text{ reg}$ の Min 値がワークレジスタにセットされる。すなわち、 $\text{Min}(A,C)$ 、 $\text{Min}(C,E)$ 、 $\text{Min}(E,\infty)=E$ がセットされる。また、 $G\text{ reg}$ は下方にシフトされる。
- ▶ $t4$ では、ワークレジスタと $G\text{ reg}$ の Min 値がワークレジスタにセットされる。すなわち、 $\text{Min}(A,C,[B])$ 、 $\text{Min}(C,E,[D])$ 、 $\text{Min}(E,\infty,\infty)=E$ がセットされる。
- ▶ $t5$ では、ワークレジスタに方向距離が加えられて、 $G\text{ reg}$ に格納され、1 サイクルの演算が完結する。すなわち、

$$B' = \text{Min}(A,C,[B]) + d(s_5, x_5)$$

$$D' = \text{Min}(C,E,[D]) + d(s_6, x_4)$$

$$F' = \text{Min}(E,\infty,\infty) + d(s_7, x_3) = E + d(s_7, x_3)$$

がそれぞれ $G\text{ reg}$ にセットされる。

[まとめ]

手書文字認識においては、複雑な処理が多いためハードウェア化による速度向上には困難な点がある。しかし、非線型マッチング法ではダイナミックプログラミングによる照合計算の速度向上が課題であった。これには、

- ▶ 計算アルゴリズムでは、対角線追跡論理により計算ステップ数の削減および単純化を達成し、
- ▶ 距離計算単位を複数化したハードウェア論理による並列処理、対角線追跡論理の回路化によりハードウェア化し、
- ▶ 全体をパイプライン処理として

実用化に成功した。

第 6 章

輪郭線追跡処理のソフトウェア化

第6章 輪郭線追跡処理のソフトウェア化

数字、英字、カタカナなどの手書文字認識に適用された図形検証法は、細め処理による詳細特徴情報の欠落による認識精度の限界を超えて、より高精度の認識を達成するために研究された認識方式である。一律な処理である細め処理を無くしたため、前処理が減少したこと、処理の大半が数値演算であること、およびLSI、ICメモリの高性能化、低価格化などから、光学文字読取装置のアーキテクチャはソフトウェアを主体に構成された。

しかしながら、ソフトウェアによる認識処理では認識速度向上が課題となる。この課題を計算アルゴリズムおよびハードウェアとの最適化の研究により速度向上を図り、光学文字読取装置に適用した。

以下では、ソフトウェアが不得意とする輪郭線のチェインコード化処理の計算アルゴリズムおよびハードウェアにより強化されたプロセッサにつき報告する。

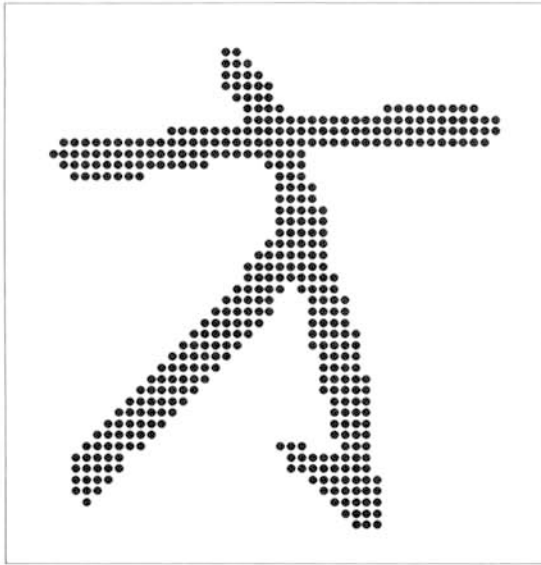
6.1 モデルパターン作成

ソフトウェアで認識処理を実施する場合には、認識速度が課題となる。ソフトウェアでの処理は認識すべき文字画像の形状により大きく左右される。このため、研究開発に当り認識時間試算・評価用のモデルパターンを作成した。

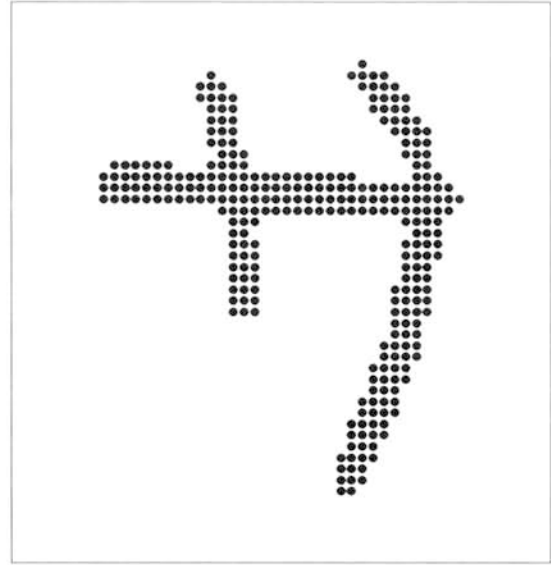
図 6.1.1 にモデルパターン作成のために用いた 4 個の手書カタカナのサンプルパターンを示す。このサンプルパターンは成分数、輪郭数、外および内輪郭を持つなどの特徴から選んだパターンである。

モデルパターンは、このサンプルパターンから図形検証法の処理に使用する各特徴を算出し平均化して求めた。この結果を図 6.1.2 に示す。ここで、各特徴項目を、図 6.1.2 により説明する。

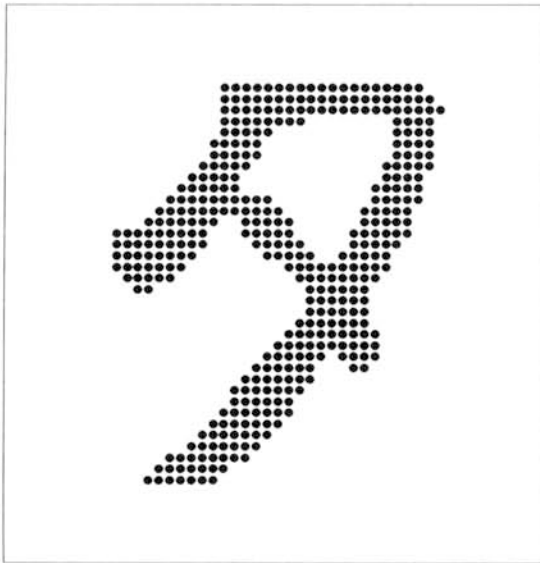
- 成分数：独立した文字線の塊で PAT 4. では 3 成分
- 輪郭数：文字線の輪郭線の数で、ループが存在するときは、ループ内も 1 本の輪郭線を構成する。PAT 3. では 2 輪郭
- 輪郭点数：輪郭線を構成する黒画素の数
- p q 点数：次節で詳述する点で、水平方向に見て白画素から黒画素に変化する黒画素（p 点）あるいは黒画素から白画素に変化する黒画素（q 点）を 1 点とする。なお、白黒白と変化する場合は 1 個の黒画素でも p 点 q 点各 1 点、計 2 点となる。
- 外点：p q 点のうち、各水平方向において、最左端 p 点および最右端 q 点の数。
- 1 内点：p q 点のうち、各水平方向において、最左端の p 点より 1 番目の p 点および最右端の q 点より 1 番目の q 点の数
- 2 内点：p q 入力点のうち、各水平方向において、最左端の p 点より 2 番目の p 点および最右端の q 点より 2 番目の q 点の数
- 輪郭線数：LD 表現で表した輪郭線数



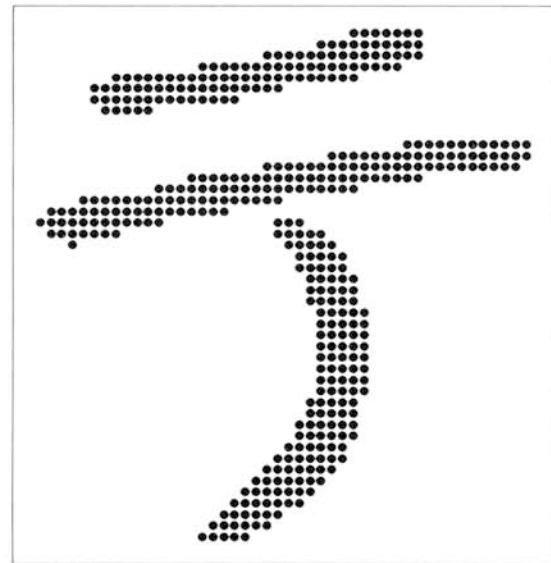
PAT 1



PAT 2



PAT 3



PAT 4

図6.1.1 サンプルパタン

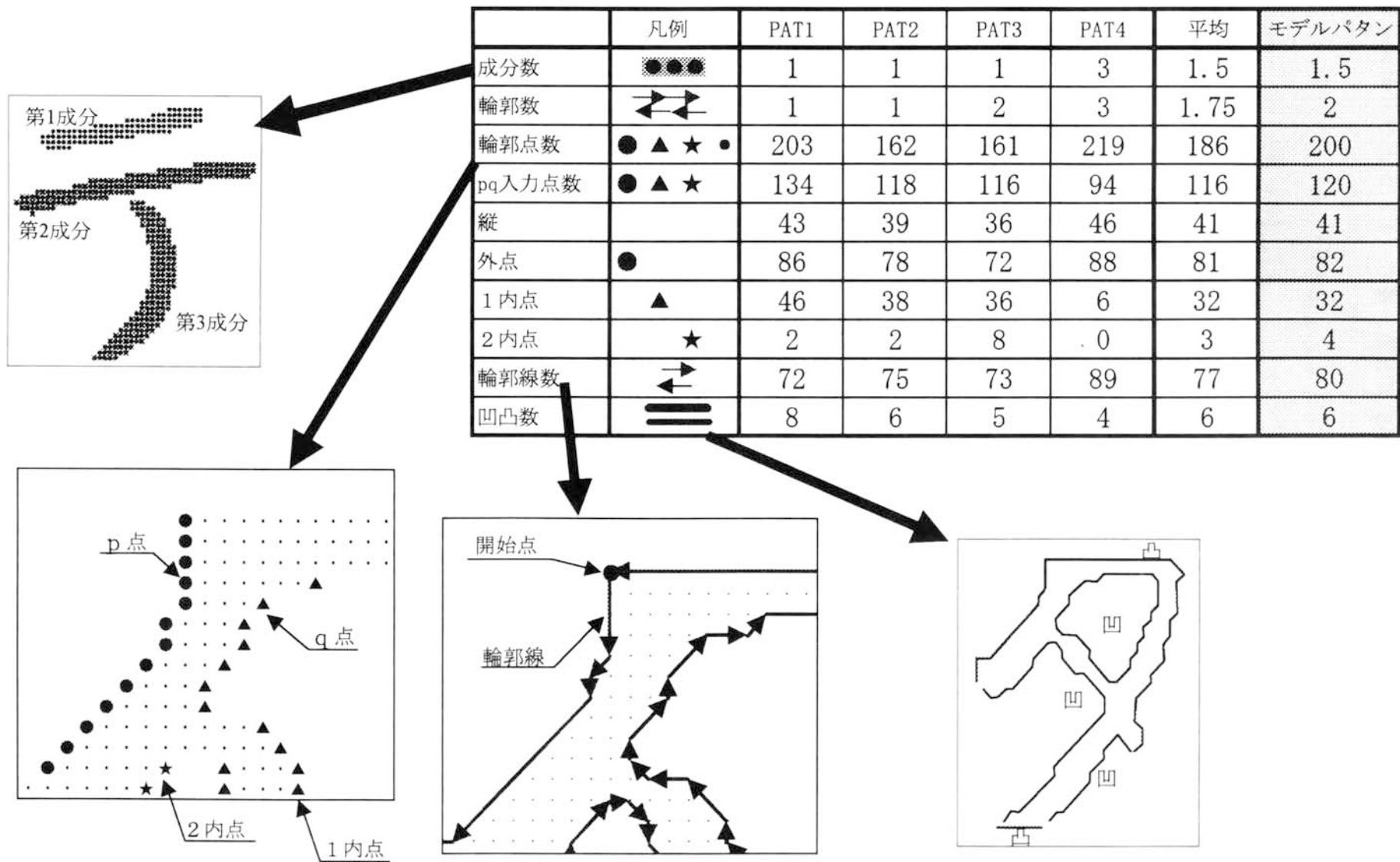


図6.1.2 モデルパターン

LD表現は、輪郭線を同一の方向コードでまとめ、長さ(L)および方向コード(D)で表現する形式である。

例：チェーンコード： 3 3 4 4 4 4 4 5 5 5

LD表現： (2, 3)、(5, 4)、(3, 5)

- 凹凸数：図形検証法の主要な特徴で、輪郭線を部分形状により分割する。詳細な多くの特徴を計算し、辞書と照合され距離を算出される。

6.2 輪郭線追跡処理

本節では、輪郭線追跡処理につき、計算アルゴリズムおよびハードウェアにより強化されたプロセッサの内容を報告する。

6.2.1 輪郭線座標表現形式

2次元文字画像の輪郭線のチェーンコード化は、通常2次元マスク処理により行なわれる。しかしながら2次元マスク処理は、ソフトウェアでは不向きなビット処理を多くの回数繰返すことになるため、処理時間が多大となる。ソフトウェアで効率的に行なうために、ソフトウェア処理に向けた新たな2次元画像表現形式を開発した。これを、輪郭点座標表現と仮称する。

(i) 輪郭点座標表現の考え方

輪郭点座標表現と輪郭線追跡との考え方を比喩的に表すと、山に囲まれた池(白地)に島(文字画像)を浮かべ、この島の水辺(輪郭線)を辿る(追跡)ことである。

(ii) 輪郭点座標表現の形式

輪郭線座標表現形式の規則を図6.2.1の例で述べる。

図6.2.1では、文字画像の横幅が253ビット以下の例を示す。

- 入力文字パターンは、 $1 < X < 254$ 、 $0 < Y$ の位置とする。
- Yは、文字パタンの高さ+2以上とする。
- 全てのYに対し、 $X = 0 = (00)_H$ にq点、 $X = 255 = (FF)_H$ にp点を生成する。
- 左端よりX方向、Y方向の順にp q点の座標および属性を各1 byteで順に表にする。各点の属性の初期値として、 $X=0$ および255のp q点は $(FF)_H$ 、他の文字を表すp q点は $(00)_H$ とする。
- p q表のアドレスポインタ表を作成する。アドレスポインタは各Yに対し、最初のp点、および最後のq点のp q表のアドレスとする。

(iii) 輪郭点座標表現形式の特徴

輪郭点座標表現形式は、ソフトウェア処理に適した下記の特徴を持つ。

- ◆ 画像が数値で表現されている
⇒ソフトウェア処理に適した表現
- ◆ 2次元マスク処理に比べ、繰返し処理が大幅減少
⇒処理速度向上
- ◆ 左右および上下対象の構造である
⇒左右および上下対象のフローチャートが使用できる。
- ◆ 水平方向は圧縮されている
⇒水平方向は、輪郭点追跡が不要
- ◆ パターンは白点で囲まれている
⇒輪郭点繰返し処理での輪郭線の終了判定が軽減
- ◆ 水平の区切りは黒点で判定
⇒輪郭点繰返し処理での水平方向の範囲判定が軽減

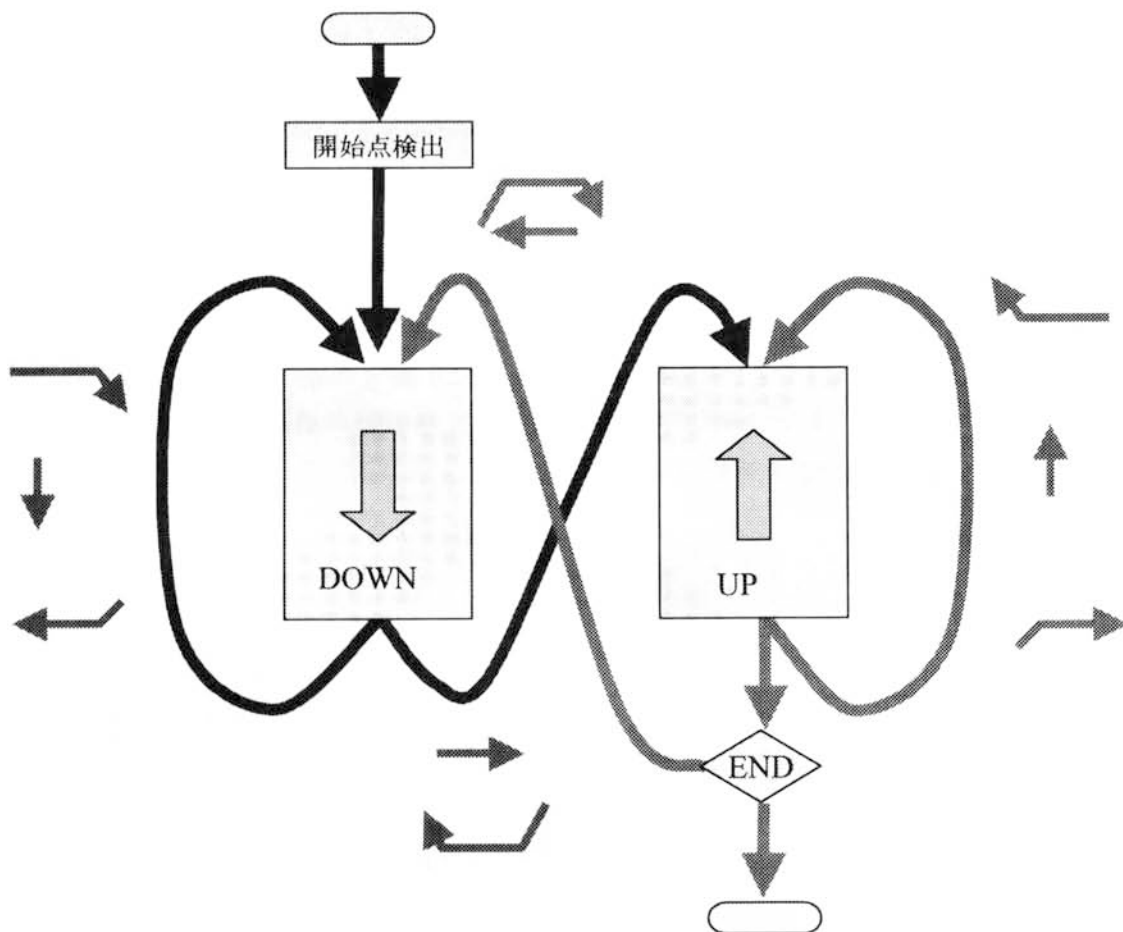


図 6.2.2 輪郭線追跡フローチャート

6.2.2 輪郭線追跡フローチャート

輪郭線追跡の概略フローチャートを、図 6.2.2 に示す。

開始時に追跡すべき輪郭線の開始点を探す。開始点は、最上左端の未処理の p 点とする。これは、p q 表においては、最小アドレスの属性 (0 0) H の p 点である。検出は高速・容易に可能であり、ここにも輪郭点座標表現形式の優れている特質が見られる。

この p 点を起点として、左回りに輪郭線を追跡する。

追跡は p 点を下方に追跡し(DOWN)、上方に転換する場合は、上方の q 点追跡に移る(U P)。追跡中に L D 形式で輪郭線データを作成すると共に、各 p q 点の属性 byte に属性を書きこむ。

このデータは輪郭線の追跡終了判定にも利用される。すなわち、q 点追跡から p 点追跡に移るときに属性を調べ、書き込まれていれば、開始点に戻ってきたため、輪郭線追跡終了が判定される。

6.2.3 単一動作プロセッサのアーキテクチャと輪郭線追跡フローチャート

ソフトウェア処理での処理時間試算のため、動作させるプロセッサを図 6.2.3(a)に示す。機能は以下とする。

- 多くの繰返し処理が想定されるため、サイクルタイムの短縮できるマイクロプログラムが動作するプロセッサとする。
- サイクルタイムは全ての命令で一定とする⇒処理時間はダイナミックステップ数で試算できる。
- 汎用レジスタ Rn は 16 個
- 1 word は 2 byte
- メモリはアドレスレジスタ S でアドレスを指定される。

このプロセッサにより、図 6.2.2 の Down 追跡中心部のフローチャートを図 6.2.4 に示す。なお、Up 追跡のフローチャートは輪郭点座標表現形式の構造が上下対称となっているため、同様のフローチャートとなる。

また、図 6.2.4 において、

- M(S)→R3 は、アドレスレジスタ S に格納されているアドレスで示されるメモリのデータを汎用レジスタ R3 に読み出す動作
- アドレスを格納する汎用レジスタを説明の都合上 S3、S4 と表記
- Comp(R3,R4)は、汎用レジスタ R3 に格納されている値と、汎用レジスタ R4 に格納されている値とを比較する動作
- p q 点の座標および属性は、図 6.2.1 では 1 byte としたが、図 6.2.4 では説明の都合上各 1 word とする。



である。

図 6.2.4 より、処理速度を決定するダイナミックステップ数の試算を行なう。

モデルパターンから



$$\text{ダイナミックステップ数} = (17+24) \times 120/2 + 11 \times 32 + 22 \times 4 = 2,900 \quad (6.2.1)$$

ここで、図 6.2.4 のフローチャートより

 : 17ステップ
 : 24ステップ
 1内点 : 11ステップ
 2内点 : 11×2ステップ

また、図 6.1.2 モデルパタンより

入力点数 = 120、1内点 = 32、2内点 = 4

ただし、垂直の上下線分は試算値に余裕をみることとし、 あるいは  としている。

なお、このアルゴリズムに類似した手法は宮武ら³⁰⁾によって独立に発見されているが、本論文で述べた手法は学会発表されていないのが残念である。

6.2.4 プロセッサ機能のハードウェアによる強化

図 6.2.4 をベースに、繰り返し処理のダイナミックステップ数の削減を図った。削減には下記の4点でハードウェアの増強を行なった。なお、この内容は、本項で詳述する輪郭線追跡処理のみでなく、認識処理の全体で同様な試算を行なった結果により採用されたものである。

(7) メモリアドレスレジスタ S 複数化

メモリアドレスレジスタ S の個数を 16 個に増し、メモリアクセスのステップ数を削減する。

図形検証法においては、多くの処理が複数のテーブルを同時に参照しながら計算することが処理の特徴である。このため、メモリアドレスレジスタを複数化することは、本質的にダイナミックステップ数削減に大きな効果があった。

図 6.2.4 においては、S3→S および S4→S の命令が削減される。

(i) 汎用レジスタ Rn の byte 演算サポート

2 byte からなる汎用レジスタを上位、下位 byte の独立同時演算を可能とした。すなわち、16 個の汎用レジスタを 16 個の 1word レジスタとしても 32 個の 1 byte レジスタとしても利用可能とし、かつ演算器 (ALU) により、上位下位同時演算を可能とした。

例 : R0u+R1u→R0u および R0L+R1L→R0L の 2 動作を 1 ステップで行なう。

(Rnu、RnL はそれぞれ Rn の上位および下位 byte とする)

これにより、word でアクセスするメモリのアクセス数削減および演算ステップ数の削減が図られる。

図 6.2.4 においては、p q 点の座標および属性を 1 byte づつとし、図 6.2.1 に示すように 1 word に格納できるため、属性書込み時のメモリアドレスの更新命令が削減される。

(ウ) マルチ Jump 命令のサポート

比較命令でテストし、分岐を行なう場合通常は 2 方向 (Yes で分岐、No で継続) であるが同時に多方向に分岐可能とする。例えば、比較命令または演算器での演算の結果、> 0、

= 0、< 0により、3方向に分岐する。

これにより、繰り返し処理中の分岐に要するステップ数が削減される。

図 6.2.4 においては、2 ステップで行なわれているテスト J u m p が 1 ステップで可能となる。

(エ) 並行処理

1 ステップでのプロセッサの動作を複数同時に実行する。

例： $M(S4)_u + R1_u \rightarrow Qu, M(S4)_L + R1_L \rightarrow Q_L, S4 + 1 \rightarrow S4$

アドレスレジスタ S4 で指定されたアドレスのメモリの内容と、汎用レジスタ R0 の各々の対応する上位および下位 byte の和を求め、かつアドレスレジスタ S4 をインクリメントする。

なお、テスト J u m p 動作は複雑になる。すなわち、分岐命令で分岐先に関係なく演算を実行してしまうため、注意が必要となる。動作例および使用法を図 6.2.5 に示す。

図 6.2.5(a)に示す処理は、(b)のような組換えにより 1 ステップ削減される。ただし、(c)

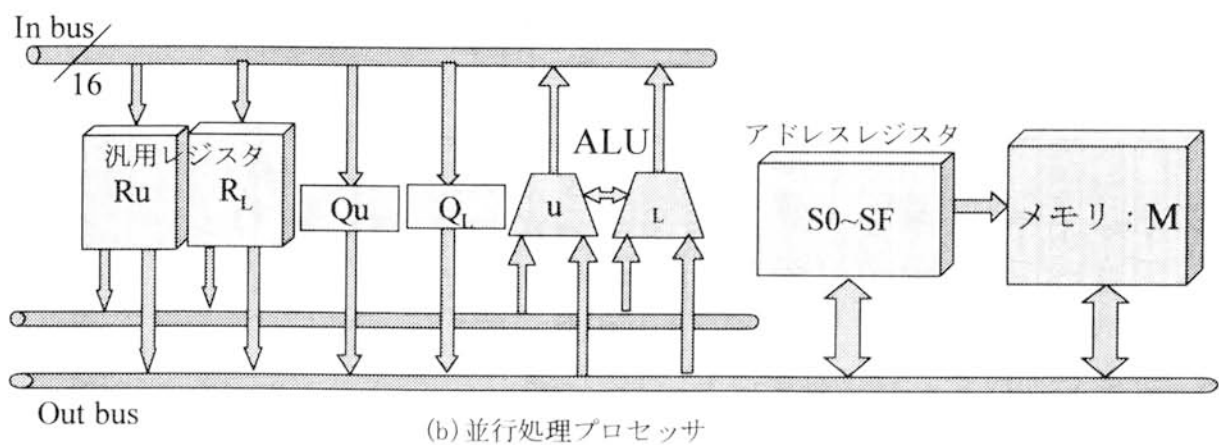
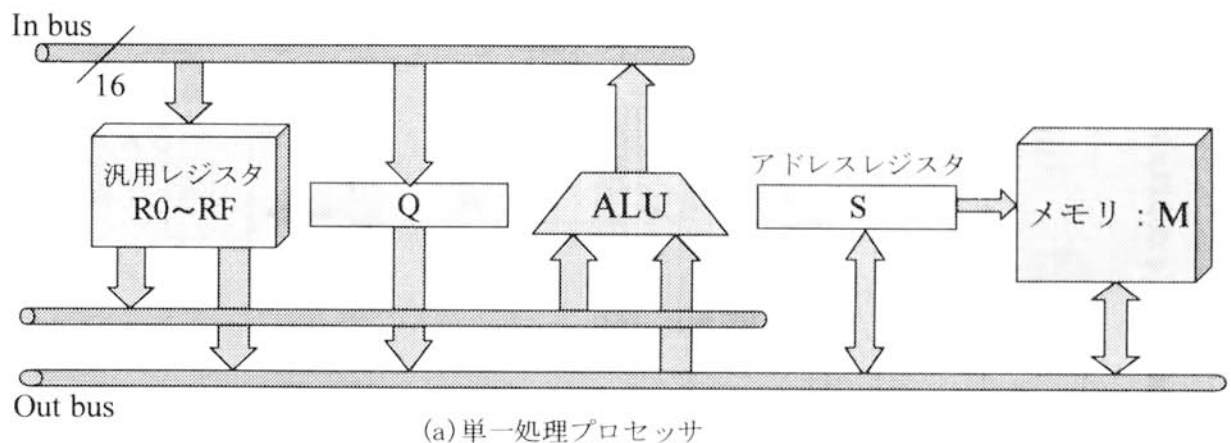


図 6.2.3 プロセッサアーキテクチャ

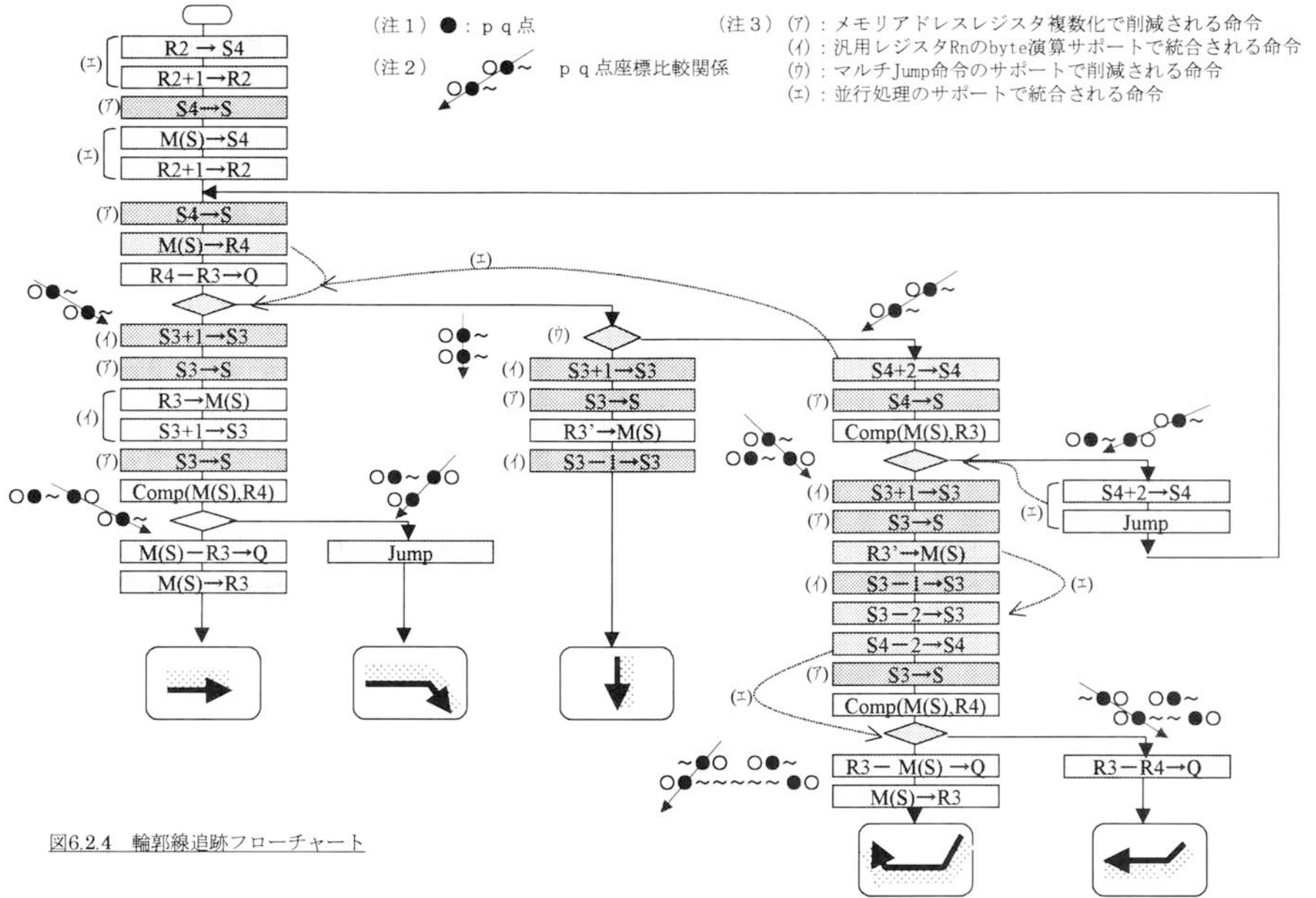
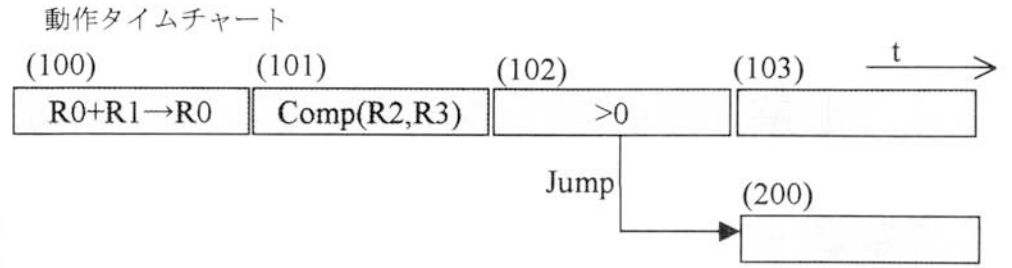
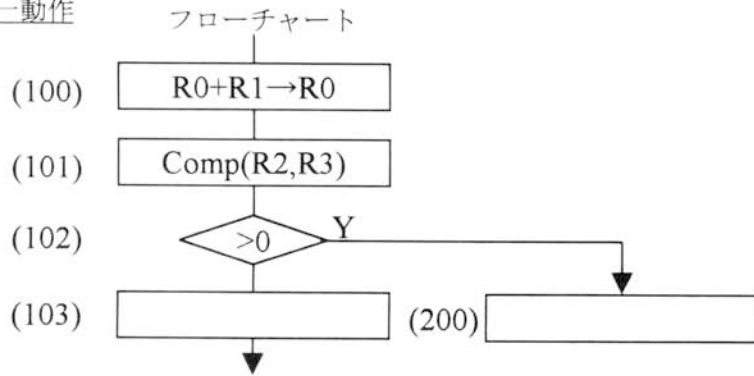
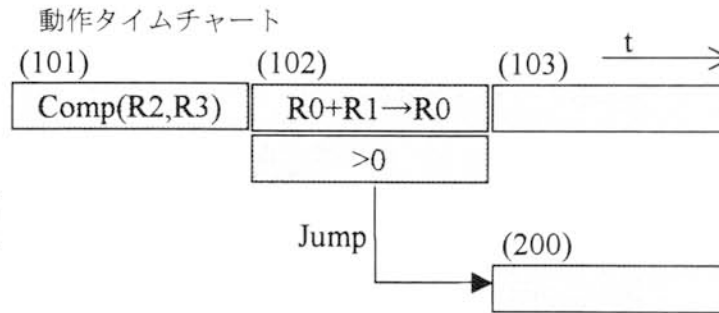
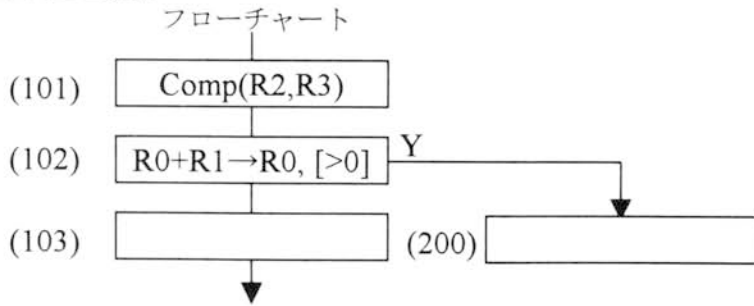


図6.2.4 輪郭線追跡フローチャート

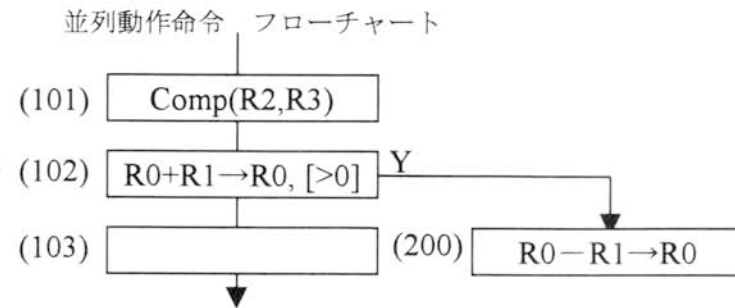
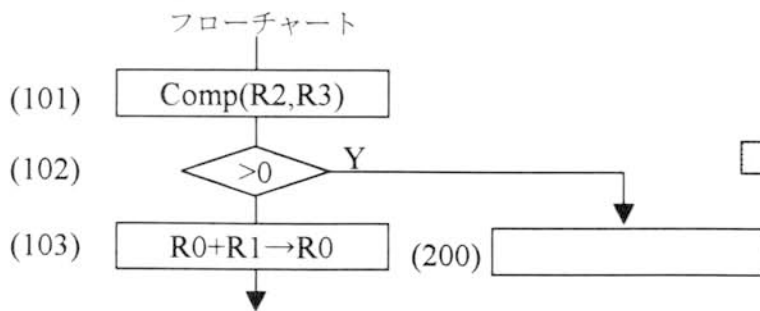
(a) 単一動作



(b) 並列命令 動作



(c) 並列命令 注意点



(注1) (100) : メモリアドレス

(注2) [>0] : テスト命令

図6.2.5 並列動作命令

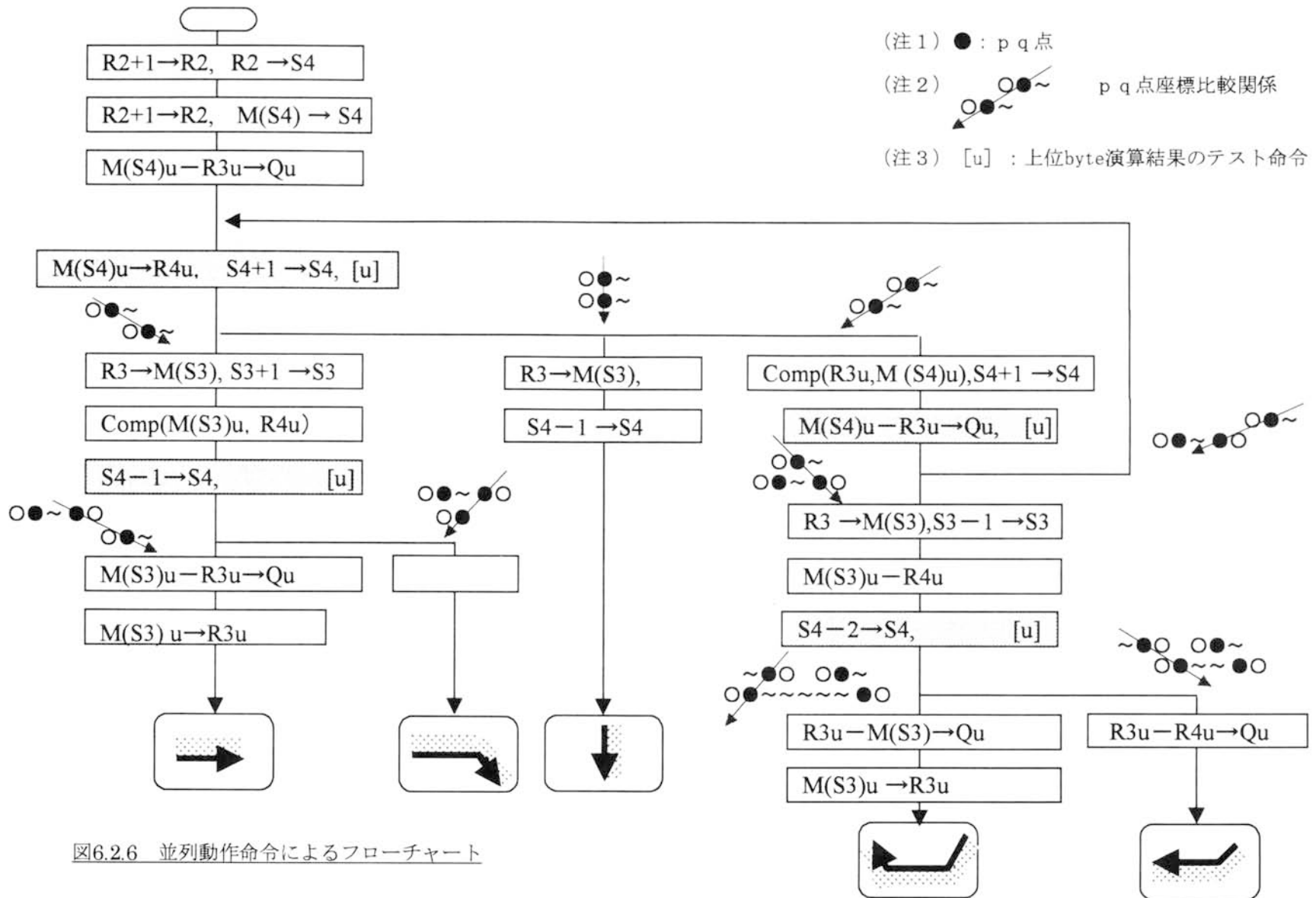


図6.2.6 並列動作命令によるフローチャート

に示す処理では、分岐先で先行して実施した演算を戻す必要がある。これでは、一見意味の無い機能のように考えられるが、yes での分岐回数が no で継続する場合より遥かに多い場合はこの 1 ステップ削減の効果が生かされる。

図 6.2.4 においては、メモリアクセスおよびメモリアドレスの更新が 1 ステップとなり、また、矢印で示すように数ステップの命令が組み合わせられて 1 ステップに集約される。

上記(ア)～(エ)までのハードウェアによる命令強化を実施した時の、プロセッサのアーキテクチャを図 6.2.3(b)に示す。アドレスレジスタ S を 16 個とし、汎用レジスタおよび演算器 (ALU) は、byte に分割された構造である。ただし、汎用レジスタおよび演算器 (ALU) とも word 演算も行なうため、結合も可能としている。

6.2.5 効果

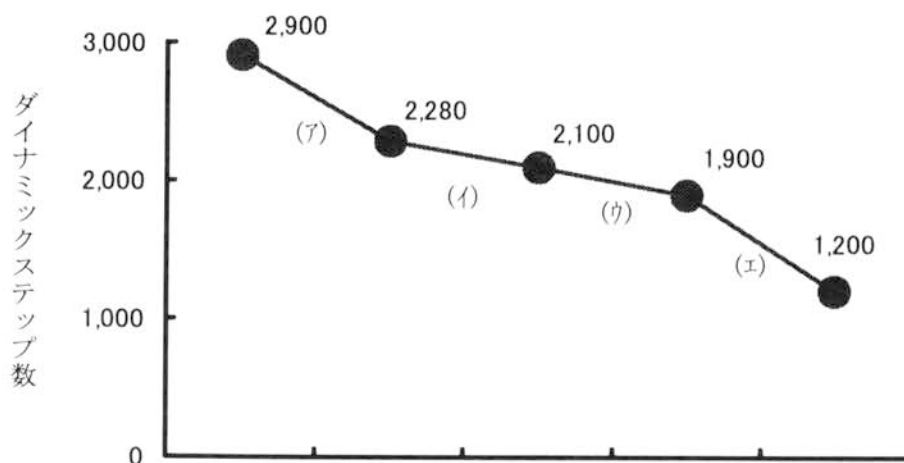
図 6.2.6 に、ハードウェアで強化した後のフローチャートを示す。

式(6.2.1)と同様にモデルパターンでの最終のダイナミックステップ数を計算する。

$$\text{ダイナミックステップ数} = (8+10) \times 120/2 + 3 \times 32 + 6 \times 4 = 1,200 \quad (6.2.2)$$

図 6.2.7 に各対策毎の効果として、モデルパターンにてダイナミックステップ数を試算した結果を示す。

この結果から、ハードウェア増強によるダイナミックステップ数の減少の効果は非常に大きいことがわかる。なお、輪郭線追跡処理では、効果の少ない対策も有るが、これらの改良策は、図形検証法全体での繰返し処理の分析改良により効果が認められたため、採用されたものである。



- (ア) メモリアドレス S の複数化
- (イ) 汎用レジスタ Rn の byte 演算サポート
- (ウ) マルチ J u m p 命令のサポート
- (エ) 並行処理

図 6.2.7 プロセッサ機能強化の効果

【まとめ】

ソフトウェア主体で認識処理を実行する場合においても、繰り返し回数の多い処理につき処理速度を試算する必要がある。

- ▶ ソフトウェアにおいては、処理するパタンの内容によりダイナミックステップ数が大幅に変動するため、試算用のモデルパターンを作成することが必要である。
- ▶ 処理速度が問題となる場合は、計算アルゴリズムの改良を行ないソフトウェア処理に適したデータ形式、計算方法などの変更により繰り返し処理回数の削減、繰り返し処理中の終了判定方法等の削減によりダイナミックステップ数を減らす。
- ▶ さらに、この計算アルゴリズムを基に、ハードウェアの強化による速度向上が可能であり、ここにおいてもハードウェアとソフトウェアの最適化が効果を発揮した。

第 7 章

前 处 理 部

第7章 前処理部

本章では、画像処理のハードウェアで特徴的な構成をとる、2次元マスク論理により2値画像の処理を行なう前処理部の例につき述べる。

7.1 処理部の処理内容

前処理部の処理内容は、表 7.1.1 および図 7.1.1 のようなものであり、いずれも2次元画像データを、マスクを利用して加工することが処理の特徴である。

表 7.1.1 前処理部の処理内容

処理内容	内容	入力データ	出力データ	処理方法	処理ステータス
切出し	フィールド等複数文字のイメージから、1文字を切出す	フィールド画像	1文字画像	マスク処理	開始／終了検出
ノイズ除去	不要なノイズを除く	切出した1文字画像			処理実施
欠け埋め	文字の欠けを修復する				処理実施
スムージング	文字線を滑らかにする				処理実施
細め	文字画像を細線化する				細め処理実施
端点分岐点検出	端点分岐点を検出する	端点分岐点パターン	端点・分岐点検出		
投影	縦横の投影パターン作成	投影パターン	マスク処理／OR処理		
位置正規化	位置正規化された文字画像を作成	位置正規化パターン	シフト処理		

① 切出し

光電変換部により入力された画像から、1文字の画像を切出す。

現在の光学文字読取装置では、紙面上の位置座標に対応したメモリ上のアドレスから1文字の画像を切出す。

② 端点検出、スムージング、欠け埋め、ノイズ除去

マスクにて検出された点は、消去あるいは埋められる。マスクの例を図 7.1.2 に示す。

ただし、実際のマスクは、5×5程度のマスクでマスク条件も各処理に応じて複雑な内容となっている。

③ 細め³⁾

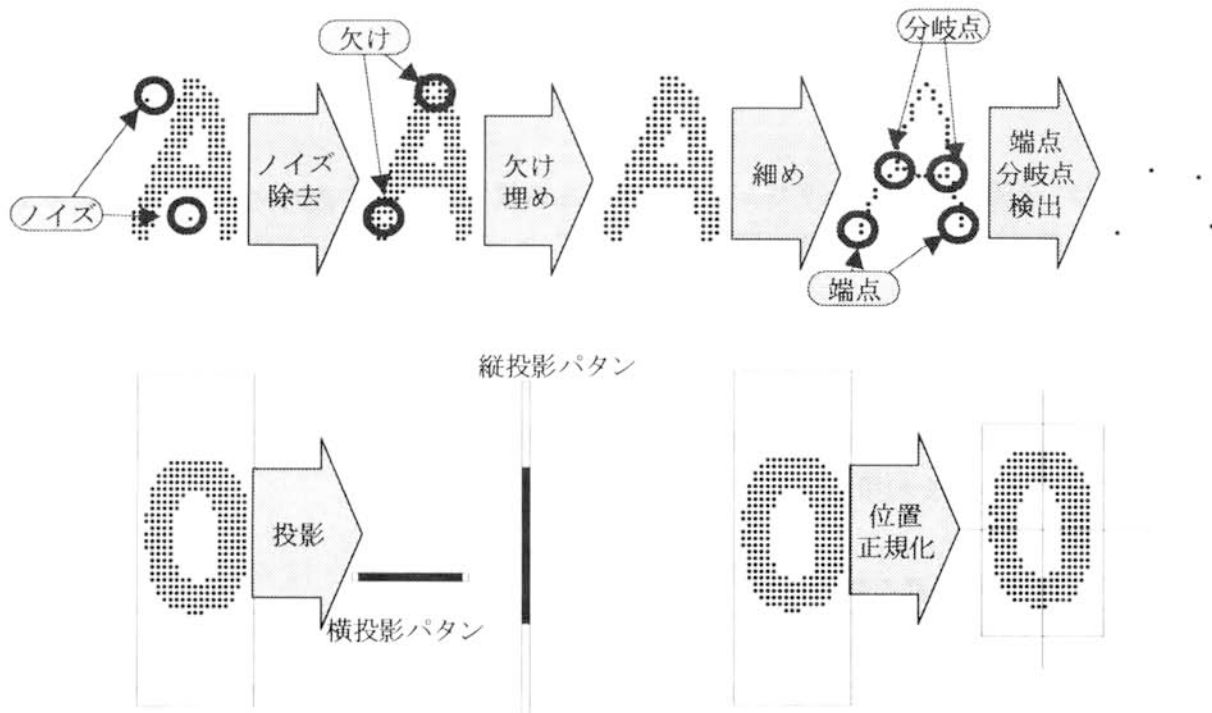


図 7.1.1 前処理部の処理内容

ノイズ除去

0	0	0
0	1	0
0	0	0

“1”を“0”に置換

欠け埋め

1	1	1
1	0	1
1	1	1

“0”を“1”に置換

細め

×	×	×
×	1	0
×	×	×

×	×	×
×	1	×
×	0	×

×	×	×
0	1	×
×	×	×

×	0	×
×	1	×
×	×	×

いずれかのマスクに該当したときは、“1”を“0”に置換

(注)

- ・×は、“1”または“0”
- ・本マスクは、概念であり、太さ2ビット以下の線は欠けてしまう。実際のマスクは、4×4、或いは5×5の複雑なマスクで実施される。

端点検出

1	0	0
0	1	0
0	0	0

0	1	0
0	1	0
0	0	0

0	0	1
0	1	0
0	0	0

0	0	0
0	1	1
0	0	0

0	0	0
0	1	0
0	0	1

0	0	0
0	1	0
0	1	0

0	0	0
0	1	0
1	0	0

0	0	0
1	1	0
0	0	0

いずれかのマスクに該当しないときは、“1”を“0”に置換

図 7.1.2 マスクの例

細めマスクにより、細線化パターンを作成。1ビットの細線にするため、処理ステータス信号が off となるまで、複数回実行される。

④ 投影

縦および横の文字画像の投影パターンを作成。投影方法は、単純なORでなく、マスク処理され、より正しい中心位置を計算できるように構成する。

⑤ 位置正規化

文字画像を、指定されたワークエリアの中心に来るよう、文字画像をシフト処理により移動させる。

7.2 ハードウェアとソフトウェアとの対比

2次元マスク処理につき、ハードウェアとソフトウェアとの対比を行なう。

表 7.2.1 ハードウェア処理とソフトウェア処理

		ハードウェア処理	ソフトウェア処理
処理時間	1画素計算時間	1クロック	20~100サイクル
	1パターン処理時間	画像サイズで固定的に決定	画像の内容により変動
パイプライン処理		可能（併行処理可能）	不可
制限事項	1スキヤンの画素数	固定（ex.64bit）	変更可
	マスクサイズ	固定	変更可

➤ 1画素計算時間

処理時間は、1画素の処理時間の見積により比較を行なう。

➤ ハードウェア

ハードウェアにおいては、1画素の計算時間は1クロック（ex.120ns）で可能となる。

1パタンの処理時間は、パタンのサイズすなわち総ビット数により一意に決定する。構成によっては並列、並行処理により速度向上が可能である。しかしながら、パタンのサイズ、特に1スキヤンの画素数は固定され変更できないのが一般的である。ただし、スキヤン数はバリエブルに設計されるため、固定されない。また、同様にマスクのサイズも3×3あるいは5×5のように固定される。このため、3×3のマスクを5×5へ大きくすることは、変更できないと考えるべきであるため、当初の決定は、慎重に行なう必要がある。

➤ ソフトウェア

ソフトウェア処理においては、サイクルタイムおよびダイナミックステップ数により、計算時間が求まる。

サイクルタイムは、通常複数のクロックから構成されるため、サイクルタイム

はクロックの数倍の時間となる。従って、ソフトウェア処理はハードウェア処理に比較しビット処理では大幅に遅くなる。

計算アルゴリズムの改良および命令の増強により、このダイナミックステップ数を減少させることは可能であるが、ハードウェアより速くすることは困難といえる。

また、速度を向上させる場合には、通常パタンの内容、すなわち黒ビットの割合などにより変化するなど、一般的に処理する画像により処理時間が異なる。

しかしながら、プログラム処理であるため、ハードウェアのように1スキヤンの画素数やマスクサイズなどの変更には柔軟に対応できる利点がある。

7.3 マスク論理の基本回路

図 7.3.1 に、2次元画像のマスク処理の基本回路²⁰⁾を示す。

(1) 構成

▶ 入出力レジスタ：

メモリとの入出力を行なうレジスタで、メモリの 1word (ex.32bit) で構成さ

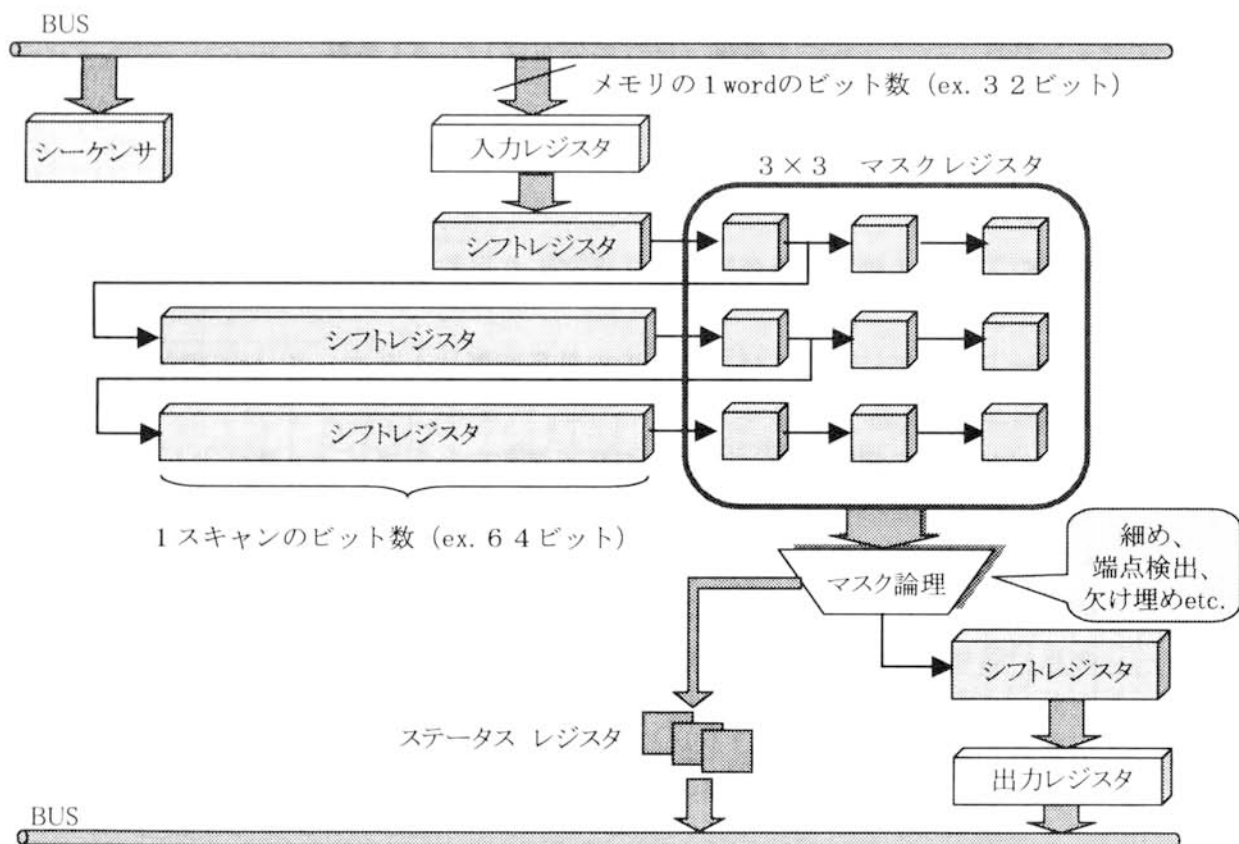


図 7.3.1 マスク論理の基本回路

れる。

▶ マスクレジスタ :

マスク論理に必要な $n \times m$ ケで構成される。細め処理のため、 5×5 等が用いられる。

▶ シフトレジスタ :

マスクレジスタに接続されるレジスタで 1 スキャンのビット数(ex.64bit)で構成される。

▶ マスク論理 :

$n \times m$ ケのマスクレジスタの出力を入力し、1 ビットの画像データを出力する。マスク論理は、7.1 節で述べた処理内容毎に切り替えて使用される。初期にはゲートで構成されていたが、現在では、ROM/RAM を使用するため、複雑なマスク条件も容易に実現できるようになった。また、RAM を使用するケースも多く、この場合、装置の電源投入時などにマスクデータがダウンロードされる手間が必要となるが、必要に応じてマスク内容を改良できるため、装置ハードウェア完成後の評価改良には非常大きな効果を発揮する。ROM/RAM 使用の場合は、マスクレジスタの出力をROM/RAM アドレスとし、ROM 出力を画像データおよびステータス信号に使用する。

▶ ステータスレジスタ :

処理終了後に、プロセッサが処理の状態を知るために用意される情報を格納するレジスタ。

ex.細めの終了を知らせるため、細め処理中、細めのため画像を変更した場合にセットされる。細め処理終了後に、このレジスタを調べることにより、細線化完了、細め処理を繰り返すかどうか判定できる。

▶ シーケンサ :

前処理部をプロセッサと独立させてパイプライン動作をさせるために、シーケンサを設け、動作モードの制御、シフトクロックの発生、メモリの入出力制御（アドレスの生成、同期制御など）およびプロセッサへの終了割り込み制御などを行なう。

シーケンサは数十から数百ステップのマイクロプログラムで構成され、プログラムは、ROMあるいはRAMに格納される。RAM使用の場合は、装置の電源投入時などにプログラムがダウンロードされる手間が必要となるが、装置ハードウェア完成後の改良には柔軟に対応できる利点がある。

(2) 特徴

マスク論理の基本回路の特徴をまとめる。

- ◆ 2次元パタンのマスク処理を、高速に行なうハードウェアの基本形
- ◆ 1画素のマスク計算が、1シフトクロック(ex.120ns)で可能
- ◆ シーケンサにより、プロセッサや他の論理と独立して併行に処理が可能

- ◆ 1 スキャンのビット数が固定されている必要がある。

(3) 動作

マスク論理の基本回路の概略動作を述べる。

- ◆ マスク処理は、1ビットずつ行なう。
- ◆ 入力レジスタには、BUS を介して、1word のデータがセットされる。各レジスタはシフトレジスタで構成され、1bit 毎にシフトされる。n × m マスクの出力は、マスク論理により、1bit の出力を生成され出力レジスタにシフトされる。また、同時にステータスレジスタにマスク状況がセットされる。

(4) 細め処理のプログラム

具体的に、前処理論理を使った細め処理の動作につき図 7.3.2 にて述べる。

- ▶ プロセッサは、前処理部の初期設定を行ない、前処理部を起動し、前処理部の終了割込み待ちとなる。
初期設定内容：処理内容（細め）、入力メモリアドレス、出力メモリアドレス、スキャン数など
- ▶ 起動された前処理部のシーケンサ部は、与えられたスキャン数および固定されている 1 スキャンビット数分シフトクロックを発生させ、終了後プロセッサに、終了割込みにより、終了を知らせる。

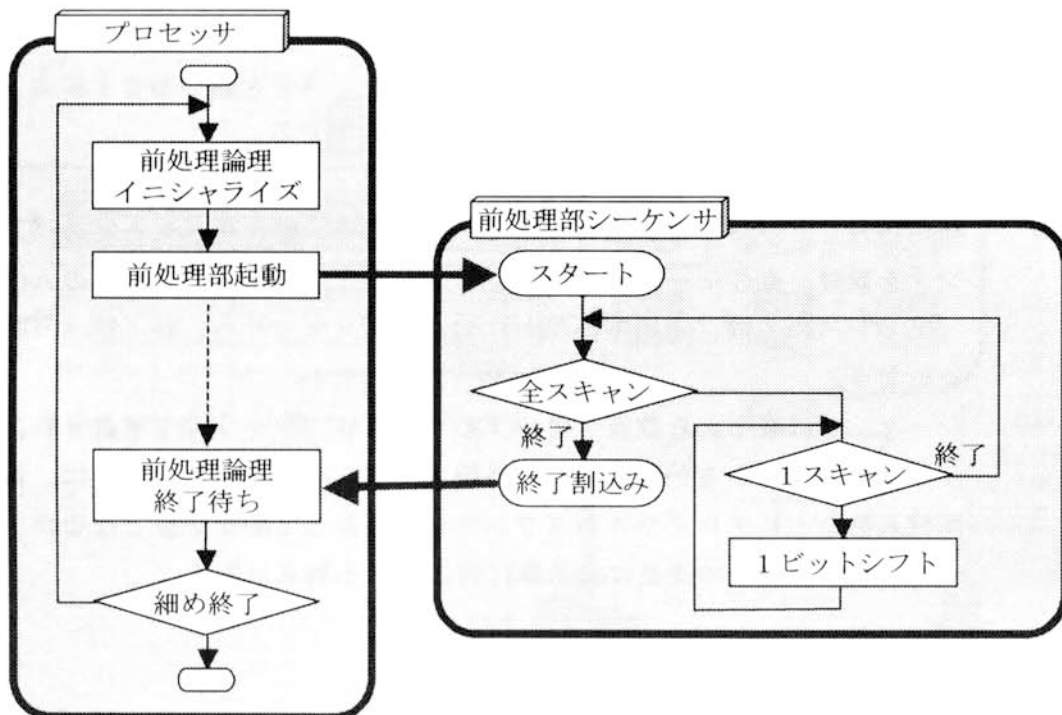


図 7.3.2 細め処理フローチャート

- ▶ 割り込みにより、前処理部の終了報告を受けたプロセッサは、前処理部の細めステータスレジスタを調べ、**on** であればまだ細めが完了していない可能性があるため、再度細め処理を繰り返す。ステータスレジスタが **off** であれば、今回の細め処理では細め処理のため画像変更を行なわなかったため、細め処理による細線化が終了したと判定する。このため、細め処理を終了し次の処理に移る。

以上のように、プロセッサは、前処理部の起動時のパラメタ設定および終了時の判定を行なうのみであり、実際に文字画像の処理は一切行なわない。このように、制御、判定処理は、プロセッサ（=ソフトウェア）で行い、一律で大量データの画像データの処理は、ハードウェアで行なうことにより、ハードウェアとソフトウェアとの最適化が達成される。

第 8 章

今後の動向

第 8 章 今後の動向

文字認識方式の研究においては、認識処理の早い段階で文字画像の情報の削減を行なう方法が一般的にとられてきた。例えば、細め処理は 2 次元画像からマスク処理により文字線を 1 ビットの太さの線にする。これにより、文字線の太さの情報が無くなり文字線の方が求めやすくなるため、後の段階での認識処理の処理量が大幅に削減される。これは、文字認識方式の研究において限られた研究設備の能力(研究に用いるコンピュータの能力)でより多くのシミュレーションを行なえるなど、現実的な方法であった。また、光学文字読取装置のアーキテクチャにとっても速度・コスト面で望ましい方向であった。

しかしながら、認識処理の早い段階での処理は、判定結果によらず一律的に行なわれることになる。選択的でなく一律な処理による情報削減は、詳細な文字の特徴を欠落させることにも繋がるため、認識精度の限界を設定するものである。コンピュータの能力向上に伴い、この限界を越えて精度の良い認識方式を得るため、一律の処理を避けるようになってきた。例えば、細め処理を行わず 2 次元画像より輪郭線を直接求めて、これから特徴抽出を行なう図形検証法などの方式である。これは手書文字認識における最初の段階の前処理の減少に現れている。これに伴い、認識処理には複雑な処理が必要となってきた。これを受けて光学文字読取装置においても、メモリ、マイクロコンピュータなどの電子部品の高性能化、低価格化に支えられて、ハードウェアからソフトウェア処理に移行してきた。

この動向は今後も継続して進むと予想される。以下では、この動向を踏まえて、現在ソフトウェア化できずハードウェアで実現されている光電変換部および 2 値化部につき、今後の動向を述べ、また文字認識方式の動向についても触れる。

8.1 光電変換部、2 値化部の処理内容

光学文字読取装置において、光電変換部および 2 値化部の役割は非常に重要である。特に、光学文字読取装置の最も重要な性能である文字認識精度を左右する。これは、2 値化画像を前提としている現状の文字認識処理では、光電変換部および 2 値化部の役割は、不

表 8.1.1 光電変換部、2 値化部の処理内容

項 目		実施部位
画像情報の選別	ドロップアウトカラー除去	光電変換部
補助情報の入手	帳票周辺データの入力	
画像信号の正規化	シェーディング補正	2 値化部
	白レベル正規化	
文字線の太さ正規化	濃淡正規化	2 値化部、前処理、特徴抽出
画像情報の整形	スムージング	
	欠け埋め	
	ノイズ除去	

要な情報を削除し有効な特徴を残した2値化画像情報を後の段階の処理に渡すことである。

このような、認識処理に有効な2値化画像を得るために、現在の光学文字読取装置の光電変換部および2値化部では、表8.1.1の処理を行なっている。

◆ 画像情報の選別

文字認識に有害な情報を排除するため、入力する画像を選別する。

▶ ドロップアウトカラー除去

“人”が文字を記入するためには、紙面に記入位置を指定する文字枠が必要である。しかしながら、この文字枠は、文字認識には有害であるため、画像からこれを除く手法が光学文字読取装置では用いられている。すなわち、認識する文字は黒色と制限し、文字枠は赤または青のドロップアウトカラーで印刷する。光電変換部では、光源またはフィルタで赤色または青色を除いて電気信号に変えることにより、文字枠を除去し、文字画像のみ入力する。

なお、文字枠として黒印刷の黒線文字枠を使用する例も増えている。これは印刷コストの低減などのため用いられているが、認識精度は文字枠の影響を受け低くなることは避けられず、使用は限定されている。

◆ 補助情報の入手

光電変換部では、文字画像のみでなく、文字の切出し位置を確定するための情報の入手も行なわれる。

▶ 帳票周辺データの入力

認識すべき文字の位置は、紙面上の縦横の座標指定により決められる。このため、紙面周辺の情報が必要となり、この情報を画像として入力することも行なわれている。これには、紙面の周辺は白色に制限して、光電変換部により走査される紙面の背面を黒色にする。これにより、紙面外は黒色、紙面周辺は白色として入力されることになり、認識部に帳票周辺の情報を渡すことができる。

◆ 画像信号の正規化

光電変換部は、光源、センサおよび増幅アナログ回路など、温度、寿命などで性能が変動する部品により構成されている。時間的・空間的に安定した画像信号を取り出すために、画像信号の正規化処理がなされる。これには、光の無い状態を反射率0%とし、帳票は白色に制限してこの反射率を100%とし、画像信号を正規化する処理がなされる。

▶ シェーディング補正

主走査方向（全面走査方式では、水平方向）の光源の照度は、場所により変動する（シェーディング）ことは避けられない。この変動は50%を超える。これを補正するのがシェーディング補正処理である。これは、シェーディングの形状が機器の光学系により固定することから、このデータを利用して補正する。

▶ 白レベル正規化

紙の白さは一定でなく、また光電変換部を通過する場合には、若干の浮き上がり等により反射率が異なる。これを補正するため、帳票は白色に制限し、走査中常時帳票の白部分の反射率を基準として測定して、補正を行う。シェーディング補正が水平方向であるのに対し、この補正は垂直方向であり、また常時観測してフィードバックするため、光源・センサの時間的変動も補正されることになる。

◆ 文字線の太さ正規化

文字認識に用いる2値化された文字画像で重要なことは、文字線の情報の欠落を防ぐことである。これは、文字線が欠落することにより、認識処理では重大なエラーである誤読が避けられなくなるからである。このため、2値化部では、記入されている文字の濃淡にも影響されずに文字線が安定して検出されるよう多くの努力が行なわれている。

▶ 濃淡正規化

文字の濃淡は非常に激しいが、文字線の濃度の変動しても文字線が安定して検出されるよう対策²³⁾される。すなわち、文字線の濃度が変化しても一定太さの文字線を入力することを目的としている。これは、白い紙面に黒い文字線が書かれるとの制限を加え、文字線自身の濃度を検出して2値化レベルを変化させる方法である。ただし、現実の濃度差は非常に大きく十分満足な結果は得られていないため、手書文字では筆記具の制限、インパクトプリンタでの印刷では、リボンの印字行数の制限などを加えている。

◆ 画像情報の整形

認識しやすくする、あるいは誤読を防ぐため、画像を整形することも行なう。これは、2値化处理や前処理および特徴抽出処理などで実施する。ただし、2値化や前処理での一律処理で画像を操作することは、必要な情報の欠落を避けられず認識精度に限度を設定することになるため、2値化部ではなく認識処理で行なう選択適用が増えている。

選択適用の例として、ノイズ除去を述べる。無条件に2mm×2mm以下の黒点を除く場合は、手書数字のみの認識では良好な結果を得る。しかし、カタカナの認識では濁点・半濁点が消えてしまう弊害が生じる。このため、無条件でのノイズ除去を止め、認識文字種がカタカナの場合にはノイズとして除去するサイズを小さく0.5×0.5mm程度とする。

▶ スムージング

文字の線縁をスムーズにする。

▶ 欠け埋め

文字線の切れ、かすれ、欠けを補正する。

▶ ノイズ除去

文字線以外の孤立したスポット等を除く。

8.2 光電変換部と2値化部の構成

2値化部は、当初はアナログ回路により、構成²⁴⁾されていた。ICの発達によりデジタル論理が安価に使用できるようになってからは、アナログ回路の不安定さを除くこと、および2値化画像に文字認識に有用な情報を残すための複雑なマスク処理²³⁾などを行なうため、安定性のあるデジタル論理に移行した。図8.2.1に2値化部の構成例を示す。

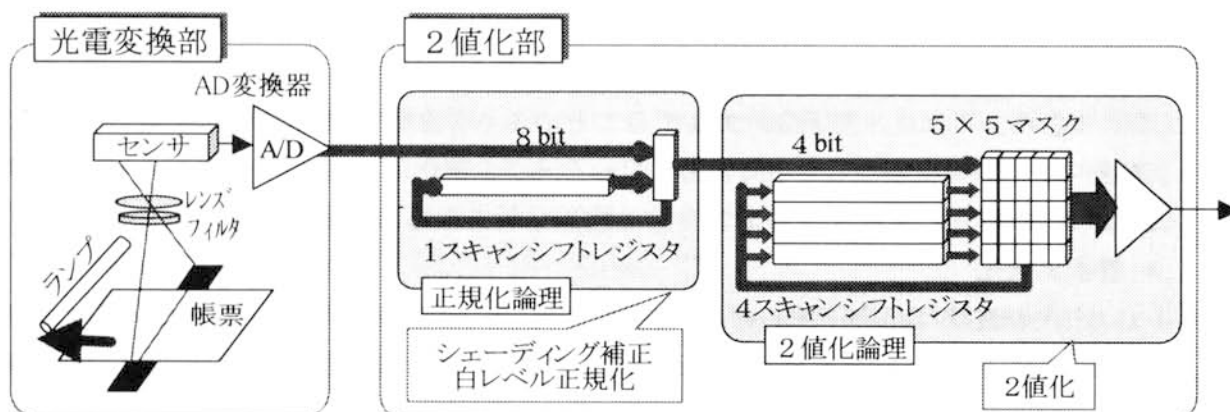


図 8.2.1 光電変換部、2値化部の構成

- 光電変換部では、帳票からの反射光をCCDセンサなどによりアナログ電気信号に変換する。このアナログ信号は、AD変換器により、8～10ビットのデジタル信号に変換される。
- 2値化部は、図7.3.1のマスク論理の基本回路を基礎にしている。ただし、各画素が多値であるため、各シフトレジスタおよびマスク論理も多値となり、ハードウェアの物量は多い。

➤ 正規化論理

シェーディング補正および白レベル正規化を行なう。このために1スキャンの画素数分の白レベルを保持する多値のシフトレジスタおよび補正論理回路により構成されている。補正論理回路の動作は、図8.2.2および式(8.1)に示すように、入力した画素信号の演算を行なう。

$$V_i = \frac{V_w}{V_{wi}} v_i \quad (8.1)$$

ここで、 v_i は*i*番目の画素のAD変換器出力、 V_i は*i*番目の画素の補正された出力、 V_{wi} はシフトレジスタに保持されている*i*番目の画素の白レベル値、 V_w は白レベル基準値（固定値）である。

➤ 2値化論理

正規化された画素の信号は、2値化論理により2値に変換される。ここでは、5×5のマスクにより2値化対象画素の周辺の濃淡情報を考慮して2値化される。

なお、ハードウェアの物量を減らすため、各画素は4ビット（16階調）程度に粗くされることが多い。

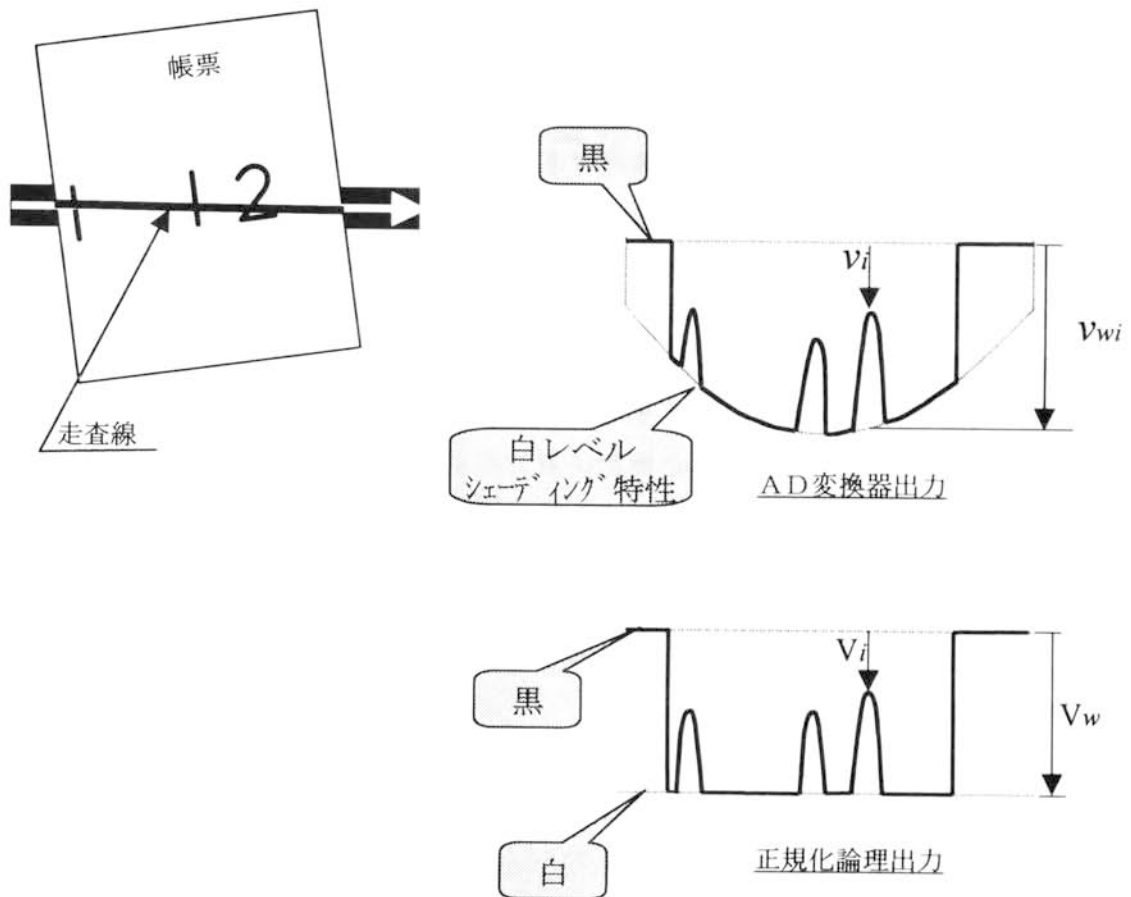


図 8.2.2 画像信号の正規化

8.3 今後の動向

8.3.1 2値化部ソフトウェア化の課題

8.2 節に示すように、2値化部では多値のマスク論理を複数用いるため、計算量は2値画像のマスク処理に比べ、1桁以上2桁に近く増加する。また1画素の処理は、クロックで動作し、100nsec程度の高速で処理される。

このため、計算量が多く第6章に述べたように計算アルゴリズムによる計算処理の削減を図っても、現在、ソフトウェア処理では必要な速度を得られない。したがって、現状ではハードウェアで実現されているが、今後マイクロコンピュータの性能向上によりソフトウェア化されると予測される。

8.3.2 認識方式の動向

上記のようにプロセッサの能力向上に伴い、2値化部がソフトウェアにされると予測されるが、同時に研究設備の能力も向上することになる。このことから、2値化処理による詳細情報の欠落から生じる認識精度の限界を超える方式が研究されると予想される。すなわち、前述のように画像整形はすでに特徴抽出処理に選択適用されているが、現在最も課題となっているのは、文字の濃度変化が激しいときに生じる文字線情報の欠落である。これは、現在の濃淡正規化処理は一律の処理であるため、うすい文字線の欠けと濃い文字線間のつぶれの両対策のバランスがとれないためである。

このため、正規化された多値の文字パターンからリトライなどの必要時に選択適用される2値化処理が考えられる。すなわち、判定処理で欠けによる読取不能と判定された場合、2値化方法を変更して2値化処理を行ない、また、文字線が太いと判定されたときには、スライスレベルを上げて文字線を細くするよう2値化を行ない再判定を行なうなどが予測される。なお、現在でも前者は2値化画像に対してボカシまたは太め処理を、後者は細め処理を実施して、ある程度効果を上げているが2値化画像の限界があり十分ではない。

さらに、2値化画像を前提とせず多値画像を直接認識に利用するアナログマッチングなども研究実用化されると予測できる。

8.3.3 制限緩和

8.1節で述べたように、現在の光電変換部および2値化部は、基本的には帳票は白紙であり、認識する文字はドロップアウトカラーの文字枠に黒色で記入するとの制限を前提としている。

しかしながら、現在では人目をひくカラフルな用紙が広く用いられている。このような帳票は光学文字読取装置の対象外となるため、装置の需要を狭めることになる。

このため、光学文字読取装置は、このような制限を除く方向に発展してゆくと予測される。この場合、光電変換部はカラー化が必要となる。また、2値化部は多色多値という一層複雑な処理が要求されるため、時期はマイクロコンピュータの性能向上に依存するが、ハードウェアでなくソフトウェア処理により実現されると予想する。

[まとめ]

今後の方向は以下のように考えることができる。

- 2値化画像を対象とする文字認識方式を採用している光学文字読取装置でも、マイクロコンピュータの能力向上および価格の低下に従い、2値化部はソフトウェア処理に移行する。
- さらに、研究設備の能力が向上するに従い、2値化による詳細特徴欠落による認識精度の限界を超えるため、多値画像を基にする認識方式に発展すると考えられる。
- また、帳票の制限緩和のため、光電変換部はカラー化されると予測される。

第 9 章

結 言

第9章 結 言

本論文は、光学文字読取装置のアーキテクチャをどのように構成するかの手法の研究に関するものである。

光学文字読取装置は自動認識機械として、常に人と認識能力を比較され、同時に他の入力装置とコストパフォーマンスを比較されるという性格を持つ。このような環境に置かれた光学文字読取装置は、進展する認識方式の研究成果を、日進月歩の先端技術および先端部品を用いて実用化することにより発達してきた。認識能力に対しては最高の認識精度を求めて能力の高い認識方式を採用しなければならないが、実用化にあたってはコストパフォーマンスの面で課題が生じる。これをその都度種々の現実的なアプローチで解決してきた。しかしこの種々の解決手段も、実用化の手法として常に一定の方針に基づいたものであり、認識処理・画像処理に共通なアプローチとして本論文の目的とするところである。

第1章では、光学文字読取装置の歴史的経緯を技術面、性能面を含めて概説し、アーキテクチャ研究の背景と位置付けを行なった。

第2章では、光学文字読取装置のアーキテクチャの要点と課題を述べた。

光学文字読取装置の役割として、第1に入力されるデータの信頼性すなわち文字の認識精度であり、また、第2に他の入力装置と比較される良好なコストパフォーマンスであることを示した。データ信頼性のためには、最も認識能力の高い文字認識方式を採用すると共に、精度向上のため装置ハードの完成後でも求められる改善可能性(柔軟性)が必要であり、パフォーマンスとしては客観的に評価される処理速度の速さである。このため、アーキテクチャとしては改善可能性(柔軟性)、高速度および低コストが求められる。また、ハードウェアとソフトウェアの特質を文字認識処理という観点から対比し、これから、光学文字読取装置のアーキテクチャには、改善可能性(柔軟性)のため、ソフトウェアの能力を生かし、また、コストパフォーマンス向上のため、ソフトウェアとハードウェアの最適化を図らなければならないことを示した。

第3章では、光学文字読取装置の開発手順から開発の成否を左右する重要なアーキテクチャ決定の手法を述べた。動作方式の決定に基づき、ハードウェア主体かソフトウェア主体かが決定する。これを受けて、それらに応じた計算アルゴリズムの研究開発および処理速度・コストの試算からハードウェアとソフトウェアとの最適化を実現する手法を述べた。特に、精度を第1優先として採用された認識方式では、実用化の課題が多いのが一般的である。高速化のためには、繰返し処理の回数削減と繰返し処理短縮化を実現するための計算アルゴリズムの開発が、重要かつ効果的であることを明らかにした。

第4章からは、これらの手法を適用した例を述べた。

第4章では、1,200字/秒の性能を必要とするドキュメントリーダにおける可変閾値法の相関器について述べた。これは速度制限がありかつ高速処理であることからハードウェア主体とし、ハードウェアでの処理の実現のため、可変閾値法の理論式に含まれている

平方演算、除算を除き、小数を整数化する計算アルゴリズムを開発した。これに基づきハードウェア化に当っては、9個の相関計算回路による並列処理、9個の相関値の最大値検出、類似度を求める乗算、および最大類似度・次大類似度の検出をシフト論理回路にて実現、かつ全体をパイプライン処理による並行動作とし実用化した。

第5章では、手書文字認識処理の照合論理の例につき述べた。手書文字認識においては、複雑な処理が多いため、ハードウェア化による速度向上には困難な点がある。しかし、標準パターンとの照合処理にダイナミックプログラミングが用いられた非線型マッチング法においては、繰返し演算処理が多くこの照合処理は実用化の課題となっていた。これを高速化したハードウェアの実現例を報告した。計算アルゴリズムでは、対角線追跡論理により計算ステップ数の削減および単純化を実現、また計算単位を複数化した並列処理、対角線追跡論理の回路化によりハードウェア化し、全体をパイプライン処理として実用化した。

第6章では、ソフトウェア主体の例として手書文字認識に図形検証法を採用した例を述べた。図形検証法では、2次元文字画像から輪郭線を抽出し、1次元チェインコードに変換する処理の高速化が課題であった。ここでは、ソフトウェア主体としハードウェアとの最適化により実用化した。計算アルゴリズムでは、2次元文字画像の数値表現化を実現、ソフトウェア処理に適した形式にした。これに繰返し処理ステップの削減のためハードウェア増強によるプログラム命令の強化および並列動作化により処理速度を達成した。

第7章では、2値化画像のマスク論理の基本回路について述べた。

第8章では、今後の動向として、認識方式の研究および光学文字読取装置の経緯から2値化部のソフトウェア化、認識方式の多値画像化、および光電変換部のカラー化の予測を述べた。光学文字読取装置の発達経緯から、電子部品の性能向上により2値化部のソフトウェア化が予測されること、研究設備の能力向上に伴い現在の2値化段階での詳細情報欠落に基づく認識精度の限界を越えるため、多値画像を基にする方式に発展する予測も述べた。さらに、帳票の制限緩和のため光電変換部ではカラー化に進む方向も明らかにした。

以上述べたように、本研究は光学文字読取装置の実用化に当たりそのアーキテクチャをいかに構成するかの手法の研究成果を述べたものであり、文字認識と同様な傾向の処理が必要な画像処理装置、自動認識機械にも貢献するものと信ずる。

● 発表論文と本研究の関係および発展・適用状況

筆者は、認識方式の研究から、実用化のための装置の研究開発および顧客への稼働支援にいたる光学文字読取装置の全般に携わってきた。本研究では光学文字読取装置の実用化開発に当り、アーキテクチャ構成の要点と課題およびその決定手法の研究成果を述べた。また、発表論文は、装置開発に必須な認識方式の研究に関するものが中心であり、市場のニーズ動向などの諸事情により実用化時期とずれを生じている。

以下では、発表論文と本研究との関係、および発展・適用状況を述べる。

- (1) 安田, 門田, 藤本, 花野井, “加重相関による単一字体印刷文字の認識”, 信学論(D), 56-D, 10, pp545-552, (1973.10)

本発表論文は、初期に実用化されたストロークアナリシス法に変わる、抜本的に新しく、高精度かつ汎用性のある印刷文字読取方式の研究成果を報告したものである。この研究成果は1972年にH-8959光学文字読取装置¹⁰⁾に適用され、現在に至っても基本的な方式として印刷文字認識、特に高精度を必要とされる印刷数字の認識に適用されている。

筆者は、この認識方式の実用化のため実現性・高速性・コスト/パフォーマンス検証を行った。本研究の第2章および第4章で述べた印刷文字認識方式は可変閾値法と統一して称したが、これは本発表論文で報告した類似度法の判定処理を発展させた方式である。

実用化に当り2.4節の図2.4.1に示す認識フローの全処理を確立したが、この中の類似度計算および2段判定、さらに位置正規化およびスライス選択の課題の基礎的研究は本発表論文に報告されたものである。これらは第4章に述べたように高速相関器および9位置相関法、マルチスライス認識方式に発展させ、高速タイプの光学文字読取装置に適用された。

- (2) 花野井, “OCR専用プロセッサ”, 情報処理, 解説, Vol18-4, pp376-380, (1977-4)

本解説は、第2、3章で述べたアーキテクチャ決定手法と、その手法により実現した光学文字読取装置アーキテクチャの概要を解説したものである。実例としてハードウェアとソフトウェアとの最適化を図った認識部のアーキテクチャと共に、第2章および第4章で述べた可変閾値法による印刷文字認識の処理動作を、低速タイプのH-8959¹⁰⁾と高速タイプのH-8957とを対比して解説した。

また、本解説の“将来の展望”で述べた内容は、その後すべて実現し、見通しの正しさが立証された。

なお、本解説にあるH-8957は、手書文字認識に非線形マッチング法を採用し、第5章に述べた照合論理を含めた認識各処理をパイプライン化したハードウェアにより、毎秒500字の高速認識を実現した光学文字読取装置である。この製品により非線形マッ

ング法の有効性・実用性を実証し、the 3rd International Joint Conference on Pattern Recognition で発表された引用文献（11）の研究に貢献した。

- （3）安田，山田，斉藤，山本，花野井，“文字認識のための相関法の一改良-相補的特徴抽出の提案”，信学論(D)，J67-D，2，pp224-231，(1984.2)

手書文字には2.4節で述べたように構造解析法が一般的に適用される。ただし、構造解析法では、標準パタンの作成を機械的に行なうのは困難で、設計者の介入により個別に調整が行なわれる。字種が数千となる漢字認識にはこの作業が膨大となり実用化に大きな障害となる。これを回避するため、統計的手法で機械的に標準パターンが作成可能な相関法を手書文字に適用すべく行った基礎的研究を報告したものが本発表論文である。

1980年代後半の手書漢字読取の実用化および1990年代後半の構造解析法と組合わせた手書文字の複合認識方式に、本発表論文は影響を及ぼした。

- （4）中野，藤澤，国崎，岡田，花野井，“文字認識と協調した表形式文書の理解”，信学論(D)，J69-D，3，pp400-409，(1986.3)

本発表論文は、光学文字読取装置の用途拡大のために煩雑な光学文字読取装置の読取書式情報（フォーマットパラメタ）設定の簡易化、半自動化を目的とした研究の成果を報告したものである。

筆者は、この研究を行なうための「文書OCR」システムの試作を行い、それによる実用的観点からの評価を行った。

試作に当たっては、第3章に述べた手法に従って、アーキテクチャを決定し、また、詳細設計、製作を主導、ハードウェア完成後の改良評価を行った。

本発表論文では、一般的な伝票類は表形式であることに着目して、枠線の抽出から表形式の構造理解、さらには文字の認識と知識処理により、帳票の書式を理解し、光学文字読取装置のフォーマットパラメタを生成するという総合的な研究であった。

この研究成果のうち、枠線の抽出によるフォーマットパラメタの半自動作成は1990年代に実用化され、現在では光学文字読取装置に必須な機能とされている。

また、「文書OCR」システムは、スキャナとパソコンから成り、パソコンにて全体動作を制御する新規な方式とした。これは第1章1.4節で述べたように、従来の専用プロセッサ制御に変わり採用された構成である。当時のパソコンでは、速度的に難があったため、低速のHT-4171光学文字読取装置（1986年）に初めて採用され、パソコンの高性能化、高機能化に伴い、現在では全ての光学文字読取装置に採用されているOCRシステムの原型となった方式である。

- （5）武田，田畑，花野井，“不完全な点対称画像のパターンマッチング-印鑑照合への応用”，信学論(D)，J69-D，4，pp542-548，(1986.04)

印鑑照合作業の省力化は、金融業界における合理化ニーズの1つであった。本発表論文は、このニーズに応える印鑑照合装置の開発を目的とした研究の成果を発表したものである。ここでの課題は、印鑑照合方式の確立、特に中心点検出論理の開発と、試作機の製作による利便性の確認検証であった。

筆者は照合アルゴリズムの全処理について実用化を前提とした開発・見直しを行った。また、第3章に述べた手法により試作ハードウェアの設計製作および実証評価を主導した。

方式の確立に当たっては、文字認識方式をベースに印鑑独特の特徴を利用した。すなわち、2.4節の“表 4.2.1 代表的な認識方式”に示す認識対象文字の特質から、印鑑は手書文字ではなく印刷文字に相当するため、照合処理には相関法の理念を貫いた方式とした。

課題であった中心点検出には、点対称であることに着目し、単純な処理で正確な中心を検出する論理²⁸⁾を新たに開発した。これは、第6章で開発した輪郭点座標表現形式の外点 p q から、点対称図形の中心を求めるもので、従来の手法では不十分であった欠け・ノイズ重畳の印に対しても正確な中心点を求めることに成功した。

試作機開発に当たっては、照合処理時間が実用上 1 秒以下であれば良いことから、2.4節の最適化手法に従い、第6章で開発したプロセッサを用いたプログラム処理とした。(本発表論文にある基本クロック 4 MHz のビットスライス型マイクロプロセッサ)

また、照合処理には、本研究に述べた多くの計算アルゴリズムを使用している。回転角検出および回転処理の計算には 5.3 節で述べた対角線追跡論理をプログラムにて利用、また乗算には図 3.4.9(c)のテーブル参照方法により高速化を図った。また、1画素以内の誤差補償のため重ね合せには、第4章で述べた9位置相関法を採用した。

開発した試作機は、印鑑照合全処理を 0.95 秒で実現し、研究内容を実証し本発表論文で報告した。

なお、確立した印鑑照合方式は、1990年代後半に実用化された。

謝辞

本論文をまとめるに当って懇切にご指導下さった信州大学工学部、中野康明教授に御礼申し上げます。また、本論文の作成に当り、信州大学工学部、師玉康成教授、岡本正行教授、松本弘教授の御指導を戴いた。

また、明星大学安田道夫教授には、筆者の研究の誠に長きにわたり励ましとご支援を賜ったことに深く感謝申し上げます。

本論文は筆者が日立製作所在職中の1968年から1990年の間に行なった光学文字読取装置の研究開発のうち、アーキテクチャの研究をまとめたものである。日立製作所の小田原工場にて研究開発され製品化するに当り、文字認識方式の研究開発は中央研究所において行なわれており、これらの事業所の多くの方々のご援助を頂いたことに感謝する。

この間、中央研究所において、研究をご指導いただいた山本真司博士、川崎淳博士、藤本好司博士、遠藤裕英博士、藤澤浩道博士、国崎修博士に御礼申し上げます。また、林晋一研究員、牧原洋技師、岡田邦弘技師には、研究の具体的な協力をいただいた。

また、システム研究所において田畑邦晃元主任研究員、武田晴夫博士には画像処理の研究でご指導いただいた。

また、岐阜大学山本和彦博士には、理論面でご指導いただいた。

小田原工場入出力機設計部における上司としてご指導いただいた森昌夫元設計部長、故佐野太一郎元主任技師、下矢吉孝元設計部長に御礼申し上げます。

また、小田原工場入出力機設計部 OCR グループにおいて協同で研究開発に当った、白貝泰男元主任技師、高橋次男主任技師、門田彰三主任技師、大原義仁部長、道野正雄副部長、伊藤敏部長、増崎秀文部長に御礼申し上げます。

引用文献

- (1) 小沢, “インプット研究”, 日本経営出版会, 東京, 1966年
- (2) 中田(編), “パターン認識とその応用”, コロナ, 東京, 1978年
- (3) 山本, 安田, 宮本, 堤, “手書き数字認識論理の設計”,
信学論(C), 53-C, 10, pp691-698, (1970-10)
- (4) 安田, 門田, 藤本, 花野井, “加重相関による単一字体印刷文字の認識”,
信学論(D), 56-D, 10, pp545-552, (1973.10)
- (5) 安田, 山田, 斉藤, 山本, 花野井, “文字認識のための相関法の一改良-相補的特徴抽出の提案”,
信学論(D), J67-D, 2, pp224-231, (1984.2)
- (6) 中野, 藤澤, 国崎, 岡田, 花野井, “文字認識と協調した表形式文書の理解”,
信学論(D), J69-D, 3, pp400-409, (1986.3)
- (7) 武田, 田畑, 花野井, “不完全な点対称画像のパターンマッチング-印鑑照合への応用”,
信学論(D), J69-D, 4, pp542-548, (1986.04)
- (8) Y. Nakano, H. Fujisawa, O. Kunisaki, K. Okada and T. Hananoi, “A Document Understanding System Incorporating Character Recognition”, Proceedings of the 8th International Conference on Pattern Recognition, pp.801-803 (1986)
- (9) 花野井, “OCR専用プロセッサ”, 情報処理, 解説, Vol18-4, pp376-380, (1977-4)
- (10) T.Sano and T.Hananoi, “Model H-8959 Optical Character Reader”, Hitachi Review, Vol22, 8, pp341-347
- (11) Y. Fujimoto, S. Kadota, S. Hayashi, M. Yamamoto, S. Yajima and M. Yasuda, “Recognition of Handprinted Characters by Nonlinear Elastic Matching”, Proceedings of the 3rd International Joint Conference on Pattern Recognition, pp. 113-118 (1976)
- (12) 遠藤, 藤澤, 小松, 国崎, 花野井, “OAにおける認識技術”, 日立評論 Vol68, 2, pp71-76(1986-2)
- (13) 花野井, 藤本, 安田, “OCRの現状と将来”, 昭和 52 年電気四学会連合大会, 221, pp6-89, (1977)
- (14) M.Yasuda, H.Yamada, T.Saito, K.Yamamoto and T.Hananoi, “An Improved Correlation

- Method for Character Recognition-A Proposal of Reciprocal Feature Extraction”,
Systems Computers Controls, Vol.15, No.4, pp.29-38(1984)
- (15) H.Takeda, K.Tabata and T.Hananoi:“Matching of Distorted Central Symmetry Image -
An Application to Seal Impression Collation”, Systems and Computers in Japan, Vol.18,
No.4, pp.24-32(1987)
- (16) 情報処理学会編, “情報処理ハンドブック, 4編7・3文字・図形入力”,
オーム, pp290-295, (1989-5)
- (17) 広瀬他編, “コンピュータソフトウェア事典, H-V 3光学文字読取り装置”,
丸善, pp736-739, (1990-4)
- (18) 倉水, 牧原, 門田, 安田, 花野井, “可変閾値法による文字認識装置”, 特許公報, 昭
52-19416
- (19) 藤本, 花野井, 安田, 倉水, “検出装置”, 特許公報, 昭 53-14894
- (20) 藤本, 花野井, 安田, 門田, 牧原, 倉水, 太田, 小木戸, “図形前処理装置”, 特許公報,
昭 55-51230
- (21) 林, 花野井, 藤本, 安田, 門田, “パターン認識装置”, 特許公報, 昭 56-15546
- (22) 藤本, 門田, 安田, 林, 花野井, “文字認識方式”, 特許公報, 昭 56-19657
- (23) 角本, 花野井, 遠藤, “図形の前処理装置”, 特許公報, 昭 56-33746
- (24) 小松, 花野井, 太田, “ビデオ増幅回路”, 特許公報, 昭 57-22265
- (25) 安田, 中野, 藤澤, 花野井, “類似度計算方式”, 特許公報, 昭 57-550
- (26) 花野井, 今藤, “光学文字読取方法とその装置”, 特許公報, 昭 56-44455
- (27) 林, 藤本, 花野井, “方向差演算回路”, 特許公報, 昭 57-15432
- (28) 武田, 田畑, 花野井, “画像処理方法”, 特許公報, 平 6-64601
- (29) 栗野, 太田, 花野井, 内山, “光学文字読取装置”, 特許公報, 平 7-66441
- (30) 宮武考文, 松島整, 江尻正員, “ラン型方向コードに基づく2値画像の高速輪郭追跡の一手
法”, 信学論(D-II), J79-D-II, 3, pp341-350, (1996.3)