

対話的マルチメディアハンドリングを
指向した画像処理・認識技術の研究

上田博唯



①

対話的マルチメディアハンドリングを
指向した画像処理・認識技術の研究

上田博唯

目次

第1章 序論.....	5
1.1 本研究の背景.....	5
1.2 対話的マルチメディアハンドリングにおける認識技術の利用.....	9
1.3 本論文の概要.....	12
第2章 対話型図形編集システム.....	17
2.1 緒言.....	17
2.2 図形の記述法.....	17
2.2.1 図形の基本定義.....	17
2.2.2 基本図形.....	19
2.2.3 複合図形.....	21
2.3 図の編集機能.....	23
2.3.1 データ構造.....	23
2.3.2 図形コマンド列の逆方向探索.....	25
2.4 図形エディタの試作.....	26
2.4.1 装置の概要.....	26
2.4.2 図形の彩色とハッチング機能.....	28
2.4.3 特殊図形.....	29
2.4.4 文字品質劣化を改善した補間法.....	30
2.5 結果の検討.....	33
2.6 結言.....	35
参考文献.....	36
第3章 対話型映像編集システム.....	39
3.1 緒言.....	39
3.2 基本構想.....	40
3.3 ビデオ解析機能.....	42
3.3.1 カット分割機能.....	42
3.3.2 運動解析機能.....	44
3.3.3 対象物抽出機能.....	46
3.3.4 対象物存在判定機能.....	48
3.3.5 オブジェクトリンク.....	48
3.4 ビデオ情報の記述.....	50

3.5	ビデオ構造の視覚化	53
3.5.1	時間軸編集の基本画面構成	53
3.5.2	オブジェクトリンクによるナビゲーション	54
3.6	実験結果	56
3.6.1	個別機能について	56
3.6.2	編集システムとして	60
3.7	結言	62
	参考文献	63
第4章	画像処理用高速プロセッサ	65
4.1	緒言	65
4.1.1	動機と目的	65
4.1.2	汎用イメージプロセッサの位置付け	66
4.2	GPIPの構成	68
4.2.1	GPIPのプロセッサユニットDSP-i	68
4.2.2	基本構成	70
4.2.3	GPIPの動作モード	73
4.2.4	シフトリングの制御	75
4.3	領域分割処理	77
4.3.1	領域分割モードに関する考察	77
4.3.2	領域分割モードにおけるシフトリングの制御	81
4.4	GPIPの性能の評価	85
4.4.1	領域のオーバーラップによる性能低下	85
4.4.2	非正常部の置換による性能低下	88
4.5	結果の検討	91
4.6	結言	93
	参考文献	93
第5章	結論	95
	謝辞	101
	論文目録	103

図表一覧

図1-1	コンピュータの利用形態の変遷	5
図1-2	人間とシステムの対話	7
図1-3	映像情報の論理的な構造	8
表1-1	代表的な画像のデータ量	11
図2-1	図形の定義	18
図2-2	基本幾何図形とその代表位置,代表寸法	19
図2-3	メモリに格納される記号の例	20
図2-4	各種図形のための総合計算体系	22
図2-5	複合図形の諸例	22
図2-6	図の記述法	23
図2-7	図形の挿入と削除	24
図2-8	図の多層構造と逆方向探索	25
図2-9	逆方向探索のフローチャート	25
図2-10	図形エディタの構成	27
図2-11	白黒CRTに表示された入力パラメータと動的仮想キーの例	28
図2-12	線図形の定義	30
図2-13	従来の補間法の結果	31
図2-14	手法Aの補間法の結果	32
図2-15	手法Bの補間法の結果	33
図2-16	適用例(漢字)	33
図2-17	適用例(ひらがな)	33
図2-18	試作した装置の外観	34
図2-19	カラーCRT上に表示された作図例	34
図3-1	ビデオ情報の階層構造	41
図3-2	ブロック毎のヒストグラム類似度 $E(t, k)$ の計算法	43
図3-3	カット自動分割の画面例	44
図3-4	動き検出の基本原理(ブロックマッチング法)	45
図3-5	動き検出の実験結果	45
図3-6	ビデオ画像からの対象物の切り出し例	46
図3-7	切り出し画像による合成映像の例	47

図3-8	カットの内容を操作するためのメタファ	48
図3-9	被写体検索の原理図	49
図3-10	映像構造の記述	51
図3-11	オブジェクトリンク情報の記述	52
図3-12	時間軸編集の画面例	54
図3-13	オブジェクトリンク・ナビゲーションの画面例	55
図3-14	カット分割アルゴリズムの従来手法との比較	57
表3-1	バン検出アルゴリズムの評価結果	58
表3-2	対象物判定機能の性能	59
表3-3	対象物の分離に使用した関数の種類と使用回数	60
図4-1	DSP-iのブロック図	69
図4-2	GPIPのブロックダイアグラム	71
図4-3	試作したGPIPの外観	71
図4-4	PUボードの外観	72
図4-5	バス制御部のブロックダイアグラム	76
図4-6	バス制御部の接続状態	76
図4-7	画像の領域分割(2次元分割の場合)	77
図4-8	フィルタリングによってできる非正常部	78
図4-9	オーバーラップさせた部分画像Sij(一点鎖線で示す)	79
図4-10	Sijから見たオーバーラップ・エリアと正常データの転送の関係	79
図4-11	部分画像間にまたがる線の例	81
図4-12	画像の領域分割(1次元分割の場合)	81
図4-13	2次元領域分割モードにおけるPUの接続関係	82
図4-14	画像サイズに対するGPIPの計算能力	87
図4-15	画像サイズに対する、GPIPの非正常部の置換に伴う性能低下	90
表4-1	PUの接続方法の比較	73
表4-2	GPIPの動作モード	75
表4-3	領域分割モードにおいて10%のオーバーヘッドとなる画像サイズ	87
表4-4	GPIPの処理時間の例	91

第1章 序論

1.1 本研究の背景

近年の半導体技術の進歩に支えられて、種々の技術が今までは考えられなかったような分野へと、その応用を拡大しつつある。マルチメディアの処理もその例外ではなく、かつては研究機関でぜいたくに計算機を使える人々だけにしか縁のなかったような、かなり高性能の自然言語処理や視覚情報処理等が、OA(オフィスオートメーション)やFA(ファクトリーオートメーション)という言葉で象徴されるように、オフィスや工場に設置される機器の中に、妥当な価格で組み入れられる時代になってきている。そして特にワークステーションやパソコンは、近年の性能向上に伴ってマルチメディア対応の幅が広がり、テキストや図面のみならず時間軸を持つ情報である映像(動画像+音声)までも扱うことを目指すようになった。

専門家	ビジネス・ユーザ	パーソナル・ユーザ
科学者の道具	仕事の道具	日常の道具 (情報社会へのアクセスの道具)
計算の高速化	効率のサポート	文化的なサポート
ハードウェア アーキテクチャ	ソフトウェア アーキテクチャ	ヒューマンインタフェース(HI) アーキテクチャ
プログラム言語	メタファ (GUI)	マルチメディア ハイパーメディア
コマンド	ダイレクト マニピュレーション	マルチモーダルIF
プログラミング・テクニック	知的情報処理	感性情報処理

図1-1 コンピュータの利用形態の変遷

これに伴い、利用形態も大きく変わりつつある。図1-1に示すように、コンピュータの開発当初は主として科学者が計算を高速に行なうための道具であった。従ってユーザは計算機の内部構造をそのまま操作するように設計されたコマンドを操り、時には機械語までも理解するような専門家の人々であった。それ故、研究

の中心課題は計算ハードウェアのアーキテクチャやプログラム言語であった。次いでビジネス・ユーザが使うようになり、データ整理や文書作成といった仕事の効率を上げることが目的となった。そこで、操作方法の習得が容易ではないという面をカバーするために、日常的な常識で操作を類推できるようにする必要があった。これに応えるためにメタファを用いたグラフィクス・ユーザ・インタフェース(GUI)が開発され、ダイレクトマニピュレーションの重要性も認識されはじめた。これからの計算機の主な利用形態は、社会の情報化とネットワーク化の進展を見ても分かるように、パーソナルなユーザの公私にわたる活動のための情報社会への双方向アクセスの道具になって行くと考えられる。ここでは、従来の高速計算や効率の向上よりも個人や社会の文化的活動をサポートする必要がある、GUIに加えてマルチメディア・ハイパーメディアやマルチモーダル・インタフェース、そしてそれらを支える知的情報処理が、さらには感性情報処理が重要な研究課題になると考えられる。

ところで、マルチメディアとはメディア(情報)の独立性を保持しながら、関係を規定して混在させるものである。すなわち、マルチメディアは集合体であり、その集合体を構成している各コンポーネントを独立に取り出すこともでき、それらの相互関係(主として時間・空間の関係)を、それを使用する時に自由に操作できることが必要とされる。このことがマルチメディアに大きな有用性をもたらしているのだが、同時に技術的な困難さを増大しているとも言える。

まず、ユーザにとって、マルチメディア情報は時間と空間に広がる膨大な量のデータであるため、その全体像を把握して効率よくハンドリング(編集、検索、修正、合成etc.)することが、慣れていない人にとっても容易なことではない。そのために映像情報の編集には熟練した専門家の知識と技術、あるいは経験と勘に頼るところが多く、しかもその作業は莫大な手間のかかる、わずらわしいものとなっている。これらの問題からユーザを解放し、創造的活動ができるようにする知的情報処理技術の開発が必要である。

また、ハイパーメディアはメディア間にリンクを張ることによって、柔軟なハンドリングをサポートするための仕組みである。しかしながら、関連する情報を見つけ、それらの間にリンクを張る作業は、現状では多大な労力を必要とする手作業に

より行なわれている。またユーザが必要とする情報間のリンクは、個人の好みや、その時その時のユーザのタスクの内容に大きく依存するから、それら全ての可能性を予測してリンクを張っておくということは、ほとんど不可能に近い。ユーザのタスクや意図に合わせ、ダイナミックに対応できるようにするためには、知的メディア操作技術、すなわちインタラクティブ環境での利用を前提とした知的情報処理技術の開発が必要である。

システムの側に関する主な技術課題は、時には感性的なものまでもが重視されるパターン情報であるマルチメディアが、数値やシンボルを論理的に扱う従来のコンピュータ技術のフレームワークに、そのままでは納まらないことにあると考えられる。データ量の膨大さや時間軸の扱いの難しさも無視できない問題である。

これらのことから、信号処理や認識・理解といったインテリジェント技術を応用して、マルチメディアのハンドリングにかかわるユーザの創造的活動をサポートする環境を実現するという観点からの研究が重要となってくる。

この研究では、時間・空間に広がるマルチメディア情報の構造をユーザの立場で解析する能力と、その構造を記述する能力をシステムに持たせることが、主要なテーマとなる。なぜなら図1-2に示すように人間とシステムは、メディアを通じて対話するからである。

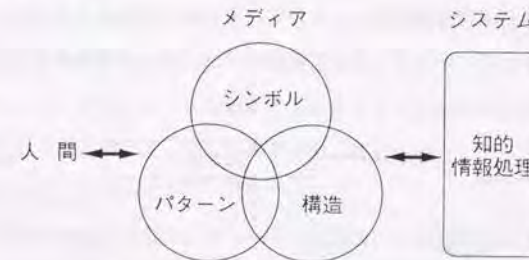


図1-2 人間とシステムの対話

マルチメディアはパターン情報であり、人間はそのパターン情報からその構造を把握して、最後にシンボリックな情報として理解している。この活動と、その逆のパターン情報を生成する活動により人間同士の対話が成立していると考えられる。

したがって人間とシステムが対話する際にも同等のことができるようにする必要があり、その第一歩として、システムがパターンを解析し、その結果として得られる構造記述を使って、ユーザがマルチメディア情報を容易にハンドリングできるようにするための機能を提供することが重要なテーマである。また、それらを適切な表現方法で見せること、すなわちユーザがマルチメディア情報の全体構造から細部までをシームレスに見渡すことが可能で、更にシステムが提供する機能が、それにどう作用するのかを分かり易く示すことのできるような新しいインタフェースメタファの研究も必要である。

具体例として図1-3にビデオ情報の階層構造を示す。最上層はシナリオであり、多くのシーンから構成される全体のストーリーである。各シーンは1個以上のカットから成る。シーンはその構成要素として、サブシーンを持つこともある。この構造においては、上部の層はシンボリックな記述、下部の層はより視覚的な記述から成る。この構造の最も低い部分(すなわちカット)には、生のビデオデータが含まれる。シンボリックな記述は、人や物や場所の名前、そしてそれらの行動や関係などの言語的あるいは意味的な情報から成る。視覚的な記述は、対象物の形状や位置、対象物の運動軌跡、対象物が画面に入った時刻と出た時刻、複数の対象物が出会った時刻というような幾何学的あるいは時間的な情報から成る。これらの記述情報をできる限りシステムが自動抽出し、さらに、それらの間をハイパー・リンクで結び付けることによって、ユーザに自由で柔軟なハンドリング環境を提供することができるようになる。

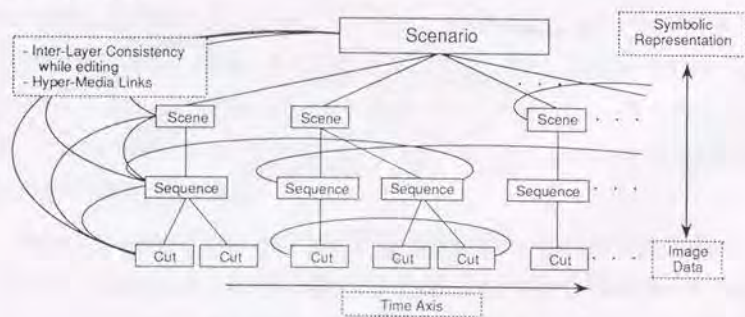


図1-3 映像情報の論理的な構造

1.2 対話的マルチメディアハンドリングにおける認識技術の利用

ここでは、本論文で目標とする対話型マルチメディアハンドリングにおける認識技術の利用に対するコンセプトを提案する。

現在までの音声や画像の認識・理解の研究では、映像編集のような対話型マルチメディアハンドリングの支援を目的とした事例はないようであるが、近いものとして、画像処理用エキスパートシステムの研究があげられる。これらシステムは自動組立てや検査の対象となる特定の画像に対する処理手順の開発支援を目的としている。このシステム的环境は、ユーザが持ち込む問題がシステムにとっては未経験のものであるという意味でオープンシステムである。しかし、このユーザの持つ問題を部分問題に分解したとき、それぞれの部分問題の解として、最適な画像処理アルゴリズムが唯一のものとして(システムで用意されているアルゴリズムの中から)見つけられる種類のものである。つまり、あらかじめ条件が整備された、かなり定形的な作業におけるAI応用であると言える。

これに対し、映像編集システムでは、ユーザがメディアを自分の意志で加工できるようにすることを目的とするので、ユーザのタスクを部分タスクに分解したとしても、その部分タスクに対する望ましい結果(ゴール)は必ずしも唯一ではない。例えばいくつかの画像処理アルゴリズムが適応可能であると考えられる場合に、何が最良の結果であるかの判断ひとつをとってみても、対象画像は多様であるから、汎用的な評価基準を用意することは殆ど不可能に近い。そして、画質や音質については、柔らかな、ザラザラした、さわやかなというように感覚的な処理も要求され、処理結果に対する判断は、ユーザの好みや意図、更には感性に依存し、唯一の基準というものは存在しない。

ここには従来の研究では取り上げられなかった新しい課題があると考えべきである。従来のエキスパートシステムの研究成果等を大いに活用しつつ、まず、ユーザに信頼される知的対話システムを構築するための方法論を確立することが重要であろう。そのためには、音声や画像の処理や認識技術について知らないユーザにでも分かり易いアルゴリズム、人間により近いアルゴリズムを開発する必要がある。そして次にインタラクティブな環境の中で、ダイナミックにユーザの要求に追随し

て行けるようなフレームワークを開発する必要がある。ユーザの好みを学習しながら、それに合った評価基準を自動生成して、ユーザの目的にあった音声処理や画像処理の手順やパラメータを自動的に選択するような、新しいアルゴリズムを実現することも長期的な課題として念頭に置く必要がある。

ところで、映像を対象とするシステムでは、時間的に連続する何枚もの画像(フレーム)の間の相関を利用することによって、1枚の静止画像を処理するよりも精度のよい結果が得られる可能性がある。また、対話的に処理を進めるとき、人間が所要所のフレームについてのみ処理手順を決定しておけば、残りのフレームについてはシステムが適応的に自動処理できる可能性もある。これらのことから、マルチメディアのハンドリングにかかわる対話処理においては、ユーザである人間とシステムの役割分担と、処理に要する時間の兼ね合いをうまくバランスさせることができれば、原状よりもはるかに快適な環境が実現できる可能性がある。

もちろん困難な課題も多いと予想される。例えばマルチメディア情報が持つ意味内容の理解、そしてユーザの意図の解釈の両者をシステムがうまくこなすことができなければ解決できないような高度な課題もあり、性急あるいは安易に認識技術を持ち込むことは慎まねばならない。しかし、作ることによって理解を深めるという方法論を実践することも必要である。特にインタラクティブに使われるシステムでは、研究者やシステム開発者が作ったものとユーザである人間との、実使用環境でのインタラクションを観察することによって、人間の知的活動に関して、より深い理解を得ることができると考えられるからである。

この研究アプローチは、ユーザとコンピュータが知的活動環境を創り出すために、どのように協調することができるのかを実証する試みであるとすることもできる。人間の得意とする能力と計算機が得意とする能力とのバランスの取れた融合を目指すものである。具体的には計算機がビデオ情報を解析し、その結果を使ってユーザがより高度な判断を下し、ユーザは自身の創造力によって情報を生成・加工するような環境を提供することである。すなわち、ユーザが処理しているビデオ情報の構造を、システムはどこまで認識することができるかということ、そして、そのシステム側の認識結果を、どのようにユーザに提示して、ユーザとシステムとの

高度な対話環境を実現するかということが、大きな研究課題である。

一方、ハードウェア面では、ワークステーションやパソコンの性能が向上したといっても、認識技術を応用して快適な対話環境を実現するには、まだまだ十分とは言えない面も存在する。代表的な画像のデータ量を表1-1に示す。画素当りのビット数も異なり画像自身の持つ性質も様々であるが、どれもそのデータ量は膨大なものである。人間が待ち時間を感じないようにするためには、処理時間は0.3秒以下でなければならないとされている。この0.3秒の間に2Mバイト程度のデータを処理するためには、1データ当りに許される処理時間がわずかに $0.15\mu\text{s}$ になってしまう。フィルタリングのような簡単な画像処理プログラムでも1画素当り15命令程度を必要とするので、1命令の実行時間が10ns以下の速度が必要ということになる。これは、100MIPS (Mega Operation Per Second) の性能である。複雑な画像処理では、この10~1000倍の処理量となるから1~100GOPS (Giga Operation Per Second) というような単一のプロセッサでは実現できないほど、大きな計算能力が要求される。そのためマルチプロセッサ型式のイメージプロセッサも必要となる。

表1-1 代表的な画像のデータ量

画像の種類	画面サイズ	画素数 (概数)	ビット数 /画素	総データ量 (バイト)
テレビ	640×480	307K	24	922K
FAX (16本/mm)	3456×4894	17M	1	2.1M
CAD/CAM端末	2000×2000	4M	24	12M
衛星画像 (可視)	14730×14568	215M	6	160M
衛星画像 (赤外)	3690×1821	6.7M	8	6.7M

ところで、画像の処理を大きく3段階に分けると、次のようになる。

(1) 前処理。ノイズ除去や特定の信号成分の強調を目的とするパターン情報レベ

ルの処理である。フィルタリングのような単純な繰返し演算が主体だが、生の画像全体を処理するため、膨大な演算を必要とする。

- (2) 特徴抽出。輪郭線の追跡や特徴的パターン存在の判定等を目的とするパターン情報からシンボル情報への橋渡しの処理である。条件分岐が多発するやや複雑な演算が必要であるが、アルゴリズムを適切に選べば、前処理ほどには演算量は多くない。
- (3) 構造解析。抽出された線分や特徴パターン相互の関係に基づいて、画像の構造を解析するシンボル情報レベルに近い処理である。ソーティング、あるいはモデル(知識)との照合を主体とした演算が必要で、データ量は前2者に比べ減少しているが、組合せ数が増えるので演算量が膨大になることもある。

従来のマルチプロセッサでは、単純だが高速演算が要求される前処理のレベルに着目していた。その場合には、画素数分だけのプロセッサを用意し、それぞれのプロセッサは1ビットの演算という低い機能しか持たない、あるいはデータフロー型の制御構造を用いるというような思い切った構成が採られた。

ところが、イメージ・プロセッサの応用分野が拡大するにつれ、取り扱う画素数(絵の大きさ×分解能)がプロセッサ数で固定される方式では不満があり、また、画素当たりのデータ量も白黒2値だけではなく、R、G、Bの3原色について、それぞれ濃淡8~16ビットまで可能としたいという要求が出てきた。更に上記各レベルで要求される処理も、より高度化してきている。すなわち、前処理のみの高速化では意味がなく、より上位の特徴抽出や構造解析までをカバーするマルチプロセッサシステムが必要とされている。データ演算方式、アドレス演算方式、データアクセス方式等のそれぞれについて、マルチメディア処理に適した多次元・多重的な演算を考慮したプロセッサ・アーキテクチャが必要である。

1.3 本論文の概要

このような背景のもと、本論文はマルチメディア・ハンドリングにおけるソフトウェアとハードウェア両面からの技術課題に対する研究成果をまとめたものであ

る。本研究では、信号処理や認識・理解といった知的情報処理技術を応用して、マルチメディアのハンドリング、中でも特にユーザにとってもシステムにとっても負担の大きい画像(図形から映像まで)のハンドリングにかかわるユーザの創造的活動をサポートする環境を実現するというを目的とする。

本論文は、5章から構成される。

第2章では、まず図形のハンドリングを取り上げて、マルチメディアにおけるマンマシン対話における課題と、その解決の方向を明かにしようとする。ここでは会議用のスライド等の図をカラー画面上に創作するための、新しい対話型図形編集装置方式について提案する。本研究の動機の一つは、画像メモリがまず目に区切られた画素として画像を記憶する構成となることに着目し、画素単位の処理に徹した方式の利点を追及することである。従来のグラフィクスが、直線発生器などの専用ハードウェアで作られる「線」を図形構成の基本要素としているのに対し、ここでは、図形を「面」として定義し、画面上で図形を操作することに重点を置いた新しい図形編集方式を提案する。すなわち図の構成要素である図形を、座標とパラメータの入力に対し、論理値を出力する計算体形として統一的に定義する。

この方式により少数の基本図形から、その論理接続によって任意の複合図形が定義できるようになる。又、図を上記のパラメータ群から成る図形コマンドの順序を持った集合として定義したので、この図形コマンド列を逆方向に探索する処理によって、図形の削除や挿入などが、隠れていた図形の復元を含めて、自然な形で実行できる。それ故、人間にとって理解しやすく、またその操作方法も親しみ易いものとなる。そしてこの方式はシステムにとっても効率よく実行できることを明らかにした。次いで、この図形編集方式を、マイクロコンピュータを主体とした桌上型装置として試作した結果について述べ、多様な図が効果的に構成できることを示す。

但し、マイクロコンピュータによる処理では、おのずと性能に限界があり、通常の文字を主体とした簡単なスライド程度ならよいが、図形の数が増える複雑な説明用の図面の場合には快適性の面で問題があるということも分かった。

第3章では動画像を中心とするマルチメディアの自由な編集・作成という、ユー

ザの創造的活動を支援する環境を実現するための新しいアプローチを提案し、プロトタイプの試作結果について報告する。このアプローチの特徴は画像処理・認識技術を応用して、動画情報情報の時間・空間構造を一貫性のある形で視覚化して、それをダイレクトマニピュレーションできるようにすることにある。これによりユーザーは対話的に映像素材を編集できる。具体的にはビデオ情報を解析して、その構造の自動記述を行なうための機能とこれらの機能によって得られるビデオの時間空間構造をビジュアルライゼーションする新しい手法を提案する。

ビデオ情報の解析機能としては、カット分割、カメラと被写体の動き、被写体の軌跡や輪郭線、特定の被写体の存在の判定及びその存在期間の記述等を提供する。このプロトタイプでは、ワークステーションを利用し、まず自動カット分割機能とカットの内容解析機能を開発した。そしてこれらの機能により分割された動画アイコンを用いた動画の構造の視覚化を実現し、その上でこれら動画アイコンのダイレクトマニピュレーションによる時間軸編集を実現した。また、画像中の対象物の切り出しについても実験的な機能を組み込み、動画の合成など、ユーザーにとってのメリットについて検討した。

更に、被写体の存在判定結果を用いて、同じ被写体はビデオ全体を通じて自動的にリンクされるようにした。このようにして得られるリンクをオブジェクトリンクと名付ける。オブジェクトリンクによるナビゲーションと画像解析結果によってサポートされるビジュアルライゼーション手法により、ユーザーが生のビデオデータのみならず、ビデオ情報の構造、そしてこれに付加される記述内容を、ダイレクトマニピュレーション(直接操作)によって自由に編集できることを示す。

従来のビデオ編集では、ビデオの情報構造が頭の中にはっきりと構築されるまで、ユーザーは早送りと巻き戻しを繰り返す必要があった。そしてこの作業は非常に時間がかかるものであった。しかし、本プロトタイプシステムでは、画像認識技術を応用することによって、この時間にかかるステップが、ほとんど不要となった。そして動画アイコンのダイレクトマニピュレーションによるカットの取捨選択と並べ替えによる実質的な編集に、いきなり取り掛かれるようになった。またユーザーは、これらのビジュアルライゼーションとオブジェクトリンクによるナビゲーション

の機能を使用して、直感的にシーンの構造とビデオのストーリーの流れを概括したり、詳細なカットの記述を自由にブラウジングしたりできる。本章の最後ではこれらのことを実験結果を用いて示し、画像処理・認識技術を応用して、マルチメディアのハンドリングにかかわるユーザーの創造的活動をサポートすることが非常に有効であることを明かにする。

但し、このプロトタイプでもワークステーションの性能不足が顕在化した。カットの自動分割や被写体の存在判定に関しては、アルゴリズムを工夫することにより、リアルタイム処理を可能としたが、動きの解析や対象物の切り出し等の処理は、オフライン処理とせざるを得なかった。NTSCビデオ方式でも数十倍の高速化が、今後普及すると考えられるハイビジョンでは、200~300倍の高速化が必要である。

第4章では、これまでの章で明かとなったパソコンやワークステーションの処理能力の不足をカバーするためのイメージプロセッサの研究について述べる。このイメージプロセッサは前処理から構造解析までの広い画像処理に適用可能とすることを目指した超高性能の並列プロセッサである。ここでは信号処理用プロセッサ(DSP)を画像処理用に拡張して新たに筆者らが開発したLSIをプロセッサユニット(PU)として用いた、マルチプロセッサ構成の超高速汎用イメージプロセッサ(GPIP)の方式を提案し、その試作結果について述べる。本研究で対象としたのは、機能を専用化したものではなく、汎用的に利用できるプロセッサである。例えば、OA応用については、文書に含まれる線画・画像・写真等を分離したり、それぞれをその特性に合わせた方式で加工したり、よりハンドリングし易いデータ構造に変換したり、これらに総合的な処理を加えるといった高度で複合的な処理を実行できるプロセッサである。それ故、機能の汎用性と共に十分な高速性を持たせることが課題となる。

ここで提案する方式は、物理的にはPUを1次元のシフトリング結合(データをシフトクロックに同期してPU間で巡回させ、プロセッサもこのクロックに同期してデータを読み書きするバス構造)とすることによって回路規模を抑え、論理的な2次元の制御を行なうものである。これはプリント基板1枚に8PUを搭載可能とするよ

うなコンパクトなマルチプロセッサ・アーキテクチャである。このアーキテクチャでは、PU間 が物理的に1次元接続であるが、これを論理的な2次元配列とみなして制御することによって、2次元の構造を持つ画像を効果的に処理することができる。また、画像処理で必要とする各種機能に柔軟に対処できるように、4つのシフトリング動作モードを用意した。

以上のような設計により、マルチプロセッサの並列動作のオーバーヘッドを常に20%程度に抑えることができた。また、シフトリング上に並列的にデータを載せる高速多重転送方式によって、PUのデータ交換の速度を大幅に向上できた。同時にGPIPは画像のサイズ、データ形式(2値、多値、カラー)や処理内容などに制限がなく、PU数も自由に換えられるフレキシブルなシステムとなった。64個のDSP-iを用いて試作したGPIPの総合性能は、2GOPS(Giga Operation Per Second)を達成した。

この方式はコンパクトであることから、複数のプロセッサとメモリをワンチップに搭載するLSIに発展する可能性があり、そうすれば従来のメモリ・アクセス・ネットワークによる性能低下の問題も解消されるので、ワークステーションやパソコンのアーキテクチャでは到底達成できないような超高性能を実現できると考えられる。

第5章は結論であり、本論文のまとめを述べる。

第2章 対話型図形編集システム

2.1 緒言

図は人間にとって最も理解し易い情報表現の一形式である。そのためコンピュータグラフィクス[2-1]が早くから注目され、例えば設計の自動化を初め、人間と計算機の接点で幅広く実用されて来た。しかしながら、一般に計算機内にできあがった図を出力することが主目的であったり、計算機内にあるデータベースを管理するための補助手段として、図形表示装置を使うことが多く、従来のコンピュータグラフィクスの技術の中では図を対話的に創作し、自在に編集していくという考え方は顕在化してはいない。

これに対し、1980年頃からコンピュータで図そのものを編集する技術が注目され始めた[2-2~5]。このような技術は、オフィスオートメーションの気運の拡大[2-6]と、マイクロコンピュータを中心とした半導体技術の進展によって、その後ますます重要かつ身近な物になってきた。このような点から、本研究では、日常よく使われる図、例えば会議用のスライド等を作成する装置を例題として、対話的な図形の編集を効率的に行える方式を探ろうとした。

又、この種の分野においては、表示のカラー化は必然の方向であった。カラーCRTは一般にラスタースキャン方式であるため、色彩情報を一時的に記憶する一面面分のリフレッシュメモリが必要となる。このリフレッシュメモリは、まず目に区切られた画素として画像を記憶する構成となることに着目し、画素単位の処理に徹した方式の利点を追及することも、本研究の動機の一つとなっている。

従来のグラフィクスが、直線発生器などの専用ハードウェアで作られる「線」を図形構成の基本要素としていた[2-1]のに対し、ここで提案するのは、図形を「面」として定義し、画面上で図形を操作することに重点を置いた新しい図形編集方式である。

2.2 図形の記述法

2.2.1 図形の基本定義

図形Sとは、図を構成する要素であって、ある計算体系A(x, y, p)が与えられ、

パラメータ群 p のもとで、その出力が真であるような点 (x, y) の集合であると定義し、式(1)のように表記する。

$$S = \{(x, y) | A(x, y, p) = 1\} \quad (1)$$

すなわち図形を、図2-1に示すように、座標 x, y とパラメータ p を変数とした2入力1出力系ととらえる。それ故、これら2入力が所定の条件を満たせば真(1)、満たさなければ偽(0)を出力する計算体系(Algorithm)として表現できるものは、すべてここでいう図形となり得る。

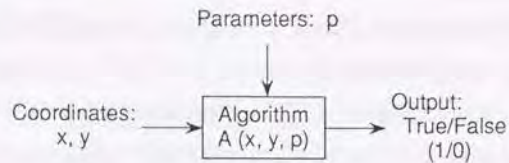


図2-1 図形の定義

図形の簡単な一例は、次のような計算体系

$$A = \begin{cases} 1; & (x-x_0)^2 + (y-y_0)^2 \leq a^2 \\ 0; & \text{elsewhere} \end{cases} \quad (2)$$

を持つ円である。ここでは、中心座標 x_0, y_0 と半径 a がパラメータ p の要素であり、これら形状や位置にかかわるものを幾何パラメータと呼ぶ。ところで、通常はコンパスで描かれる図形、すなわち輪を円と呼ぶことが多いが、本方式では図形を面として扱うので、輪は後述するように大小二つの円 S_A, S_B の論理演算 $S_A \cdot S_B$ によって作られることになる。

さて、この図形の計算体系において、パラメータ p の一つに図形の種類コードを加えると、この計算体系の中で、まずこの種類コードを判定し、円なら円、四角なら四角の定義式を参照するようにはできる。こうすれば各種図形を一括して扱うことができ、従って図2-1あるいは式(1)の定義は、この方式で用いる全ての図形を含む

ものとなる。

このような計算体系は、計算機上では、座標 x, y とパラメータ p (の格納番地)を引数として、論理値を返すサブルーチンとして実現される。従って座標 x, y を次々と与える(走査する)とき、計算結果が真のときだけその画素に色コードを書込むという処理を実行すれば、パラメータ p で定まる図形が画面上に視覚化される(座標駆動)。逆に x, y を固定して、既に描かれた図形のパラメータ p を走査すれば、点 (x, y) がどの図形上にあるかを同定できる(パラメータ駆動)。この機能は後述するように、図形編集のための基本機能となる。

2.2.2 基本図形

本方式では、基本図形として、基本幾何図形、文字、記号の3種を扱う。基本幾何図形は、図2-2に示す9種の図形であり、その計算体系は、先に円を例にして述べたように関数型で実現される。なお、図形と図形の組合せの便宜を考慮して、各図形の代表位置(図2-2中の黒丸)は、面の中心あるいは辺の中点に統一するように選んだ。又、平行四辺形の隣合う辺のなす角は45度としている。文字はJIS C 6226の第1水準の全漢字、平仮名、片仮名、英数字を、それぞれ16×18ドット構成の2値パターンとして記憶する。又、記号は図2-3に示すように、用途に応じた任意パターンを16×16ドット構成で同様に記憶する。

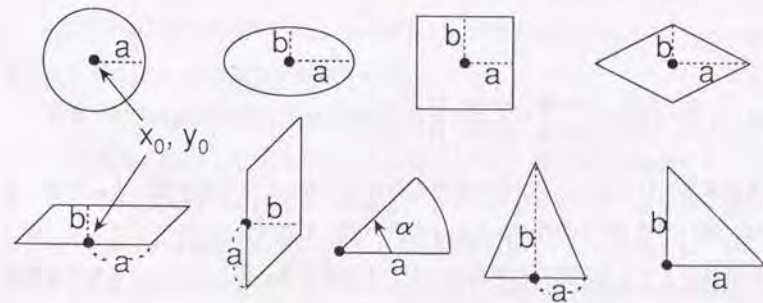


図2-2 本幾何図形とその代表位置、代表寸法



図2-3 メモリに格納される記号の例

ここで、文字と記号は、内部にドットパターンを持った長方形と考える。文字の計算体系では、まず代表寸法が $a=8, b=9$ の長方形とみなし、与えられたパラメータである中心座標 x_0, y_0 、回転角 θ 、倍率 r_x, r_y によって

$$A = \begin{cases} 1; & |X| \leq a, |Y| \leq b \\ 0; & \text{elsewhere} \end{cases} \quad (3)$$

但し

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} \cos \theta / r_x & \sin \theta / r_y \\ -\sin \theta / r_x & \cos \theta / r_y \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \quad (4)$$

なる演算を行い、点 (x, y) がこの長方形の内部にあるかどうかを調べる。内部にあれば、続いて式(4)で求められた X, Y を用いて、文字パターンをアクセスし、ドットの有無によって最終的な計算体系の出力を決定する。但し、このとき X, Y は整数値とは限らないので、補間を行ってドットの有無を決める。

このパターン参照型の計算体系の導入によって、文字や記号を任意の姿勢と大きさで、任意の位置に配置することができる。さらに、全ての図形が、座標とパラ

メータを入力として、論理値を出力する計算体系に統一されるため、もう一つの利点として、図形の論理接続が可能となり、複雑な図形を基本図形の組合せで実現できることになる。

2.2.3 複合図形

任意の基本図形 S に対する否定図形 \bar{S} を、 S の計算体系の出力が偽である点の集合と定義する。例えば、円の否定図形は、その円の外側の全領域である。つぎに図形間の論理演算を定義する。図形の積集合は(5)式で定義される。

$$S_1 \cdot S_2 = \{(x, y) \mid A(x, y, p_1) \cdot A(x, y, p_2) = 1\} \quad (5)$$

そして図形の和集合は(6)式で定義される。

$$S_1 + S_2 = \{(x, y) \mid A(x, y, p_1) + A(x, y, p_2) = 1\} \quad (6)$$

また、このようにして作られた図形を複合図形と呼ぶ。

本方式では、図形と否定図形を任意の数だけ論理接続できるようにした。但し、ここでは操作者の入力順序が図形を組立てる順序であると解釈するようにした。すなわち、 $S_1, \bar{S}_2, S_3, \dots$ という入力で、複合図形 $((S_1 + \bar{S}_2) \cdot S_3) \dots$ が指定される。従って、この複合図形に対しては、パラメータ p_1 に対する図形の計算体系の出力に、 p_2 に対する出力の否定をORし、その結果に p_3 に対する出力をANDし…という演算によって、点 (x, y) の真偽が決定される。

これまで述べた機能を全て包含した統合計算体系をフローチャートで示すと図2-4のようになる。図2-4と図2-1を比べても分かるように、複合図形の場合でも、全体を一つの図形と見なすことができ、座標とパラメータを入力として論理値を出力するという基本概念は保存されている。又、複合図形のパラメータ p の中には、それを構成する基本図形それぞれのパラメータと共に相互の論理接続情報や位置関係の情報が含まれるようにする。従って、ひとまとまりとしての移動や回転、伸縮などが可能となる。この方法で、使用頻度が高い、あるいは作図法を忠実に守る必要のあ

る図形、例えば社章等を一旦作成して保存しておけば、後の利用に非常に便利である。

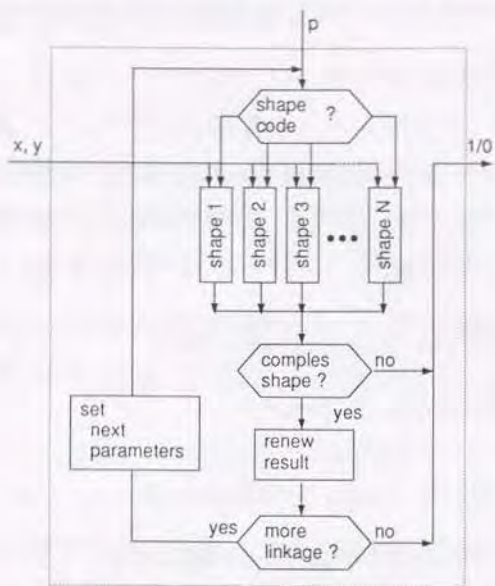


図2-4 各種図形のための総合計算体系

複合図形を作成した例を図2-5に示す。このうちの4例には、後で述べる方法によって、ハッチングが施されている。

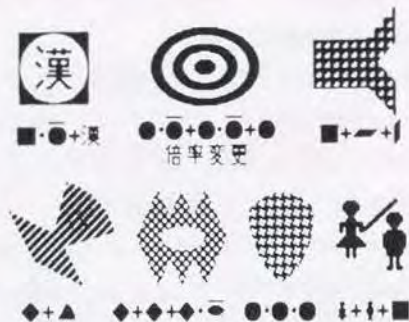


図2-5 複合図形の諸例

2.3 図の編集機能

2.3.1 データ構造

基本図形か複合図形かを問わず、一つの図形は、一つの図形コマンドとして扱い、それぞれのパラメータ p を順に記憶する。図は図形の集合であるから、図形コマンドの列として記述され、図2-6に示すように、一種のリストとしてメモリに記憶される。図2-6では、二つの基本図形から成る複合図形が例示されている。又、一つの図形を表現するのに必要なパラメータとして、前章で述べた幾何パラメータや図形の種類コードの他に、彩色情報、ハッチング情報などの外観を記述する修飾パラメータ、および変域 (range) などが記憶される。変域とは、その図形に関する処理を行なうべき画面上の領域であり、処理の高速化のために用いられる。このデータは、操作者がキーインしなくとも、その図形に外接する長方形 (すなわち最大・最小の xy 座標) として自動的に計算されるものである。

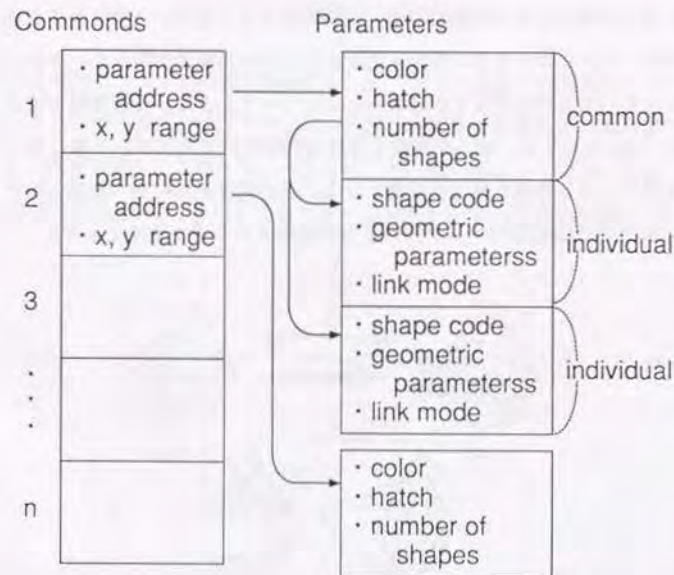


図2-6 図の記述法

さて、既に描かれた図形 $S_1, S_2, \dots, S_i, \dots, S_n$ があり、ここから S_i を削除するときには、図2-6において、単にそれに対応するコマンド i を取去って、コマンド $i+1 \sim n$ を順に繰上げればよい。このように図形コマンド列の編集は、一次元の操作であるため極めて単純であるが、これに対応する、しかも本来の目的である二次元の画面上での削除や挿入の実行は、図形の編集における大きな課題となる。

描画の際には、画像メモリ上のその図形に対応する全ての x, y 位置に、指定された色コードが書込まれるから、図形の重なった部分では、以前の色コードは失われてしまい、画面上には、図2-7(a)に例示するように、最も新しく描かれた図形、ここでは三角形が優先してみえる。ここで以前に描かれた図形、例えば図2-7(a)の長方形を削除する場合、画像メモリ上の色コードのみに着目して処理すれば(各図形の色コードが異なるものとして)、結果は図2-7(c)のようになる。又、この図形のコマンドに従って、描画の逆操作を実行するだけでは、図2-7(d)のようにしかない。

これらの不都合を避けるためには、全体を消去して始めから描き直す方法も考えられるが、それにはかなりの時間を要するし、その間画面が失われるから、編集を行なっている人間の思考を中断することになってしまう。全体を描き直さずに図2-7(a)→(b)のように、挿入や削除を自然な形で実行するには、一般に面と面の重なりを調べ上げる複雑な処理が必要となる。この処理を単純かつ整然と実行できるよう、本研究では図形コマンド列の逆方向探索という考え方をとった。

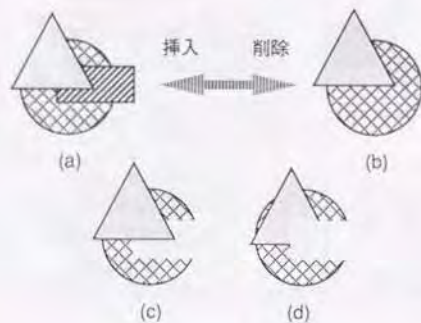


図2-7 図形の挿入と削除

2.3.2 図形コマンド列の逆方向探索

一つの図形は一枚の透明シートに不透明塗料で描かれたものであると考えると、できあがった図は、図2-8に示すように、このシートを順に積み重ねた構造になっている。従って、ある座標 x, y が与えられたとき、どこまで透けて見えるかを調べると、その点に対応する図形を発見することができる。この処理が、図形コマンド列の逆方向探索である。

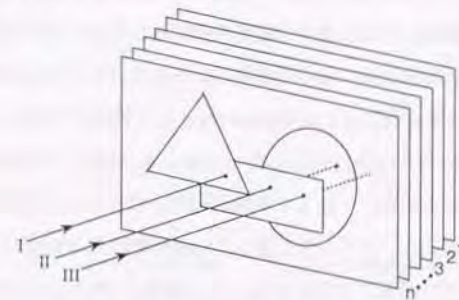


図2-8 図の多層構造と逆方向探索

具体的な図形コマンド列の逆方向探索は、2.1で述べた図形の計算系のパラメータ駆動機能を使って、図形コマンドの n から 1 へ順次各図形の計算体系の論理出力を調べることで実現できる。フローチャートで示すと、図2-9のようになる。なお、ここでは各図形の変域 (range) を用いてスクリーニングを行い、処理の高速化を計っている。

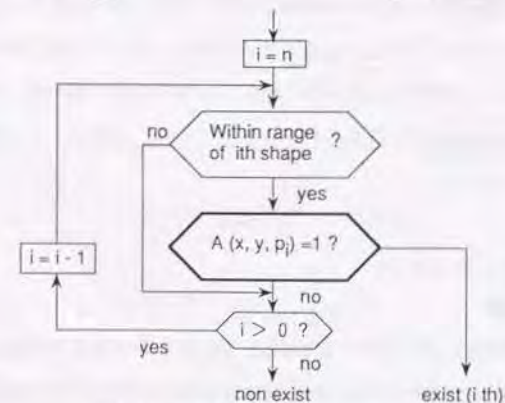


図2-9 逆方向探索のフローチャート

今述べた逆方向探索は、画面上で図形を指定するときに利用できる。すなわち、カーソルで図形上の一点が指定されると、直ちにその図形に対応する図形コマンドが発見される。

こうして指定された図形 S_i を削除するとき、画像メモリ上での処理は、次のように実行される。まずその図形 S_i の変域内で座標 x, y を走査し、ある x, y で S_i に対する計算体系Aの出力が1であれば、その点(画素)は消去対象の候補となる。このときには、その座標で図形 S_0 から S_{i-1} までを逆方向探索し、もしA=1が成立する図形があれば、この画素は新しい図形によって書き換えられていることを意味するから、何もせずに次の座標に移る(図2-8の矢印Iを参照)。この探索で一度もA=1が成立しなければ、この画素は消去対象である。従って更に S_{i-1} から S_1 への逆方向探索を続行し、初めてA=1となる図形、すなわち削除図形の後に現われる図形の色コードに書き換える(図2-8の矢印II)。もし S_1 まで達しても図形が発見されないときは、そこは背景であるから、この画素には背景色コードを書込む(図2-8の矢印III)。以上の処理を変域全体にわたって実行すれば、図形 S_i は過不足なく削除される。

一方、図形 S_i の挿入は次のように実行される。 S_i の変域内で座標を走査し、A=1となったときには、その画素は挿入対象の候補である。そこで S_0 から S_{i-1} への逆方向探索で、その前方に描かれた図形の有無を調べる。もし図形があれば何もせず、なければ、その画素に挿入図形の色コードを書き込む。

その他の編集機能、例えば図形の移動、回転、伸縮、色の変更などは、全て削除と挿入の組合せで実現している。ここで述べた図形コマンドの逆方向探索を用いた編集方式の特長は、書換えの必要のない画素には何の変化も生ぜず、画面には常に操作者の意図したままの図が表示されている点にあり、人間にとって極めて自然な処理となることである。

2.4 図形エディタの試作

2.4.1 装置の概要

提案した図形編集方式を実証するために、図2-10に示す構成の図形エディタを試作した。16ビットマイクロコンピュータ(8086+8087)を中心に、図形表示用のカ

ラーCRT、マンマシン対話用の白黒CRT、ジョイスティック、キーボード、図形コマンド列を格納保存するためのフロッピーディスク等から成る卓上型装置である。

マイクロコンピュータのアドレス空間に直接配置されたカラー画像メモリは、512×512画素で4ビットの深さを持つ。その出力はカラーマッピングメモリのアドレスとして加えられる。このメモリは、6ビットのレジスタ16個に相当しており、R、G、Bの3原色に割当てられた各2ビットがD/A変換されてカラーCRTの入力となるよう構成されている。従って、64種の色の表示が可能であり、カラーマッピングメモリの内容を変えることによって、そのうちの任意の16色を指定できるが、ここでは、カラー画像メモリの上位1ビットを、一時メモリとして多目的に利用するため、画像メモリの下3ビットに対応する8色だけを表示色としている。

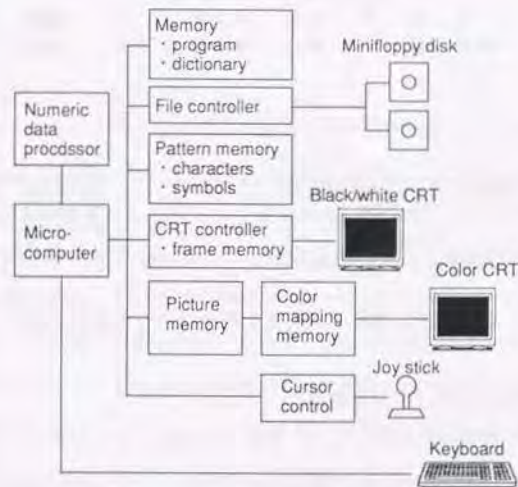


図2-10 図形エディタの構成

コマンドの入力には、動的仮想キー方式[2-7]を採用した。この方式では、図形のパラメータを始めとする選択項目が、図2-11に示すように、白黒CRT上のテンキーを模した枠組(画面右下)の中に、次々と割付けられて日本語で表示される。従って

操作者は、複雑なコマンド形式を覚える必要もなく、キーボード上のテンキーの一つを順次選択するだけで、円滑、適格なデータ入力ができる。同様に漢字を指定するときも、読みを入力するとメモリの中の辞書が参照され対応する漢字が枠組中に表示されるから、希望のものを確認・選択すればよい。又、座標関係のデータは、数値としてキーから、あるいはカーソル位置としてジョイスティックからでも入力ができる。入力データは図形コマンドとして、順次メモリに記憶され、一つのコマンドが確定すると、カラーCRT上での処理が開始される。



図2-11 白黒CRTに表示された入力パラメータと動的仮想キーの例

2.4.2 図形の彩色とハッチング機能

本装置では、図形の彩色法として、単色、混合色、シェーディングの3方法を可能とした。ここでいう単色とは、前記8色のうちの一つで、図形全体を一樣に塗ることである。混合色とは、微小領域の中で各画素に異なった色を配し、全体として一樣な多種の色を見せるようにしたものである。ここでは、4×4画素を単位とした色の塗り分けを繰返すようにして、3原色赤緑青の混合比を0から16の数値で指定するようにしたので、 $17^3=4913$ 種の色が表現できる。

一方、シェーディングとは、指定された代表点における混合色データをもとに補間を行い、次第に変化していく色付けを行なう機能である。水平、垂直、斜め、

円、楕円状の5様式を設けた。

又、図形の外觀に変化を与える他の方法であるハッチングを可能とした。これは、前述の記号パターンの周期的繰返しで模様をつけることであり、具体的には次のような処理を行なう。画面上の座標x, yに対し

$$\left. \begin{array}{l} X=x \bmod 16 \\ Y=y \bmod 16 \end{array} \right\} \quad (7)$$

なる演算を行ない、0~15の数値として求めたX, Yで、指定された記号パターンにアクセスする。このときX, Y位置にドットがあれば、画像メモリへの書き込みを禁止する。これによって、ハッチング部分だけ後が透けて見える図形が描かれる。ハッチングと任意の彩色法を組合せて、多様な修飾効果を得ることができるようになった。

2.4.3 特殊図形

装置の試作にあたっては、既述の基本図形と複合図形の他に、文字列、閉図形、線図形の3種を追加した。文字列とは、一字単位ではなく、一連の文字(文章)を一つの図形として扱えるようにしたコマンドである。実行時には一字ずつ繰返し型で処理するが、操作者から見た場合には、文字列全体として回転や移動などの指定ができるため便利である。

閉図形とは、任意数の点列を指定したとき、それらを直線で結んだときにできる図形を扱うためのコマンドである。座標x, yがこの閉領域の内部にあるかどうかを判定する計算体系として、点(x, y)を通る直線が境界線と交差する回数を調べ、その奇偶で決定するいわゆるパリティチェック法[2-8]を用いている。

線図形とは、一つのコマンドで、任意数の点列と点間の線幅を指定して、折線図形を描けるようにしたものである。この場合、各線分を長方形として扱うことによって、前述の図形の計算体系に矛盾なく組込むことができた。更に一歩進めて、この長方形を、線分間の接続状態を考慮した四辺形に置換えることによって、線分の接続部に生じる不連続性[2-5]を解決した。加えて、これらの四辺形の変域を適度に制限することによって、むやみに剣先が長くならないように配慮している。

図2-12に三つの点P, Q, Rと線幅 d_1, d_2 が与えられた場合の、線図形を例示する。図中、着色部が線図形として描かれた「面」である。本方式では、線分に対応する長方形の辺と辺の交点(図2-12ではX, Y, Z)の位置を解析するプログラムによって、線分PQは長方形ABCDを経て四辺形AZXDに、QRは長方形EFGHを経て四辺形EYZHに置換される。一方、各線分の辺域は図中破線で示した長方形abcdとefghである。これらは、各線分の端点からXY方向に、それぞれ線幅の1/2だけ離れた点を内包する長方形として求められたものである。以上の3種の特殊図形も図形の計算体系の条件を満たすように考慮したので、既述の諸図形と全く同等に扱うことができ、任意の彩色やハッチングはもとより、自由な編集が行なえる。

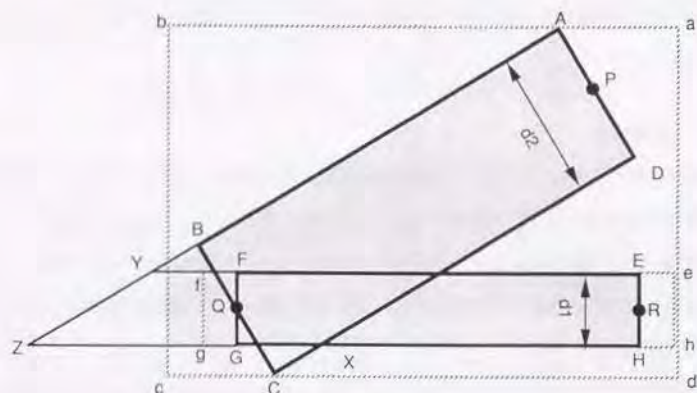


図2-12 線図形の定義

2.4.4 文字品質劣化を改善した補間法

本システムでは文字パターンの倍率変換や回転等の機能を持たせているが、文字パターンはドットパターンで定義されているために、これらの変換時に文字品質の劣化が問題となった。具体的には、濃度値の補間法の良否が変換パターンの品質を左右する。そこで従来の欠点を改良した補間法を開発した。

従来の補間法として、点(x,y)の濃度が $f(x,y)$ として与えられるデジタル画像を座標変換した画像 $g(x,y)$ の濃度値を求める方法として、周囲4点の濃度値を用い(1)式で

演算する線形補間法[2-9]が知られている。

$$\begin{aligned}
 f(A, B) &= (1-a)(1-b) f([A], [B]) \\
 &+ a(1-b) f([A]+1, [B]) \\
 &+ ab f([A]+1, [B]+1) \\
 &+ (1-a)b f([A], [B]+1)
 \end{aligned} \tag{8}$$

ここではA,Bは座標x,yを逆変換したx,y座標系での実数座標、[]はガウス記号、 $a=A-[A], b=B-[B]$ である。

ところが、この補間法を文字パターンに適用すると、図2-13に示すように斜線にくびれを生ずる。この図2-13及び後述の図2-14、2-15は各補間法の特長を比較し、明確にするために、代表的斜線パターンを50倍した場合の結果を示す。図中、破線は原パターンの画素のピッチに対応しており、補間演算で得られる2値化前の濃度値を等高線表示してある。そして、黒く塗られている部分が、0.5をしきい値としてこの濃度値を2値化した結果として得られるパターンである。

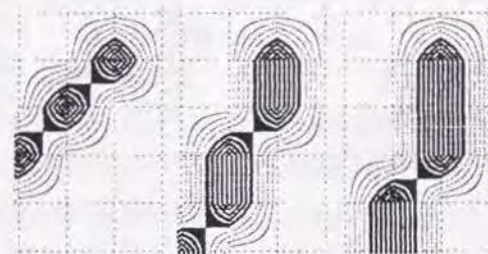


図2-13 従来の補間法の結果

くびれは(1)式がa,bに関し2次式である(水平・垂直線の場合には、0,1の濃度値を代入したとき、2次の項はキャンセルし、実質的には1次式となる)ことに起因する。そこで、対処が比較的単純な45度の斜線の改善に着目した手法Aと、手数はかかるが任意の傾きの斜線に対応できる手法Bの2種を、文字種によって使い分ける補間法を開発した。

(A) 補間曲面を平面化する手法

45度の斜線を検知したとき、すなわち点(A,B)を囲む4点の濃度値が、対角線上で1,他は0であるとき、(8)式を(9)式に置換する。

$$\begin{aligned}
 f(A, B) = & \min \{ (1-a), (1-b) \} f([A], [B]) \\
 & + \min \{ a, (1-b) \} f([A]+1, [B]) \\
 & + \min \{ a, b \} f([A]+1, [B]+1) \\
 & + \min \{ (1-a), b \} f([A], [B]+1)
 \end{aligned}
 \tag{9}$$

(9)式中min[]の演算が斜線の方向に応じた1次式を発生し、鞍型の補間曲面を平面化する働きをする。結果を図2-14に示す。

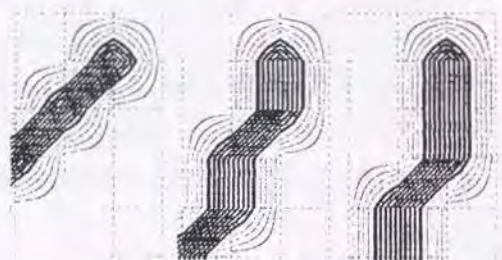


図2-14 手法Aの補間法の結果

(B) 補間前に濃度補正を加える手法

まず原パターンの1,0の並びを、パターン・マッチングによって調べ、各種の斜線パターンの存在を検出し、次のように濃度値の補正を施す。

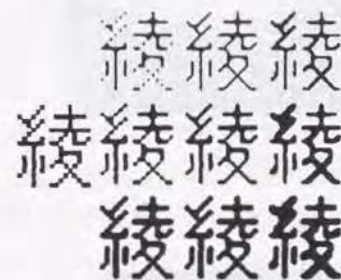
0 0 0 0	0 0 0 0	0 0 0 0	0 : 原画のまま
0 * 1 0	0 0 1 *	0 0 1 *	1 : 原画のまま
* 1 * 0	0 * 1 0	0 # 1 #	* : 0 → 1 / 2
1 * 0 0	0 1 * 0	0 * 1 0	# : 0 → 1 / 4
* 0 0 0	* 1 0 0	0 1 * 0	
0 0 0 0	1 * 0 0	# 1 # 0	他の傾きも同様

その後、式(1)を適用すると図2-15に示すように、実用上問題のない程度の滑らかな線が得られる。実際に2値16×18ドットの文字パターンを4.85倍した結果を図2-16と図2-17に示す。補間後の2値化しきい値は、上から順に0.7, 0.5, 0.3とした。

漢字パターンでは、ほとんどの斜線が45度成分からなること、複雑な線が太くなると見苦しいことから手法Aが、ひらがなでは、45度以外の成分が多いことから手法Bが効果的であることが確かめられた。

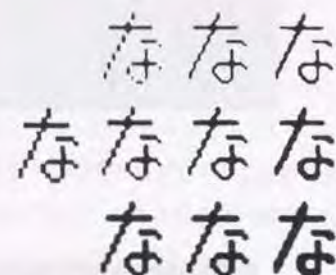


図2-15 手法Bの補間法の結果



最近接 従来法 手法A 手法A

図2-16 適用例(漢字)



最近接 従来法 手法A 手法A

図2-17 適用例(ひらがな)

2.5 結果の検討

試作した装置の外観を図2-18に示す。装置試作の結果、図2-19に示すように、多彩な図形が創成でき、任意に編集できることが明らかとなり、提案した図形編集方式が有効であることが確認された。図形編集で考慮すべき第一の点は、図をどう保存するかにあった。もし一画素一バイトとしてそのまま記憶すると、一画面は256Kバイト(当時のミニフロッピーは一枚分)もの容量が必要である。しかもそのような記憶方法では、描画後の編集は容易ではない。本方式では、図を図形コマンド列として表現する一種のリスト構造としたので、平均的に数Kバイトの容量となった。



図2-18 試作した装置の外観



図2-19 カラーCRT上に表示された作図例

第二の問題点は、パラメータの数の多さであり、図形コマンドを誰もが簡単に入力できる方法を見出すことにあった。本装置では、動的仮想キー方式を採用し、更にスペースキーを押した場合には、自動的にデフォルト値が入るようにした。従って、まず描いてみてその結果で、大きさや位置、色などを所望のものに変更するという作図法がとれる。このように、いちいち操作法を覚えなくてよい上、大まかな図形の配置のあと、細かな修飾をするという方法によれば、極めて効率のよい作図ができることが確認された。

第三の問題点は、ユーザの操作性の向上や編集操作に対する画面変化の違和感の削除であった。本システムでは前述のデータ構造に加え、図形を座標駆動とパラメータ駆動の両形式で操作できるような構造で定義したこと、図形コマンド列の逆方向探索と名付けた手法を用いたことによって、画面上の図形の編集を、操作者の意図どおりに(不自然な画面を出さずに)進行させられるようになった。

一方、本方式のように、ベクトル発生器のような専用のハードウェアを持たず、マイクロコンピュータで画素単位の処理を行なう場合、その処理速度が問題になるが、次のような結果が得られた。例えば単色で半径30画素の円は1.6秒、拡大率1の漢字は0.2秒、図2-5の左上のパターン(正方形と円と拡大率2の漢字からなる複合図形、60×60画素)は4.7秒で描かれる。混合色にすると20%、シェーディングを指定すると、100%程度遅くなる。逆にハッチングを指定すると、ハッチング部における図形の計算体系の演算が省略されるため、20%程度早くなる。又、実際に各種のスライドを作成した例では、ファイルから読出して図の全体を再生するのに要する時間が、10~100秒に分布している。一方、図形の挿入や削除に要する時間は、逆方向探索を行なうために図の複雑さによって異なるが、描画時の1~3倍となる。但し、描いた直後の図形は一時メモリ上にあるため、1秒以下で消去できる。これらの処理速度は、昭和56年頃のマイクロコンピュータ(8086と8087)によって得られたものである。現在のマイクロコンピュータを用いれば数十倍高速になる。

2.6 結言

「面」の概念を基本に置き、図形を座標とパラメータの2入力系ととらえ、これ

ら2入力に対し論理値(内か外か)を出力する計算体系によって統一的な定義を与え、画素単位で処理を行なう新しい図形編集方式を提案した。

この定義のもとでは、図形間の論理接続による複合図形の定義が可能になったばかりでなく、図を構成する図形コマンドの列を逆方向探索するという基本処理の導入によって、図形の削除や挿入などの編集処理が操作者にとって違和感なく、自然な形で実行できることが明らかとなった。

本方式により、卓上型スタンドアロン装置を試作した結果、比較的単純な少数の基本図形の組合せで多様な図形が描けること、編集作業がマンマシン対話で効果的に進められることなどが分り、本方式の有用性を確認することができた。しかし、マイクロコンピュータによる処理では、おのずと性能に限界があり、通常の文字を主体とした簡単なスライド程度ならよいが、図形の数が増える複雑な説明用の図面の場合には快適性の面で問題があるということも分かった。この処理速度の問題に関しては、第4章で述べるイメージプロセッサLSIによって解決することができる。

参考文献

- [2-1] 穂坂衛: "コンピュータグラフィクス", 産業図書 (昭49).
- [2-2] 寺西, 末永: "画面の組込機能を有する英文原稿編集システム", 信学論 (D), J64-D, 2, pp.148-155 (昭56-02).
- [2-3] 南川, 大洞, 矢内, 西村: "図形・漢字入出力機能をもった μ CAI図形編集機能", 情報処理学会第21回全国大会, SL-10 (昭55-05).
- [2-4] 中浜, 田中, 田中: "コマンド表現による図形生成", 同上, 6H-1 (昭55-05).
- [2-5] 久保田, 石井: "線幅を有する線分発生方法に関する検討", 同上, 6H-6 (昭55-05).
- [2-6] 神田泰典: "オフィスオートメーション技術 (1)", 信学誌, 64, 2, pp135-142 (昭56-02).
- [2-7] 藤方, 上田, 江尻: "動的仮想キー方式漢字選択法とその和文ワードプロセッサへの応用", 電学論 (C), 1010-c, 1, p.1 (昭56-01).

- [2-8] Pavilidis, Theo: "Contour filling in raster graphics", Computer Graphics (Proc.SIGGRAPH), 15, 3, p.29 (Aug., 1981).
- [2-9] Rosenfeld: "Digital Picture Processing", Academic Press

第3章 対話型映像編集システム

3.1 緒言

マルチメディアシステムへの期待が高まっている。しかし、特にビデオ、すなわち自然動画像を含むシステムにおいては、従来のキャラクタベースのメディアと比べ、まだまだ使い勝手が良いとは言えない。種々の情報処理技術による、より進んだサポートが必要とされる研究分野である[3-1][3-2][3-3]。例えばマルチメディア・オーサリング・システムでは、従来メディアに比べて極度に量の大きい音声や動画像データについて、ユーザはその内容の把握や前準備のデータ処理に多大な時間を費やしてしまうのが現状である。もちろんユーザ自身が知的で創造的な活動の主体であるが、もし知的オーサリングシステムから適切な援助を受け、煩わしい作業から開放されるなら、ユーザは創造的活動に専念できることになる。

筆者らは、ユーザの創造的活動を活性化するシステムを実現することを目的として、IMPACT (Interactive Motion Picture Authoring system for Creative Talent) と名付けたマルチメディア・オーサリングシステムを開発している。IMPACTは映像信号のリアルタイム取り込み機能を持つワークステーション(シリコングラフィクス社のIRIS 210/GTX)上に構築されている。この研究開発では、画像認識技術に対話環境の中で利用することによる新しいアプローチを進めている[3-4][3-5]。本章では、このアプローチについて詳細に述べ、最後にプロトタイプの結果により、我々の提案するアプローチの妥当性を示す。このシステムの開発と実験的な使用を通して、筆者らは次の3つの重要な研究対象があると考えている。1番目は、システムが時間と空間に広がるマルチメディア情報の構造を解析することである。2番目は、この構造を有効に使えるようにシステム内に記述し、格納することである。3番目が、この構造を理解しやすい形で適切に視覚化(ビジュアライズ)してユーザに提供することである。

検索のサポートを目的としてビデオを解析する研究は近年盛んになりつつある[3-2][3-3][3-6]。またビデオにユーザが付与する注釈(アノテーション)を階層的に記述するという考え方が[3-7]で提案されている。一方、視覚化手法の一つとして、ビデオを扱うための縮小動画像アイコン(micon、マイコン)の考えが、[3-1]に示されてい

る。文献[3-8]では情報の視覚化において情報の詳細な部分から総合的文脈までをなめらかに統合化することの重要性が強調された。そして、文献[3-9]ではビデオの時間軸を段階的に拡大して見せる手法が提案された。今後はこれらの観点を統合化するようなシステムの研究が重要となってくると考えられる。

本章では始めに基本構想を論じ、次いでビデオの構造を自動的に記述するいくつかの知的ビデオ解析機能について述べる。さらに、これらの機能によって得られるビデオの時間・空間構造をいかに視覚化するかについて述べ、最後に、試作結果を示して議論する。

3.2 基本構想

図3-1にビデオ情報の階層構造を示す。最上層はシナリオであり、多くのシーンから構成される全体のストーリーである。各シーンは1個以上のカットから成る。シーンはその構成要素として、サブシーンを持つこともある。この構造においては、上部の層はシンボリックな記述、下部の層はより視覚的な記述から成る。この構造の最も低い部分(すなわちカット)には、生のビデオデータが含まれる。シンボリックな記述は、人や物や場所の名前、そしてそれらの行動や関係などの言語的あるいは意味的な情報から成る。視覚的な記述は、対象物の形状や位置、対象物の運動軌跡、対象物が画面に入った時刻と出た時刻、複数の対象物が出会った時刻というような幾何学的あるいは時間的な情報から成る。

実際に専門家による映画の作成・編集は、まず企画に基づいてシナリオを書き、シナリオからコンテ(代表的な場面を簡単な線画等で表わしたもの)などを作成して全体的なイメージを固め、次にいくつかのシーンに分解して、更にシーンをカットに分解し、最後にカメラを使ってカットを撮影し、それらをつなぎあわせるという手順で作られる。もちろん、このように常にトップダウン的に作業が進むわけではなく、図3-1の階層構造の階層間でフィードバックや試行錯誤も行なわれる。そして創造的な活動は、企画からシナリオ作成の段階のみではなく、この階層間の往復による試行錯誤にも存在するという事に注目する必要がある。

専門家ではないユーザが動画の編集を行なう局面では、この試行錯誤を十分に

支援することが特に重要である。旅行中に思い付くままに撮ってきたものを、後でビデオ旅行記としてまとめるような場合には、撮影した素材が先にあり、これを使えるカットに切り分けて、全体構成を考えながら編集するという完全にボトムアップ的な活動となる。

これらのことから、動画オーサリングシステムでは、図3-1のような構造を一貫性のある形でメタファ化して、それをダイレクトマニピュレーションできるように編集環境を提供する必要がある。またコンテ、シーン、カットなどは、その中が更に空間的に構造化されたものとして定義されるので、それらの内部構造も連続的(seemless)なメタファとして、時間軸構造のメタファとつながっているようにしたい。

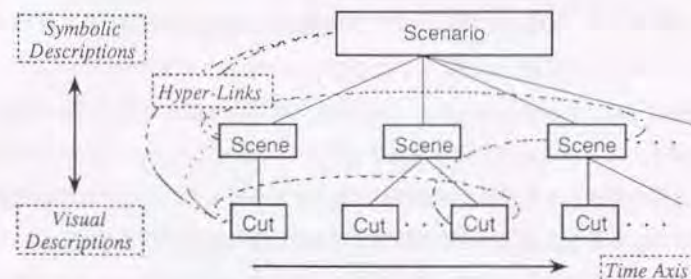


図3-1 ビデオ情報の階層構造

更に、編集作業が図3-1のどの階層レベルにおいてでも可能で、しかもそのレベルで行なわれた編集作業の結果が他のレベルにも適切に反映されるようにする必要がある。その実現には、情報の持つ意味とユーザの意図の解釈をシステムがうまくこなすことができなければ解決できないというような、相当の困難が予想される課題もある。しかし、一つの階層の編集結果が影響を及ぼす範囲を、全体の構造の中で見やすくユーザに提示し、関連箇所のチェックをユーザにうながす程度のことから始め、徐々に役に立つシステムへと育てていくことが重要であると考えられる。その点、マルチメディア処理計算機環境においては、ハイパーメディアとして開発されつつある種々の技術[3-7][3-8]を使うことによって、シンボリックな表現のレベルと画像や音声のような生のマルチメディア・データとの間にリンクを張ることができる

から、関連する部分を自由に参照できる環境を提供することは比較的容易である。

本研究ではこれらの記述情報をできる限りシステムが自動抽出し、さらに、それらの間をハイパー・リンクで結び付けることによって、ユーザに自由で柔軟な編集環境を提供することを目指している。

ところで、ハイパーリンクとは、ハイパーメディア中で関連情報を結び付けるために張られるリンクのことである。MHEG 標準化活動において、ハイパーメディアはマルチメディアの性質を持ち、かつリンクが明示的であることと定義されている。筆者らは特に、この明示的なリンクの意味を次のように解釈している。即ち、映像情報をハンドリングする際には、そのリンクの概念がビジュアルで分かりやすいこと、そのためには従来の画面に貼り付けられたボタンを介しての間接的なリンクでは不十分で、被写体そのものまでもが各種の記述情報と同等に、ハイパーリンクのノードとして定義されることが必要であると考えている。

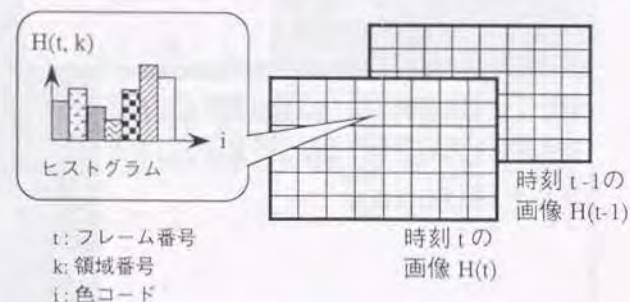
IMPACTは、ユーザとコンピュータが知的活動環境を創り出すために、どのように協調することができるのかを実証するプラットフォームであるとする見てもできる。人間の得意とする分野と計算機が得意とする分野とのバランスの取れた融合を目指すものである。具体的には計算機がビデオ情報の解析を律儀に実行し、その結果を使ってユーザがより高度な判断を下し、ユーザ自身の創造力によって情報を生成・加工するような環境を提供することである。すなわち、ユーザが処理しているビデオ情報の構造を、システムはどこまで認識することができるかということ、そして、そのシステム側の認識結果を、どのようにユーザに提示すれば、ユーザとシステムとの高度な対話環境が実現できるかということが、大きな研究課題である。

3.3 ビデオ解析機能

3.3.1 カット分割機能

従来のビデオ編集では長い映像を、ひとまとまりごとのカットに分割する作業に多大な時間を費やしていた。映像編集をシステムがサポートするためにはカットの自動分割が不可欠である。カットの自動分割方式として種々の提案があるが[3-10]、筆者らは図3-2に原理図を示すように、フレームを8×6の小領域に分割し、次

フレームとの間で、その領域毎の類似度を求め、これがしきい値を越える領域の数をフレーム間相関値と定義している。そして、このフレーム間相関値の時間差分がしきい値を越えたときにカットの変化点であると判定する[3-11]。カットが分割されると、それぞれに縮小動画アイコン(マイコン)を作成し、ビデオ機器から供給されるイン点とアウト点のタイムコードと共に記憶する(後述の図3-9参照)。



$$E(t, k) = \frac{1}{N_c} \sum_{i=1}^{N_c} \left(\frac{H(t, k, i) - H(t-1, k, i)}{H(t-1, k, i)} \right)^2$$

図3-2 ブロック毎のヒストグラム類似度 $E(t, k)$ の計算法
(カット分割の原理)

プロトタイプにおけるカット自動分割実行中の画面例を図3-3に示す(このプログラムは8フレーム/秒の処理速度をもつもので、時間精度として4フレーム程度の誤差が許容できるユーザは、リアルタイム処理として利用可能である)。画面左上のウィンドウには、ビデオテープレコーダあるいは追記型レーザーディスクプレーヤからワークステーションに供給される入力映像(NTSC形式)が表示されている。右上のウィンドウの下部には、横軸を時間としてフレーム間相関値が表示されている。そして、上部のグラフがその時間差分値を示している。この差分値がしきい値を越える時点が新しいカットの始まりであると判定し、そのフレームのタイムコードを記録し、入力映像のその瞬間の画像を縮小したものを、アイコンとして下のウィンドウに順に並べて行く。また、このアイコンは奥行き方向に厚さを持つ立体として表示し、この厚さは次の変化点までの時間、即ちカットの継続時間を視覚的に示すよ

うになっている。

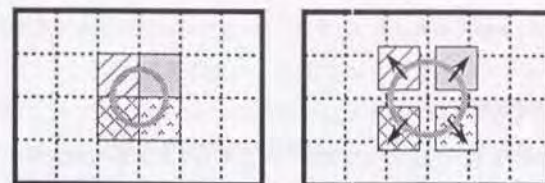


図3-3 カット自動分割の画面例

3.3.2 運動解析機能

IMPACTに実装した運動解析機能は図3-4に示すように、まずに分割した部分画像をテンプレートとして、パターンマッチングによって次フレームから対応点を探索し、その移動量(オプティカルフロー)を求める方式である。ズームの場合はオプティカルフローが放射状になり、パンの場合は平行になること、そして被写体の移動では局所的領域における平行移動となることを利用している。この機能により、カメラあるいは被写体(対象物)の運動の記述が得られる。図3-5に実験結果の例を示す。

なお、論文[3-5]の報告では、この処理はオフライン処理として利用することとしていたが、その後の実験の結果、640×480のすべての画素を使って演算する必要はなく、80×60画素に間引いた画像で処理しても同等性能が得られることが確認された。また、この演算に利用する画素数の切りつめによって、画像をメモリ上に配置できるため、トータルで100倍の高速化を図り、現在のIMPACTではオンラインで利



(a) ズームの検出



(b) 被写体移動(またはパン)の検出

図3-4 動き検出の基本原理(ブロックマッチング法)

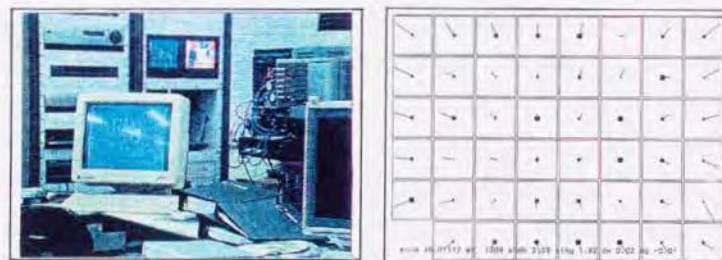


図3-5 動き検出の実験結果

用できるようになっている。

3.3.3 対象物抽出機能

対象物抽出機能はプリミティブな画像処理関数群を用意しておき、それらを組み合わせ対話的に処理手順とパラメータを動的に生成する方式としている。静止画に対してはエッジ保存平滑化と輝度成分による領域分離を中心とする処理手順を、そして動画に対しては連続する3フレームの画像の差分を2値化して得られる2つの画像の論理積を取ることを基本とする処理を用いる[3-5]。動画中の移動する対象物の抽出機能を図3-6に示す。この例は対象物である赤色の自動車がある画面の左側から入り、右側に出て行くまでの約2秒間の屋外シーンである。図4(a)、(b)、(c)は時刻が6フレームづつ進んだ3つの入力画像である。まず画像(a)と画像(b)の差分を求め、エッジ強調処理を施した後2値化したものが(d)の対象候補領域である。次に画像(b)と画像(c)について同様の処理を行い(e)を得る。そして(d)と(e)のAND処理により対象候補領域の共通部分を探し、膨張・収縮処理を行うと対象抽出領域(f)が得られる。そして最後に(b)に対して(f)を用いたマスク処理を行った結果が(g)である。この処理を全フレームについて実行する。なお、このプログラムの処理速度も0.3フレーム/秒なので、オフライン処理として利用する。

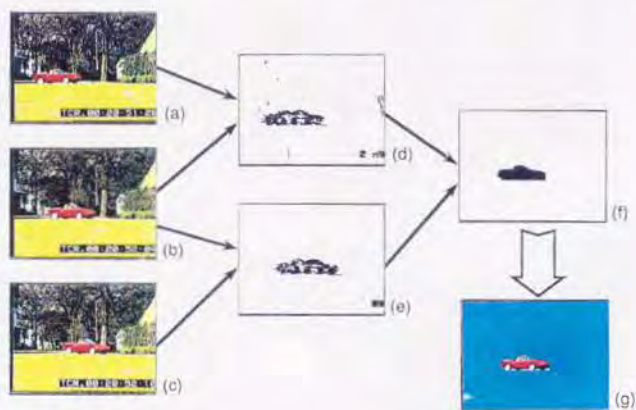


図3-6 ビデオ画像からの対象物の切り出し例

この機能は処理速度が遅いのでオフラインで利用するが、動画については、ユーザが最初の数フレームの結果をチェック(必要なら処理パラメータを調整)した後は、残りの数百のフレームについては自動処理できる。静止画像からの対象物抽出では対象物の輪郭線が得られる。そして、動画中の移動対象物抽出機能からは対象物の輪郭線と運動の軌跡データ、および対象物が画面に現われた時刻と画面から消えた時刻(それぞれはタイムコードで管理され、両者を合わせて対象物存在期間と呼ぶ)が得られる。これらは、後述する映像の時間・空間的記述の重要な要素となる。

対象物の輪郭線のデータは、2つの映像を合成する際のキー信号(マスク)としても用いることができる。図3-7に映像合成を行った例を示す。輪郭線を精密に求める処理を行って、アンチエイリアシング処理を行えば、より完全な合成となるが動画像として見せる場合には、このままでも輪郭の不自然さはあまり気にならないようである。従来のクロマキー方式では青い大きな背景を用意する必要があったために、ここに例示したような屋外を走る自動車等には適用が困難であった。しかし、このように画像認識を応用することで、画像合成に利用できる対象物の範囲が広がる。そして、一般ユーザでも気軽に映像の合成を行うことができるようになった。



図3-7 切り出し画像による合成映像の例

この情報は画像合成のキー信号として用いる以外にも、例えばユーザが、「赤い自動車が左から右に横切ったカット」というような検索を行う場合に利用される。但し、現時点でのプロトタイプでは、抽出された赤い対象物が自動車であるという属性を別途対話的に与えるか、検索条件を「赤い対象物が…」と言い換えて検索する必要がある。

また、この情報は動画の構造を視覚化したインタフェースとしても活用できる。図3-8にその一例を示す。これは図3-3で示したカット・アイコンを拡大したもの

に相当し、直方体の最後面にも縮小画像を表示し、移動物体の軌跡をその中に赤い線で表示している。そして、直方体の中ほどに表示している画像は、マウスによって時間軸方向に移動される一種のカーソルである。ユーザはこのビジュアルインタフェースにより、移動物体の軌跡を参考に、立体中の1点をポインティングして時刻と空間位置を指定したり、2つの移動物体が交差するフレームを見付けたりできる。また2つのカットを合成する時の被写体の関係等の詳細なタイミングを検討するような局面でも利用できる。

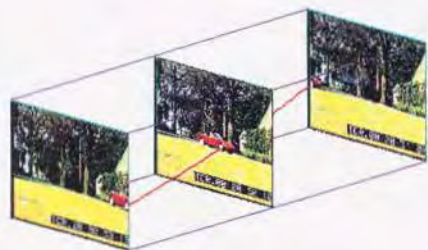


図3-8 カットの内容を操作するためのメタファ

3.3.4 対象物存在判定機能

対象物存在判定機能は文献[3-6]で提案されたアルゴリズムを基本とするものである[3-12]。図3-9に原理図を示すように、映像中のあるフレームについて、ユーザが対象物を指定すると、残りのフレームの中から、同じ色の組合せを持つものを全て自動抽出することができる。これは画像を小領域に分割した上で、隣接する領域に含まれる色の対が、指定された対象物が持つ色の対と一致するかどうかを検定する方式である。この機能では、カット中の対象物の有無、対象物存在期間、対象物のおおまかな(上述の領域の大きさの精度の)形状・位置情報が得られる。

3.3.5 オブジェクトリンク

3.3で述べた対象物の抽出法では、対象物の正確な輪郭形状や位置の情報は得られないが、他のカット中に存在する対象物と同一物であるかどうかは、人間が判断する

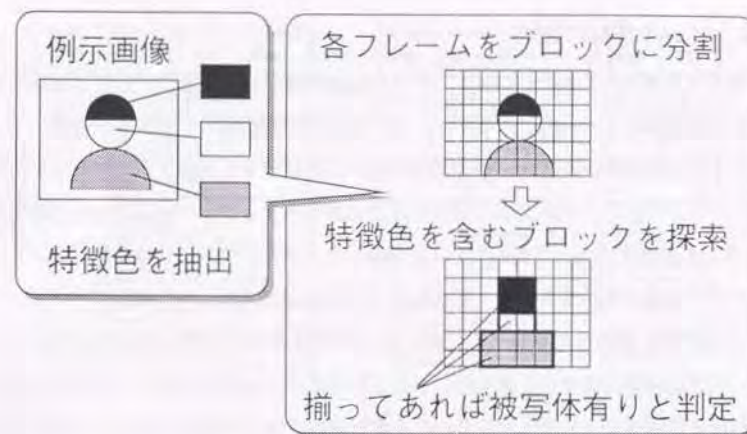


図3-9 被写体検索の原理図

必要があった。ところが、3.4の対象物存在判定機能を用いれば、各カット中の特定の対象物の有無に関する自動記述が可能となる。言い換えれば、この機能によって各カット中に存在する対象物間にハイパーリンクを自動生成することができる。

従来のハイパーリンクのノードは、単に映像中のキーフレームであったり、このキーフレーム上に貼り付けられたボタンアイコンであった。しかし、ここで提案するリンクのノードは被写体であり、ユーザの感覚から言っても、それ自身が本来的に種々の(意味)情報を持っているものである。従来のリンクと、ここで提案するリンクとの実現法の違いは、ビデオ情報の構造の視覚化とハンドリング技術に大きな進展をもたらすと考えられる。それゆえ筆者らは、これをオブジェクトリンクと名付けた。オブジェクトリンクとは、オブジェクト(被写体)をもノードとし、ノード間の関連情報を属性として持つハイパーリンクである。これによって、被写体を中心とした映像の情報構造の記述が可能となり、マルチメディア対話環境の実現のための強力なツールとなると考えている。このリンクと上述の各種の映像解析機能で得られる時間・空間構造記述を組み合わせる(図3-1参照)ことによって、4.1で述べるように効果的な映像情報構造の記述を実現することが可能となる。

3.4 ビデオ情報の記述

視覚化を有効にするためには、ビデオの階層構造を適切に記述することが重要である。MITのDavenportのグループは、ユーザが意味的情報(カメラ位置、照明、アクション及び解説等の構造)を、構造的な変数としてビデオに付加するシステムを開発している[3-7]。このシステムではユーザが付加するアノテーションをstrata(断層構造)として表現することを試みている。すなわち、異なる視点で付けられたアノテーションがフラットに層を成して積み重ねられる構造である。

これに対し筆者らのシステムでは、ユーザが付加したアノテーションのみならず、ビデオの解析機能から自動的に供給される記述の視覚化を重視している。筆者らの記述の基本は、自動カット分割と対象物の存在判定の結果であり、分割されたカット毎の各種構造情報が、まず図3-10に示すような記述として格納される。イン点、アウト点のタイムコードとマイコン画像もここに格納されるが、運動解析機能で得られるカメラや対象物(被写体)の運動データはmotionポイントで指示される領域に格納される。そして対象物抽出機能によって得られる対象物の存在期間は、extractionポイントで指される領域に格納される。フレーム毎に得られた輪郭線(これから運動軌跡も分かる)データは、さらにその先のe-contourポイントが指す領域に格納される。annotationポイントが指す領域には、ユーザがキーボード等から入力するそのカットへの注釈が格納される。

一方、対象物存在判定機能によって得られる情報は、図3-11に示す形式で格納される。対象物存在判定を行う度にこのリストは右の方に延びていく。判定は通常複数のカットに対してなされるので、そのカット毎に得られた判定結果が、カットポイントの先のリストとして並ぶ。カット番号を格納しておくことにより、図3-10の記述情報との対応をとる。ここからオブジェクトポイントで指される先に対象物存在判定機能で得られた存在期間と大まかな形状・位置情報が格納される。対象物が入り出すために、1つのカット内で複数回見つかることがあるのでリスト形式としている。また、対象物抽出機能が、この対象物に対して起動された場合には、それとの対応をとるために、extract番号をここに格納しておく。

オブジェクトリンクは、図3-3に示したように同一対象物間には自動的に張られ

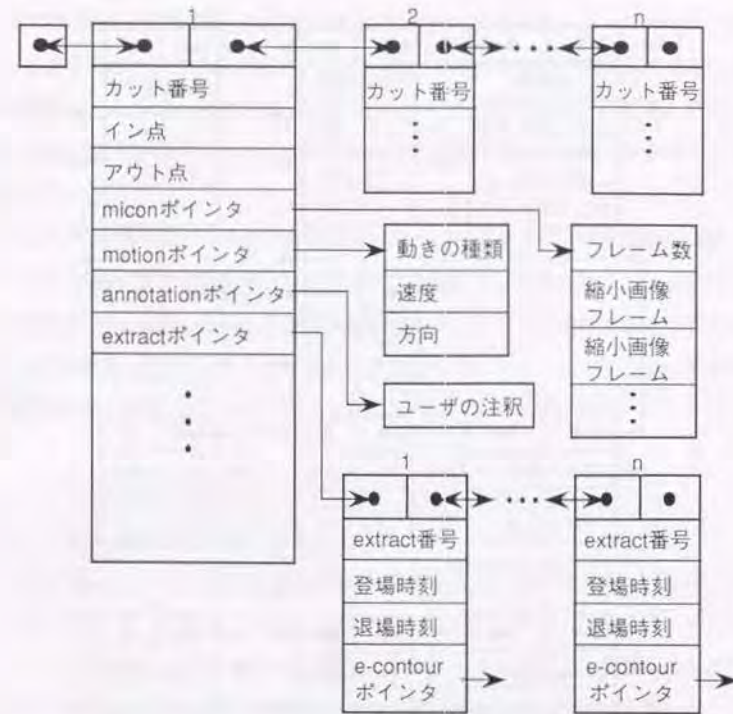


図3-10 映像構造の記述

る。そして、このプロトタイプでは、複数のカットを代表する図3-10の最上位のリストの要素(複数カットに登場する被写体の集合に相当)と最下位のリストの要素(各カット内の特定の状況にある被写体に相当)の2種類のオブジェクトにはポイント格納領域を設け、ユーザはここから任意のオブジェクトにリンクを張れるようにしている。

ところで、図3-10と図3-11の時間・空間的な記述は互いに補完的なものもあるので、これらを統合する処理を行う。対象物の位置・形状情報(e-contourとo-contour)については、対象物抽出機能から得られる情報の方が対象物存在判定機能から得られるものより精度が高いため、前者が存在する場合にはそれを優先的に利用する。

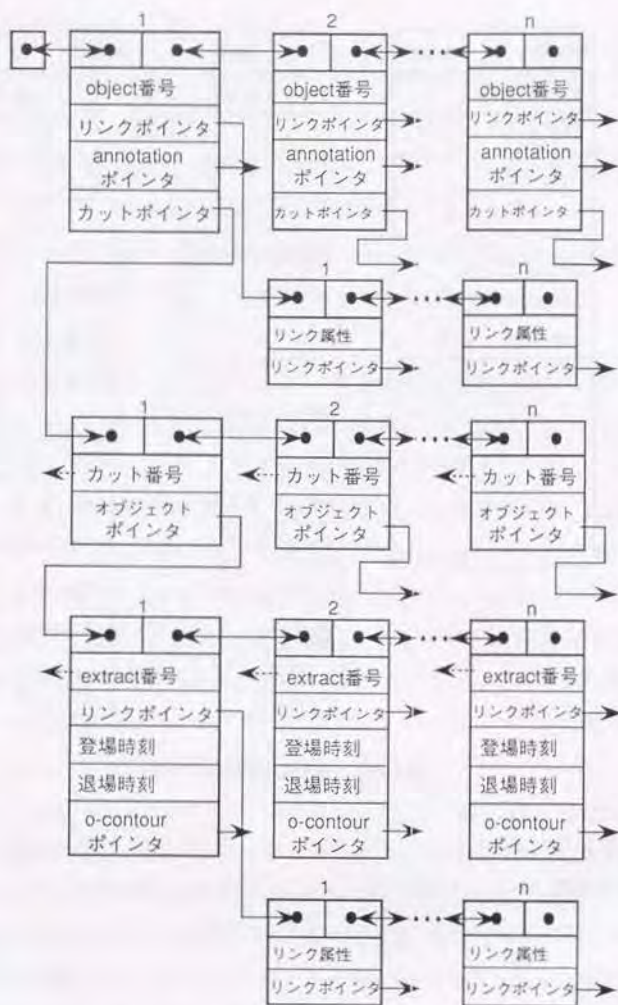


図3-11 オブジェクトリンク情報の記述

もし、両者が大きく食い違う場合には、ユーザに警告を出し、実画像と輪郭線を画面に表示してチェックを促す。対象物存在期間(登場時刻と退場時刻)についても同様である。ユーザには、それぞれの時刻の画像を見せてチェックしてもらう。

対象物間に自動的に張られたハイパーリンクであるオブジェクトリンクは、自動的に、あるいはユーザとの対話によって、図3-1に示した映像情報構造の上方の層にまで拡張される。ユーザはダイアログ・ウィンドウによりこれらの記述を修正したり、新規の注釈を加えることができる。もちろん編集時に、これらを参照することができる。

また、編集や合成に伴う構造記述の変化はオブジェクトリンクを通して伝播し、変化分に対応する他の記述やアノテーションを自動的に調整したり、あるいはユーザに対してチェックを促し、必要な修正を行なうようアドバイスするのに利用される。ユーザは、オブジェクトリンクを用いたナビゲーションにより、カット属性の効率的な検索を行なうこともできる。

3.5 ビデオ構造の視覚化

3.5.1 時間軸編集の基本画面構成

図3-12は、ユーザがビデオの時間軸編集をするときの画面例である。画面下方のコンポーネント・ウィンドウは、自動的に分割されたカットの視覚化リストを示す。カットは動画アイコン(micon、マイコン)によって表される。筆者らのシステムでは各マイコンを直方体として表示し、その奥行きでカットの継続時間を示している。この表現法により、カット分割機能によって得られるビデオの時間構造をコンパクトに視覚化できた。更に、いくつかのマイコンには、ビデオ解析機能によって得られた被写体やカメラの動きの情報が着色された矢印としてスーパーインポーズされている(矢印の色でその区別を行ない、さらにこれらの動きが不適切である場合の警告を行なう)。従ってユーザは、単にこの視覚化リストを一目見るだけで、ビデオの時間・空間的な全体の構造を直感的に理解することができる。

従来のビデオ編集では、図3-12のコンポーネント・ウィンドウ中に示されるようなビデオの情報構造が頭の中にはっきりと構築されるまで、ユーザは早送りと巻き戻しを繰り返す必要があった。そしてこの作業は非常に時間のかかるものであった。しかし、本プロトタイプシステムでは、画像認識技術を応用することによって、この時間のかかるステップを不要とし、カットの取捨選択と並べ替えによる実

質的な編集に、いきなり取り掛かれるようになった。この編集作業は、コンポーネント・ウィンドウから右上の編集ウィンドウへのマイコンのダイレクトマニピュレーションによって簡単に実行できる。このユーザ操作に対するビデオ機器の制御は、各カットに付与されているタイムコードを使ってシステムが適切に実行する。

ところで、本システムではユーザはマイコンの一覧表示によってビデオの情報構造の把握が可能であるが、個々のカットの中の人間の行動や物の移動が意味するところなど、より上位の意味情報は、ユーザが実際に映像を(部分的に)再生して判断する必要がある。スクリーンの左上のウィンドウはそのために設けたモニターである。ユーザは任意の一つの、あるいは選択した一連のカットを、ここで再生することができる。



図3-12 時間軸編集の画面例

3.5.2 オブジェクトリンクによるナビゲーション

オブジェクトリンク・ナビゲーションと名付けた機能をユーザが呼び出すと、図3-13の右側に示されるようにオブジェクト・インデックス・ウィンドウが表示され

る。このウィンドウ左部のアイコンは対象物(オブジェクト)の索引である。ユーザがある対象物が存在する箇所を捜すために、索引アイコンのひとつをクリックすると、先に図3-11を用いて説明したオブジェクトリンクによって対応付けられている、コンポーネント・ウィンドウ中の箱状のカットマイコンの側面が帯状に着色される。そしてこの帯の位置と幅が、そのカット中での対象物の出現時刻と存在期間を表現している。また、それまでカットの先頭フレーム(あるいはユーザが指定した代表フレーム)の画像の縮小画面で表示していたマイコン画像を、対象物が出現したときのフレームの縮小画像に変え、その表示位置も、出現時刻に対応させて箱状アイコンの奥まった所に表示するようにしている。

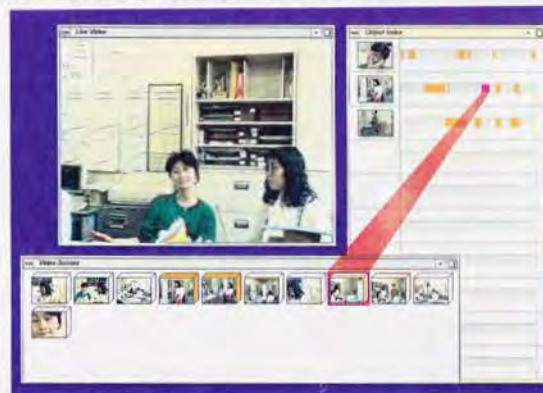


図3-13 オブジェクトリンク・ナビゲーションの画面例

オブジェクト・インデックス・ウィンドウの右部の棒グラフは、ビデオの時間軸における対象物存在期間を示す。従って、この表示を見ることでユーザは対象物の存在の相互関係を一覧できる。また、薄い垂直線がカットの切れ目を示しているので、あるカットの中に、どの対象物が共存しているかというようなことも簡単にチェックすることができる。さらには、ユーザがそのような棒グラフのひとつをクリックすると、コンポーネント・ウィンドウの対応するカットアイコンに向けて誘導表示がなされる。なお、このオブジェクトリンク・ナビゲーション機能は、編集ウィンドウに対して動作させることも可能である。

ユーザは、これらの機能を使用して、直感的にシーンの構造とビデオのストーリー

の流れを概括したり、詳細なカットの記述を自由にブラウジングしたりできる。これらの結果としてユーザが理解した内容は要約して、図3-1における上位の記述として追加される。この一連の流れはオブジェクトリンク・ナビゲーションによってスムーズに進行する。またこの上位の記述のためだけではなく編集に必要なカットを選び出すためにも、このブラウジングは重要な役割を果たす。

3.6 実験結果

本章では、このプロトタイプシステムを使ってビデオ編集を実験的に行った結果について述べる。開発にあたったメンバーの中から3人が被験者となった(多数の被験者を用いた大規模なテストは今後の課題である)。ビデオ素材としては、ある程度のバリエーションを考えて以下に示す5種類を用いた。

- (1) テレビ放送から家庭用ビデオ (VHS 方式) に録画した自動車レース (約30分)
- (2) テレビ放送から家庭用ビデオ (VHS 方式) に録画したクイズ番組 (約30分)
- (3) 家庭用ビデオカメラ (8mmビデオ方式) で録画した家族のハイキング (約5分)
- (4) 業務用ビデオカメラ (ベータカム方式) で録画した庭園内の人物や動植物 (約5分)
- (5) 業務用ビデオカメラ (ベータカム方式) で録画した事務室内の仕事風景 (約5分)

3.6.1 個別機能について

先に述べたように本プロトタイプでは映像の時間的構造の視覚化を可能とするために、ビデオカメラ等で撮影された映像を自動的にカットに分割する機能を開発した。このカットの自動分割機能は、従来手法のアルゴリズムを人間の感覚(たとえ動きは早くともコンスタントな速度でカメラや物体が動いている場合にはシーンの変化はないと判断する)に合うように修正したものである。つまりユーザが直感的にあるいは何らかのラフなモデルで推測して使えるようにする工夫である。具体的には変化量の時間差分を求めるようにしたことによって、これを可能としている。この改良の効果を図3-14に示す。図3-3の右上ウィンドウと同じく下に示したグラフが相関値であり、上がその相関値の時間差分値である。相関値をそのまま用いた場合に

はパンやズームによって、三角形状のグラフの変化(即ち、コンスタントな速度を持つ変化)が多発し、そのピークは本来のカット変化点で生じる相関値の大きさに匹敵することがあるために誤検出が生じるが、相関値の差分については、これらは小さく納まり、カット変化点での値は、そのままの大きさを保っていることが分かる。

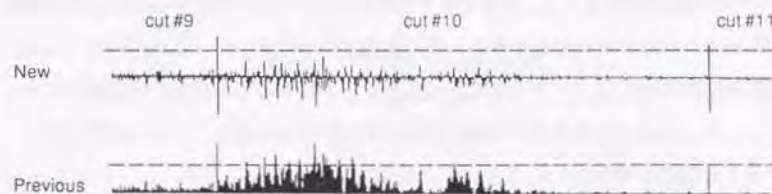


図3-14 カット分割アルゴリズムの従来手法との比較

相関値をそのまま使う手法でのしきい値T2は、対象画像が変わる毎に大きく調整する必要があったが、差分を用いる本方式でのしきい値T3は、屋内、屋外等の撮影条件によらず、非負でとり得る値のはほぼ中央値に固定したままで望ましい動作をするようになった。なおT1のデフォルト値も取り得る値の中央値とし、調整は不要であった。また、ユーザがこの機能を使うにあたって、その理解を助けるために、変化量をグラフで表示し、変化点検出のスレッシュホールドT1とT3の変化に対応して検出感度に変化する様子をリアルタイムで見ること(フィードバック)ができるようにしている(図3-3、図3-14参照)。また、アルゴリズムの詳細まで知りたいユーザには、差分を取る前の相関値のグラフと、それをそのまま用いた場合の変化点検出結果を見せることもできる。

一方、カメラのパンやズームの解析に関して、撮影時の速度むらの補正という観点での実験は、従来なされていなかったようなので、いくつかの実験を行ってみた。屋外、屋内を含む10種の動画像について適用した結果、この方式は表3-1に実験例を示すように人間の感覚に匹敵する十分な検出精度を持ち、前述したパン・ズーム速度のむらの補正にも適用できる見通しが得られた。なお表中の真値とはフレーム画像のハードコピーを用いて実測したものである。

カット分割機能については、不要なカット(いわゆるNGカット)を取り除く作業

が、快適に行えることが、本システムの大きなメリットであることが確認できた。しかしながら、ユーザの撮影時の意図の違い等によって、標準的なしきい値判定では、カット分割の結果がユーザにとって満足行かないケースもあることが分かった。上記のビデオの中で、子供達の動きの激しいところなどで分割過剰が約1時間15分の中で2箇所存在した。また前カットの最後のフレームと次カットの最初の画面が似通っているために分割できないという現象が15件あった。これらについては、感度調整機能付きのアンドウ機能を用意することによって、対話的に対処できるようにした。なお、原理的にこの機能では、ゆっくりとしたディゾルブに対しては、カット分割できない。

表3-1 バン検出アルゴリズムの評価結果

速度	真値 (画素/フレーム)	検出値 (画素/フレーム)	誤差
低	1.28	1.36	12.3 %
中	5.75	5.52	4.0 %
高	22.20	22.96	3.4 %

対象物存在判定機能については、特定の人物や車両が登場しているカットを一覧表示できる機能が、ストーリーの流れを確認したり、ダイジェスト版を作成するような場合に役立つことが確認された。性能評価の結果を表3-2に示す。これはサンプルビデオの(2)と(5)、それぞれ694、284フレーム分について前もって目視によって対象物の有無を判定したものを使用し、対象物としては青いシャツ(A)、緑のセータ(B)、白いブラウスに赤白の花柄のスカート(C)の人物について実験した結果である。利用者からみて致命的となる検出漏れは、これらのサンプルビデオに関する限りゼロであった。過剰検出が1~17%でやや大きいのが、これについてはユーザが対話的にリンクを加えたり削除したりできるインターフェースを使って、手作業で修正することが可能である。

もっとも、対象物存在の判断が色の組み合わせに基づいているので、当然この機

能は類似した色の組み合わせを持っている対象物、あるいは無彩色の対象物に対しては誤検出を生ずる。上述の結果はそれを理解しており、比較的是っきりした色の対象物について存在判定を行うというユーザを前提とした実験結果である。存在判定が色のみでは困難なケースについては、次のステップでは音声による話者識別などを組み合わせて、性能向上を計る予定である。

表3-2 対象物判定機能の性能

		ビデオ(2)	ビデオ(5)	
		対象物 A	対象物 B	対象物 C
正 認 識	対象物有	63	62	32
	対象物無	631	197	205
誤 認 識	検出漏れ	0	0	0
	過剰検出	6	25	47
トータル・フレーム数		700	284	284

静止画像および動画からの対象物の分離機能は、表3-3に示すように前者は11ステップ、後者は9ステップの処理手順になったが、途中の処理結果を見ながら、処理パラメータをチューニングすることは、それ程困難なものではなかった。また各パラメータの値の変化が処理結果におよぼす効果は、直感的に判り易いものが多いので、初心者にもなじみ易いと思われる。そして各ステップでの処理は、17種類の各種フィルタリングと加減、論理演算等の基本関数を組み合わせることによって実現できた。静止画像処理では、関数を合計22回呼び出し、使用した関数の種類数は16種、動画の方は合計19回呼び出し、使用した関数の種類数は11種であった。このように前者と後者の処理手順は異なるが、使用する基本関数はほとんど共通であった。

このことから対象物が変わっても、基本的な画像処理関数はあまり増やすことなくその組合せ方やパラメータのチューニングを工夫することで容易に対応できる可能性が高いと考えられる。従ってユーザインターフェースを工夫すれば、画像処理

の専門家でも容易に関数を組合せ、そのパラメータを対話的に調整できると思われるので、次期プロトタイプで、その実現を図る予定である。

表3-3 対象物の分離に使用した関数の種類と使用回数

	関数	静止画像中の 対象物分離	動画中の 対象物分離
1	最大値フィルタ	2	0
2	最小値フィルタ	1	0
3	平均値フィルタ	1	0
4	2値化処理	2	2
5	Sobel フィルタ (x方向)	1	2
6	Sobel フィルタ (y方向)	1	2
7	加算 (画素ごと)	1	4
8	減算 (画素ごと)	0	2
9	マスク (画素ごと)	1	1
10	NOT 処理	3	0
11	膨張処理	2	1
12	収縮処理	1	1
13	小面積領域除去 (白)	1	2
14	小面積領域除去 (黒)	1	0
15	AND 処理	2	1
16	OR 処理	1	1
17	EOR 処理	1	0

3.6.2 編集システムとして

このシステムでは、ユーザは従来のテープベースの編集と比較して、はるかに少ない回数しかモニターウインドウでの再生を使用しないという興味深い現象が観察された。対象物存在判定機能により、登場人物や物を探すためにモニターを目で見る必要がないことが最大の理由であるが、それ以外の局面でも大きな差異が認められた。まずカットをつなぎあわせていく過程で、従来のテープベースの編集では、編

集者が何度もモニターで再生してみて、そのつながり具合を確認しては別のカットの組み合わせを試すということを繰り返すのに対し、本プロトタイプでは、ほとんど一回でつながりが決められる。これはシステムが提供する時間・空間構造の視覚表現を見ることによって、ユーザが実際の再生時の様子を簡単に予想することができるからであると考えられる。例えば俗に言う壁塗りパンとなるカットつなぎや、不適切なパン・ズーム速度、カット継続時間の変化から作られるリズム間などが、このカットマイコンの厚さや矢印の視覚表現だけで判断できてしまうことの効果である。

またモニターでの再生による確認が減るという現象は特に自分が撮影したビデオの場合に顕著であり、それは人物の登場する順序や移動といった下位の情報だけで、容易に撮影時の状況を思い出せるからである。他人が撮影したビデオであっても、一度通して全体を見た後であれば、ほぼ同様のことが可能となる。そしてその時点で適切な上位の意味情報をアノテーションとして付けておくべきである。それらのことをせずに、興味のあるカットのみをランダムに再生して、欲しいものを探そうということを試みたユーザは、丁度トランプゲームの神経衰弱をしているような状況になることもあった。これも興味深い現象である。

これらの観察結果を結び付けて考察すると次のようなことが言える。自分の意図で撮影したものを編集するユーザ、あるいは他人が意図を持って撮影したものを一度全体を通して見たユーザは、例え撮影する順序が最終的な作品の順序と一致しなくとも、また同じ場面の撮り直し等が数多く存在しても、編集に際してそれなりの自分の意図を持つことができる。これに対し、他人の撮影した映像については、単にカットマイコンの一覧表示を見るだけでは、人や物の存在や動き、カメラ、そしてカメラレンズの動きという下位の情報だけで、その上位の意味の記述がないために撮影者の意図までは理解しにくい。従って何をどうつないで何を表現するかという編集意図に合うカットを、容易には見つけられなかったのだと考えられる。

先に述べた神経衰弱の状況は、実はカットをランダムに並べておいて、被験者には本研究を紹介するためのデモビデオを作成するという編集意図を与えたケースの中で観察されたものである。従って編集者と撮影者は別人で、また撮影者がシナリ

オを知っていて撮影したものではない。これらのことから、撮影者と編集者が事前にシナリオをどこまで共有しているかということや、上位の意味の記述の有無は、編集過程でのユーザの活動のパターンに大きく寄与すると思われる。

筆者らは、今後本論文で提案したシステムが普及すると、ビデオ素材(汎用的に使えるカット)を大量に提供するような業種も発生すると予測している。それ故、そのような汎用ビデオ素材には何が付加的情報として記述されるべきかということが重要な問題になると考える。今回の実験では時間的に、上位の記述としてのアノテーションを、一定数以上のカットにつけるというレベルまでの実験ができていない。もっとも、これを適切に行うためには、ビデオ作品作りのための知識や、ある程度の慣れが必要なようであることが分かってきた。この面を系統的にどうサポートするかということも今後の課題である。また、最初からシナリオがあって撮影と編集を別の人が分担するというような場合には、状況が大きく変わるので、このような局面での本システムの評価も改めて行う必要がある。今後実験と検討を続け、よりよいビデオ情報の記述方式、視覚化方式とヒューマンインタフェースの実現に結び付けて行きたい。

3.7 結言

ビデオの情報構造を解析して、その結果をビジュアルライゼーションするための手法を開発した。解析機能は、カット分割、カメラと被写体の動き、被写体の軌跡や輪郭線、特定の被写体の存在の判定及び存在の期間を構造記述として提供する。更に、被写体の存在判定機能により、同じ被写体はビデオ全体を通じて自動的にリンクされる。このようにして、得られるリンクをオブジェクトリンクと名付けた。そして、このオブジェクトリンクとビデオ解析結果を組み合わせることによって新しいビジュアルライゼーション手法を提案した。このビジュアルライゼーション手法は、ユーザが生のビデオデータのみならず、ビデオ情報の構造、そしてこれに付加される記述内容を容易に理解できるようにした。その結果、ダイレクトマニピュレーションによってビデオ情報を、自由に編集・修正したり、ブラウジングしたりすることを可能とした。

但し、このプロトタイプでもワークステーションの性能不足が顕在化した。カットの自動分割や被写体の存在判定に関しては、アルゴリズムを工夫することにより、リアルタイム処理を可能としたが、動きの解析や対象物の切り出し等の処理を精度良く実行するためには、オフライン処理とせざるを得なかった。NTSCビデオ方式でも数十倍の高速化が、今後普及すると考えられるハイビジョンでは、200~300倍の高速化が必要である。

第4章では、これまでの章で明かとなったパソコンやワークステーションのマルチメディア処理能力の不足をカバーするためのイメージプロセッサ研究について述べる。

参考文献

- [3-1] Mackay, E. W. and Davenport, G.: "Virtual Video Editing in Interactive Multimedia Applications," *Communications of the ACM*, 32(7), pp.802-810 (1989).
- [3-2] 上田博唯: "インテリジェント映像編集技術," *インテリジェント映像メディアフォーラム'92*, テレビ学会, pp.13-21 (1992).
- [3-3] 坂内正夫: "画像・マルチメディアデータベース," *テレビ誌*, Vol.46, No.11, pp.1474-1479 (1992).
- [3-4] Ueda, H., Miyatake, T. and Yoshizawa, S.: "IMPACT: An Interactive Natural-Motion-Picture Dedicated Multimedia Authoring System," *CHI'91*, ACM, New Orleans, pp.343-350 (1991).
- [3-5] 上田, 宮武, 吉澤: "認識技術を応用した対話型映像編集方式の提案," *信学論*, Vol.J75-D-II, No.2, pp.216-225 (1992).
- [3-6] Nagasaka, A., and Tanaka, Y.: "Automatic Video Indexing and Full-Video Search for Object Appearances," *Proc. IFIP WG 2.6 2nd Working Conference on Visual Database Systems*, pp.119-133 (1991).
- [3-7] Davenport, G., Aguiere-Smith, T. and Pincever, N.: "Cinematic primitives for multimedia," *IEEE Computer Graphics and Applications*, 11(4), pp.67-75, (July 1991).

- [3-8] Card, S. K., Robertson, G. G. and Mackinlay, J. d.: "The Information Visualizer, An Information Workspace," CHI'91, ACM, New Orleans, pp.181-188 (1991).
- [3-9] Mills, M. Cohen, J. and Wong, Y. Y.: "A Magnifier Tool for Video Data," CHI'92, ACM, Monterey, pp.93-98 (1992).
- [3-10] 大辻清太, 外村佳伸: "映像カット自動検出方式の検討," テレビ学技報, Vol.16, No.43, pp.7-12 (1992).
- [3-11] 宮武, 吉澤, 上田: "フレーム相関係数の変化率に着目したカットの自動検出法," 信学秋季全大, p.6-301 (1990).
- [3-12] 長坂, 宮武, 上田: "動画像情報リンク付けの一手法," テレビ学技報, Vol.16, No.10, pp.25-30 (1992).

第4章 画像処理用高速プロセッサ

4.1 緒言

4.1.1 動機と目的

近年の半導体技術の進歩に支えられて、種々の技術が今までは考えられなかったような分野へと、その応用を拡大しつつある。画像処理もその例外ではなく、かつては研究機関でぜいたくに計算機を使える人々だけにしか縁のなかったような、かなり高機能の視覚情報処理が、OA(オフィスオートメーション)やFA(ファクトリーオートメーション)という言葉で象徴されるように、オフィスや工場に設置される機器の中に、妥当な価格で組み入れられる時代になろうとしている。

筆者らは十数年来、画像処理の研究を続け、その中でいくつかの視覚認識装置を実現してきた。第2章と第3章では、それらを通じて得られた技術を対話型マルチメディアハンドリングに拡張・応用する研究成果について述べた。しかし既製品の半導体素子を用いた装置の開発に頼っているだけでは、我々の理想とするようなハードウェアを設計することが困難であることも事実であった。しかも応用分野を広げるにつれ、ますます多機能・高機能化が要求され、同時に小型化が強く要求されるようになり、この期待に答えていくためには、自らが画像処理用LSIを設計し実現していくことが不可欠と考えるに至った。

このような状況を背景として、筆者らはイメージプロセッサ(以下IPと略すことがある)の研究を取上げることにした。本論文のこれまでの章で述べてきたように、パソコンやワークステーションの処理能力の不足は、どうしても専用のハードウェアによってカバーする必要があるということである。本章ではこのような問題に対処するために筆者らが進めてきたイメージプロセッサの研究について述べる。

画像処理にはいろいろなレベルがあるが、2値画像の単純な拡大縮小あるいはデータ圧縮などについては、既に専用プロセッサが、国内外でいくつか開発されていた。本研究で対象としたのは、そのように機能を専用化したものではなく、もっと汎用的に利用できるプロセッサである。例えば、FA応用については、従来はコスト的にほとんど不可能であった、多値画像のプログラマブルなリアルタイム処理である。そしてOA応用については、例えば第2章と第3章で述べた図形や映像の対話的

編集がターゲットであり、また、文書中に含まれる線画・画像・写真等を分離したり、それぞれをその特性に合わせた方式でデータ圧縮したり、よりハンドリングし易いデータ構造に変換したり、これらに総合的な処理を加えるといった高度で複合的な処理を実行できるプロセッサである。それ故、機能の汎用性と共に十分な高速性を持たせることが課題となる。

本章では、小型で超高性能の汎用イメージプロセッサに関し、その基本的なハードウェア構成を提案し、その試作結果について述べる。

4.1.2 汎用イメージプロセッサの位置付け

画像処理では、1GOPS(Giga Operation Per Second)というような単一のプロセッサでは実現できないほど、大きな計算能力が要求されるため、マルチプロセッサ形式のイメージプロセッサの研究が盛んである[4-1~6]。画像の処理を大きく3段階に分けると、次のようになる。

(1) 前処理ノイズ除去や画像強調を目的とする。

フィルタリングのような単純な繰返し演算が主体だが、生の画像を処理するため、膨大な演算を必要とする。

(2) 特徴抽出

輪郭線の追跡や特徴的パターンの抽出等を目的とする。

条件分岐が多発するやや複雑な演算が必要であるが、アルゴリズムを適切に選べば、前処理ほどには演算量は多くない。

(3) 構造解析

抽出された線分や特徴パターン相互の関係に基づいて、画像の構造を解析する。

照合やソーティングを主体とした演算が必要で、データ量は前2者に比べ減少しているが、組合せ数が膨大になることを回避するために、高度なアルゴリズムが要求される。

従来のマルチプロセッサでは、単純だが超高速演算が要求される前処理のレベルに着目していた。その場合には、画素数だけのプロセッサを用意し、それぞれのプロセッサは1ビットの演算という低い機能しか持たない[4-1,4](完全並列型と呼ばれる)、あるいはデータフロー型の制御構造を用いる[4-6]というような思い切った構成が採れた。そして、上位の特徴抽出や構造解析に関しては(高度な演算命令を使える代わりに)速度を犠牲にして汎用マイコンのプログラムにまかせるか、用途を限定して専用化した特殊なハードウェアを個別に設計して用いるという方法が採られていた。

ところが、イメージ・プロセッサの応用分野が拡大するにつれ、取り扱う画素数(絵の大きさ×分解能)がプロセッサ数で固定される方式では不満があり、また、画素当たりのデータ量も白黒2値だけでなく、濃淡8~16ビットまで可能としたいという要求がでてきた。さらに、上記各レベルで要求される処理も、より高度化してきている。すなわち、前処理のみの高速化では意味がなく、より上位のレベルを扱うことのできるプロセッサが要求される。そこで筆者等は、今回開発する汎用イメージ・プロセッサ(GPIP)のターゲットとして、前処理から特徴抽出までをカバーし、構造解析の一部までをサポートするマルチプロセッサシステムを目指すことにした(構造解析処理の高度化については、知識処理マシンとして開発されているプロセッサの応用も含め、次のステップとして検討する)。

GPIPをこのように位置付けた場合、プロセッサは汎用マイコン並の豊富な命令セットを持ち、取り扱うデータ幅も16ビット程度と広くなければならない。そして、その上で画像処理を効率良く実行できるようなバス構造と、大容量のオンチップメモリ、専用の命令セットを合わせ持ち、かつ汎用マイコンよりもはるかに速いマシンサイクルと、演算の並列実行による高速化を図る必要がある。このような観点から、筆者等は、画像処理用に拡張した信号処理プロセッサLSI(DSP-i)を開発した。本LSIの開発により、8プロセッサ程度をその周辺も含めてまとめて1ボードに搭載可能となり(要求性能に応じてプロセッサ数を増減すればよい。もちろん1プロセッサ、すなわち1チップだけを使ってもよい)、OAシステムの中などでも使えるような、経済的で実用的な超高性能イメージプロセッサが実現できるようになった。

一方、マルチプロセッサ形式で演算能力を高める場合、プロセッサ間通信あるいはプロセッサ間でのデータの共有や交換がオーバーヘッドとなって、期待したほどの能力が得られないという問題がある[4-7]。この一般的な問題に加えて、画像処理では2次元的に隣接するプロセッサと高速にデータ転送を行えるようにする必要がある。このため、前述の完全並列型のマルチプロセッサシステムでは、すべてのプロセッサが隣接8方向のプロセッサとデータ転送できるように接続されている。但し、この場合には演算機能自身が1ビットであることから、データ転送も1ビット毎となっている。GPIPの場合にも、2次元的な高速データ転送手段が必要である。しかし、16ビットデータを8方向に転送できるようなバスを、全プロセッサ間に張り巡らせることは、実装上の問題からとても不可能であり、新しいアーキテクチャが必要である。又、画像データおよび処理プログラムをどのように分割して複数のプロセッサに分担させるかが、マルチプロセッサシステム固有のオーバーヘッドの問題を解決する大きな鍵となる。本報で述べるGPIPのアーキテクチャは、上記各種の問題に対処すべく開発したものである[4-8]。

4.2 GPIPの構成

4.2.1 GPIPのプロセッサユニットDSP-i

ここではマルチプロセッサ構成をとるGPIPのプロセッサユニットとして使用することを前提に開発したDSP-iについて述べる。DSP-iは画像処理用にエンハンスされた16ビットデジタルシグナルプロセッサである[4-9]。そのブロックダイアグラムを図4-1に示す。1.3 μ m CMOSテクノロジーにより、50nsのサイクルタイムを達成しており、その規模は430Kトランジスタ、チップサイズは11.5 \times 12.9である。

DSP-iはマイクロ命令とピコ命令の2階層の命令制御方式を採用しており、前者に内部データの処理、後者に外部データの入出力を分担させることができる。通常、外部データの入出力は内部データの処理に比較して低速であるが、マイクロ命令から起動されるピコ命令が、それを独立並行に処理する。こうしてDSP-iのオンチップメモリに取り込まれた後は、このデータは50ns毎に、レジスタファイル的に自由にアクセスできる。特に画像処理では、フィルタリングなどに代表されるように、局

所的なデータへの繰返しアクセスが行なわれるので、オンチップメモリをキャッシュ的に利用できる本方式は全体のスループットを大幅に改善することができる。

マイクロ命令では、2つのデータ演算(図4-1中のALU又はBPUとMULT)、3つのデータバス制御(X,Y,D)、3つのアドレス演算(AUA,AUB,AUE)の合計8つのオペレーションが並列に制御される。ピコ命令が起動されると、AUEとデータバス(ED)およびその周辺が、マイクロ命令の制御から離れ、ピコ命令によって独立に制御される。

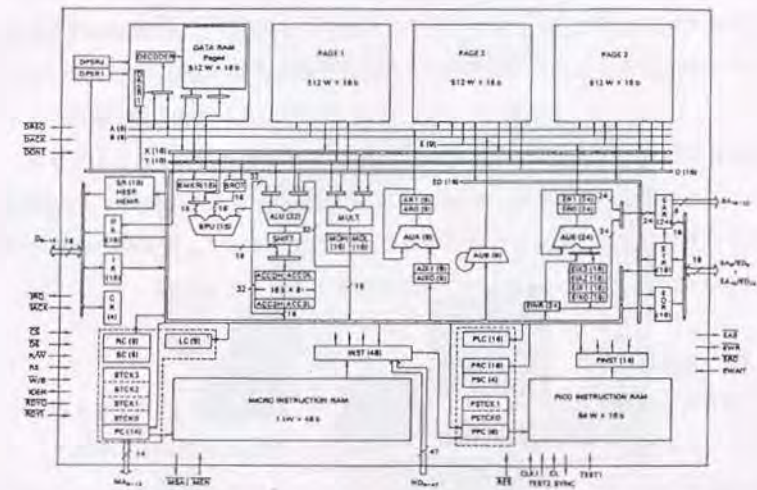


図4-1 DSP-iのブロック図

ALUは8つの16ビットアキュムレータ(4本の32ビットアキュムレータとしても使用可)を持つ演算器で、MULTは2つの16ビットデータから32ビットの結果を得る掛算器である。MULTとALUはパイプライン接続されているので、50ns毎に積和演算(40 MOPS)が可能である。BPUはワード単位でのビット処理を実行する特殊演算器であり、例えば16画素分の2値画像のパターンマッチングが100nsで実行できるなど、2値画像処理性能を大きく向上するものである。

アドレス発生は、3つの演算器で同時に行なわれるが、AUAとAUBはチップ上にあるDATA RAM(512ワード×4ページ)用であり、AUEは外部データメモリ(最大16Mワード)用である。AUAとAUEには、2次元アドレスを効率良く発生できるように、モディファイレジスタを持たせている。

DSP-iでは、チップ上の4つのデータバス、3つのアドレスバス、インストラクションバスは、それぞれ完全に分離されているので、1マシンサイクル(50ns)毎に、内部メモリの2ページの読出しと、1ページへの結果の書込み、2つのデータ演算を全て並列に実行することができる。このため、従来の汎用マイクロプロセッサのアーキテクチャに比べて、同じマシンサイクル速度でも10倍以上もの画像処理性能を得ることができる。又、マルチプロセッサ構成での効率良い制御ができるホスト(マイコン)インタフェースを持っている。その一つがプロセッサ番号識別機能であり、DSP-iには、それぞれ個別の2次元のプロセッサ番号を割り付けることができる。そして、ホストから指定する、RUN,HALT,WTREG(レジスタへの値のセット)などのコマンドに、このプロセッサ番号を含ませることによって、最大64×64個までのDSP-iを個別に、あるいはグループ単位で制御することができる。

4.2.2 基本構成

GPIPの全体構成を図4-2に示す。図中破線で示したプロセッサ・ユニット(PU)は、プロセッサ・エレメント(PE)と、そのローカルメモリ(Local Memory)から成り、これを64組並べる。尚、このPEとしてDSP-iを用いる。それぞれのPEとローカルメモリの間には、バス制御回路(Path Ctl.)が入っており、64個のバス制御回路はシフトレジスタ状に接続されている。又、このシフトレジスタはシフトリング(Shift Ring)を形成しており、シフトリング制御回路(Shift Ring Ctl.)によって制御される。そして、全体はマイクロコンピュータ(モトローラ社製MC 68000)で制御され、入出力のためのインタフェース部(DMA I/FとPIO I/F)を介して、ホストコンピュータと接続される。又、シフトリングは外部機器との間で画像の入出力を高速に行なうためにも利用する。

図4-3に試作したシステムの、図4-4に8PUを実装したPUボードの写真を示す。

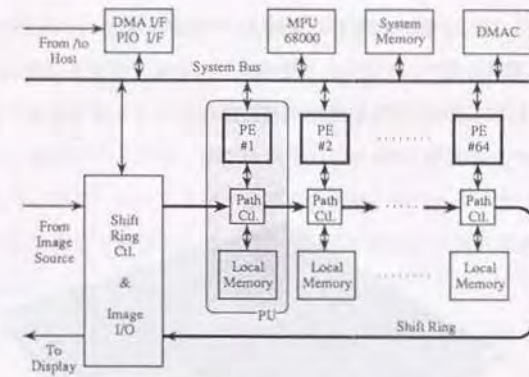


図4-2 GPIPのブロックダイアグラム



図4-3 試作したGPIPの外観

尚、今回の試作では、PU数を64(ボード8枚)としたが、本アーキテクチャによれば、PU数は任意に選択可能で、必要とする処理能力に合わせて自由に増減できる。又、ソフトウェアに関してもPU数を意識せずにプログラミングできるようなオペレーティングシステムを実現している。

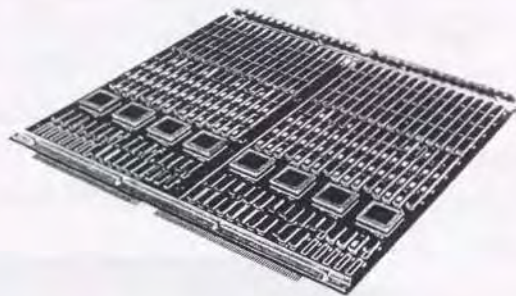


図4-4 PUボードの外観

GPIPでは、PUのハードウェア的な接続関係を、(画像処理にとっては有利ではないと考えられる)1次元的なシフトリング接続としている。敢えてこうした理由は次のとおりである。まず候補として考えた他の接続方法との比較を表4-1に示す。多次元のネットワークの接続は、PU間に交換機を置くような方式であるため、PU間接続の自由度は高いが、ネットワーク自身のハードウェア量が膨大となるので、小型の装置に纏めることができない。2次元の8方向接続は、完全並列プロセッサで採用されており、画像処理には適しているように見えるが、PUの2次元配列を(例えば縦8×横8というように)固定的にしてしまうため、取り扱う画像サイズの変更などが困難で、自由度は高くない。又、転送データ幅を大きく取るとハードウェア量が大きく、適当ではない。

これらに対し、シフトリング接続では、構造が単純であるために、ハードウェア量に関しては、小型化に適しており、転送データ幅を1ワード(16ビット)と広く採れる。又、物理的接続は1次元であっても、論理的には(8×8、4×16、2×32というように)任意の2次元配列として扱うことができるため、(8方向接続に比べ)かえって自

由度は高い。問題は、論理的な2次元配列の隣接関係にあるPUにデータを送る場合、1次元的にPU間を次々とシフトして実行することとなり、多数回のシフトを必要とすることである。しかし、この問題に関しては、後述するように、実際の利用形態に基づいて解析したところ、解決法が見出せたので、シフトリング接続を採用することにした。なお、シフトリングをプロセッサとローカルメモリの接続点の位置に設けたのは、次節で述べる多様な動作モードを実現するのに最適な場所であったためである。

表4-1 PUの接続方法の比較

	転送データ幅	速度	ハードウェア量	自由度	総合
ネットワーク接続 (多次元)	ビット	△	△	○	△
	ワード	◎	×		×
8方向接続 (2次元)	ビット	△	△	×	×
	ワード	◎	×		×
リングバス接続 (1次元)	ビット	×	◎	○	×
	ワード	○	○		○

4.2.3 GPIPの動作モード

GPIPでは、図4-2に示したように、処理対象である画像データはシフトリングを経由して、各PUのローカルメモリに送り込まれるようにしている。又、各PEは独立に専用のプログラムメモリを持ち、その内容(プログラム)はシステムメモリ(System Memory)から、必要に応じてDMAコントローラ(DMAC)によって転送される。このとき全PEに対し、一斉に同一プログラムを転送することもできる。そしてPEの任意アドレスからの起動や停止などの制御は、マイクロプロセッサ(68000)から行い、各PEは必要な時にはマイクロプロセッサに対して割込みを掛けることができる。

ところでGPIPでは、複数のPUによって画像を処理する訳であるから、画像データおよび処理プログラムを何らかの形で各PUに分配して、効率良く並列処理することが重要である。汎用性のあるイメージプロセッサとして、多種多様な画像処理ア

ルゴリズムを支障なく実行できるように、GPIPでは4つの動作モード(利用形態)を可能とした。

これらを表4-2に纏めた。以下それぞれのモードについて説明する。

(a) 複数データモード

処理すべき画像が大量にあって、これらに対し順次同一処理を施すというように、複数の画像をそれぞれ複数のPUに分配できるとき、このモードを用いると、PU間でのデータのやり取りが不要となるため、オーバーヘッドがほとんどない。但し、複数のPUの処理の終了時刻が一致したときには、画像データをローカルメモリに供給するシフトリングの競合が起き、待ちが生ずるので注意を要する(実際にはPUの終了時刻は、適度にランダム化されるので、大きなオーバーヘッドにはならないと考えられる)。

(b) 領域分割モード

本モードは、1枚の画像を高速に処理したいときに用いる。但し、画像を部分領域に分割し、これらを各PUに分配して処理する(後述の4.4節参照)ので、一連の処理が終了した時点で、部分領域の境界線上で、処理結果をつなぎ合わせる処理が必要となる場合がある。これは領域分割モードに固有の余分な処理であり、時に大きなオーバーヘッドになる恐れがあるが、後で詳述するように、処理方式の工夫により小さく抑えられるようにしている。

(c) 機能分割モード

本モードは、1枚の対象画像に対し、多数の特徴量を求めたり、多数の辞書とのマッチングを行なうような場合で、処理機能を複数PUに分担させられる処理アルゴリズムに適する。この場合、処理の結果として得られるデータ量は、元の画像のデータ量に比べはるかに少量になるので、各PUに分散する処理結果を集積する処理は、オーバーヘッドと呼ぶ程にはならない。

(d) バイブラインモード

本モードは、スキャナから得られる画像信号のように、発生する画素データの速度が遅く、このデータ・レートに合わせて(適当なバッファリングを行ないつつ)実時間で処理できるような場合であって、かつ処理内容がシーケンシャ

ルな複数の機能に分けられる場合に適する。DSP-iでは、1つのPUが処理結果を次々と隣のPUに受け渡して行く動作を、データ演算との並列処理で実行できるような機能を持たせているので、データ転送によるオーバーヘッドは生じない。

表4-2 GPIPの動作モード

モード	処理の概要	リングバス・サポート
複数データ	複数台のプロセッサに異なる入力画像を割り付けて、個別に処理する	バイパス機能
領域分割	入力画像を複数個に分割して、複数台のプロセッサで処理する	境界領域データの8方向高速転送
機能分割	複数台のプロセッサに異なる機能を割り付けて、同一画像を処理する	同一データのブロードキャスト
バイブライン	複数台のプロセッサに異なる機能を割り付けて、処理結果を順次受け渡す	隣のPEおよびイメージメモリとの接続

4.2.4 シフトリングの制御

図4-2で示したシフトリング(Shift Ring)のバス制御部(Path Ctl.)の1PU分の詳細を図4-5に示す。バス制御部には2種のラッチ(Data LatchとFlag Latch)があって、それぞれシフトレジスタ状に隣のPUの同様のラッチに接続され、全体としてシフトリングを構成している。

図4-5に基づき、まずデータの流れについて説明する。隣の段のデータラッチから送られて来る16ビットのデータは、データバス制御部(Data Bus Ctl.)に入る。ここには、PEのデータバスとメモリのデータバス(それぞれ16ビットの幅を持つ)が接続されている。そしてその先にその段に所属するデータラッチがあって、隣のPUへと出て行く。データバス制御部はセレクト信号(Select Signal)によって切替られる。セレクト信号とは、PEから出力されるリード信号(RD)、ライト信号(WT)、およびアドレスバス(図では省略している)の組合せによって決まる信号であって、これらによって図4-6に示す7種類の接続状態が選択される。前節で述べたGPIPの4つの動作モードに対応して必要となる全ての画像データ転送は、これら7種の接続状態を使い分けることによって実現できる。又、シフトリングはデータの転送のみの役割と

し、ローカルメモリへのアドレス及びリード・ライト信号は(プログラムによって)PEから供給するようにして、アドレスカウンタなどのハードウェアを省略し、回路規模の縮小を図っている。

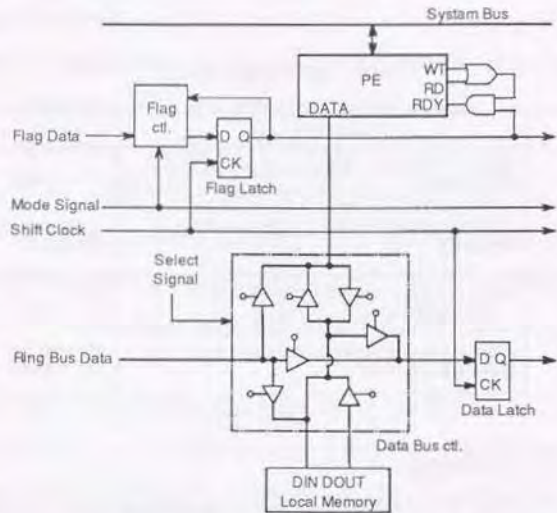


図4-5 バス制御部のブロックダイアグラム

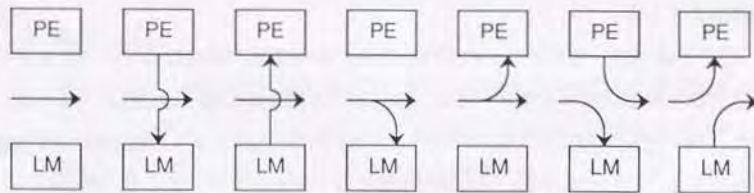


図4-6 バス制御部の接続状態

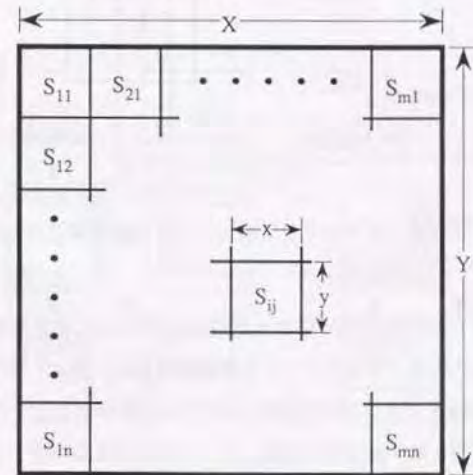
次にフラッグラッチの機能について説明する。図4-5に示したように、フラッグラッチの出力はPEのリード・ライト信号(RD, WT)とアンドを取って、PEのレディピン(RDY)に接続されている。従って、PEのリード・ライト動作は、フラッグラッチに1がセットされている時のみ実行され、フラッグラッチが1になるまでは、PEの

リード・ライト動作は引き延ばされる。つまり、シフトリング上を流れてくるデータが、所望のPUのデータバス制御部に到達するタイミングで、フラッグラッチがセットされるようにして、PEのプログラムによるリード・ライト動作との同期を取っている。尚、フラッグラッチの状態は、シフトリング制御回路(Shift Ring Ctl.)から送られて来るモード信号(Mode)、前段のフラッグラッチの状態、及び前シフトサイクルにおける自分自身の状態によって、フラッグ制御回路(Flag Ctl.)が決定する。

4.3 領域分割処理

4.3.1 領域分割モードに関する考察

領域分割モードにおいては、このモード固有の問題が生じるので、それをここで詳しく説明する。本モードでは、図4-7に示すように $X \times Y$ 画素の大きさの画像を $x \times y$ 画素の大きさの部分画像 S_{ij} に分割する。そして、この $m \times n$ 個の部分画像 S_{ij} を、 $m \times n$ 台のPUに割り当てて処理する。従って、実行すべき処理が濃度変換のように、画素単位のものである場合には、1台のPUで処理する場合に比べ、 $m \times n$ 倍の速度を得ることができる。



S_{ij} : 部分画像($1 \leq i \leq m, 1 \leq j \leq n$)

図4-7 画像の領域分割(2次元分割の場合)

ところが、フィルタリング処理のように、1つの画素の処理結果は、その画素を中心とするマスクエリアのなかに含まれる複数の画素のデータによって決定される場合には、状況はやや複雑になる。つまり、図4-8に示すように、画像の周辺部の画素においては、その画素を中心とするマスクエリアが、画像の外にはみ出すため、正しく処理することができない(このようにしてできる正しく処理できない部分を非正常部と呼ぶことにする)。この問題は画像処理一般に生じるが、特に領域分割モードにおける処理の場合には、後で各部分画像をつなぎ合わせて元の大きさに戻してやる必要があるため、部分画像の周辺(すなわち境界部)に非正常部が生じると致命的である。

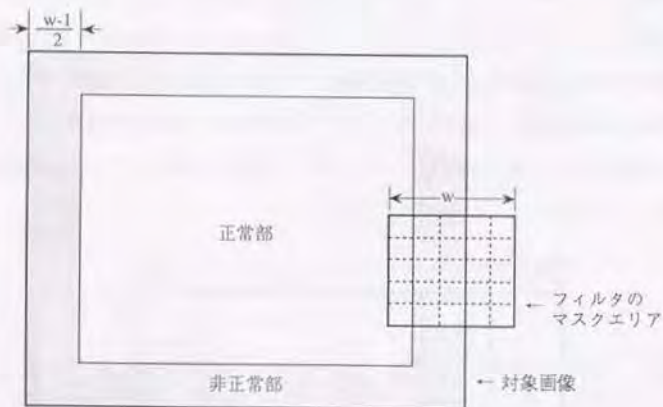


図4-8 フィルタリングによってできる非正常部

これを避けるために、領域分割モードでは、図4-9(a)に示すように、部分画像をあらかじめオーバーラップさせておくことが必要である。更に、部分画像の周辺に生ずる非正常部の幅(マスクエリアの幅から1を引いた数の半分)は、フィルタリングを繰り返すことによって、加算的に増加していくので、これを阻止する必要がある。つまり、フィルタリング処理を実行する度に(あるいは何回かの繰り返し毎に)非正常部を正常に処理された画像で置換することが必要となる。この非正常部の置換は次

のようにして実現できる。部分画像 S_{ij} をフィルタリング処理して生じた非正常部を、隣接する部分画像間で見ると、図4-10(a)に示すように、互いに相手の正常に処理されたエリア内にあることが分かる。従って、8方向に隣接するPUのもつこれらの正常な画像データを用いれば、先の置換は正しく行なうことができる。

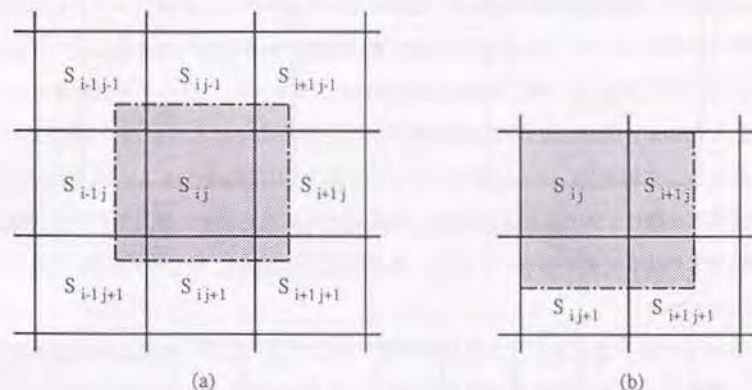


図4-9 オーバラップさせた部分画像 S_{ij} (一点鎖線で示す)

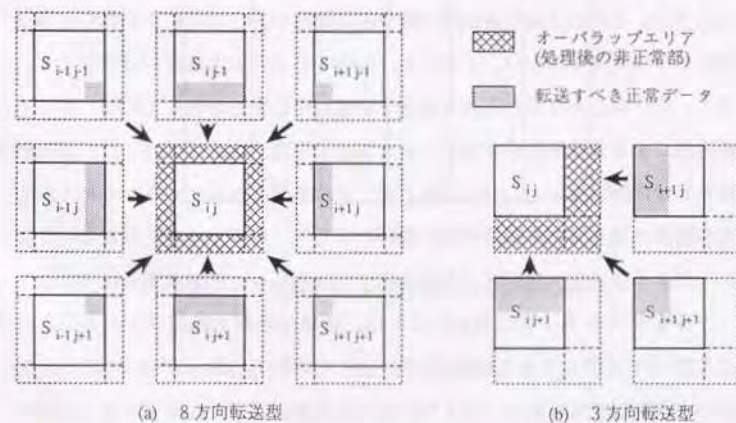


図4-10 S_{ij} から見たオーバーラップ・エリアと正常データの転送の関係

ところで、非正常部の置換に際し、8方向のPUからの画像データを転送せずに、これを3方向で済ませることが可能である。先の説明では、直感的に分かりやすい処理方式であることから、フィルタリング処理後の画像の位置が処理前と変わらないように、マスクエリアの中心位置に処理結果の画素を書き込む方式について述べた。しかし、結果の画像が処理前の位置に対して相対的に、全体にシフトすることを許すならば、マスクエリアの左上隅位置に処理結果の画素を書き込むようにしてもよい。こうすると、非正常部は図4-10(b)に示すように、右方と下方のみに生じる。それゆえ、前述の置換のためのデータ転送は8方向ではなく3方向のみでよいことになる。この場合、オーバーラップエリアの取り方は図4-9(b)のようにする。尚、非正常部の幅はマスクエリアの幅から1を引いたものとなるが、非正常部全体の画素数は前者の場合と変わらない。以後、前者を8方向転送型、後者を3方向転送型と呼ぶこととする。

ここまでは、フィルタリング処理の例について述べたが、輪郭線の追跡のように、画像データから折線(及び曲線)近似した点の列を得るような処理の場合には、少し異なる考察が必要である。領域分割モードで輪郭線追跡処理を行なう場合、隣接する部分画像間にまたがる線が存在すると、図4-11に示すように、これらは2分されてしまう。それ故、部分画像毎に輪郭線の追跡を終えた時に、それらを再びつなぎあわせなければならない。すなわち、各PUで求められた線分の端点同士を突き合わせて、同一点とみなされる点を統合する処理が必要である。この時、もしもこの折線近似による追跡処理が非常に小さな誤差で可能であって、点と点の対応がうまく取れば問題はない。しかし実際には、追跡処理の誤差が大きかったり、ちょうど部分画像の境界線上を線分が通っていたりして、対応がうまく取れないことがある。このような場合、対応付けを容易にするためにも、部分画像にはやはりオーバーラップが必要である。更に言えることは、領域の分割は単純な方がよい。つまり、先に図4-5で説明したような2次元的な分割ではなく、図4-12のように1次元的に(短冊状に)分割した方がよい。こうすれば、点と点に対応付けるアルゴリズムは、場合分けの数が圧倒的に減少し、単純かつ高速なものとなる。

但し、前述のフィルタリング処理の場合には、今述べた輪郭線の追跡処理のよう

に、境界線上での複雑な処理アルゴリズムを使わないので、敢えて1次元の領域分割を用いる必要はない。逆に後で述べるように、2次元の領域分割を用いた方が、オーバーヘッドを小さく抑えられる。

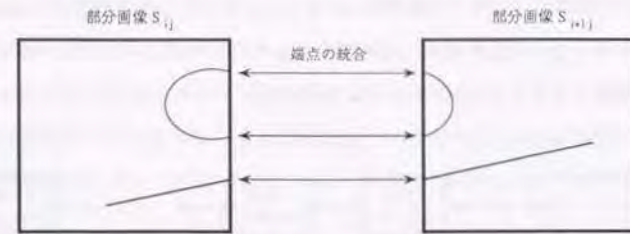
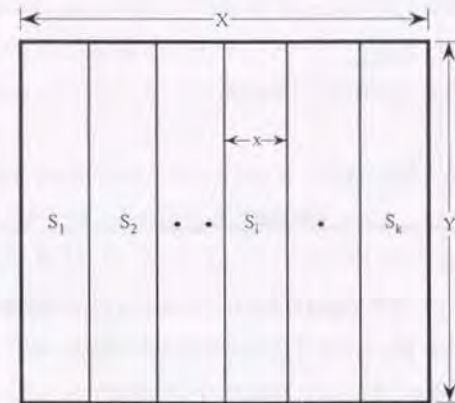


図4-11 部分画像間にまたがる線の例



S_i : 部分画像($1 \leq i \leq k$)

図4-12 画像の領域分割(1次元分割の場合)

4.3.2 領域分割モードにおけるシフトリングの制御

前述したように、GPIPでは複数のPUがシフトリングによって物理的に1次元的に接続されている。これを図4-7に示したような2次元の領域分割モードで利用する場合に、各PUを論理的な2次元配置($m \times n$)に対応させる。このとき、PU間の接続の関

係は図4-13のようになっており、実は画像のラスタスキャンとの整合性がうまく取れている。更に、PU間の論理的な2次元方向の距離と物理的な1次元の距離との対応も単純になる。例えば、論理的な2次元空間での右へのデータ移動は、シフトリングの1回のシフトであり、下への移動は m 回のシフト、左上への移動は $(mn-n-1)$ 回のシフトという具合に対応するが、この関係は全てのPUの組合せについて成立する。これらの特徴を生かしたシフトリングの制御方式について、以下に2つの例を示す。

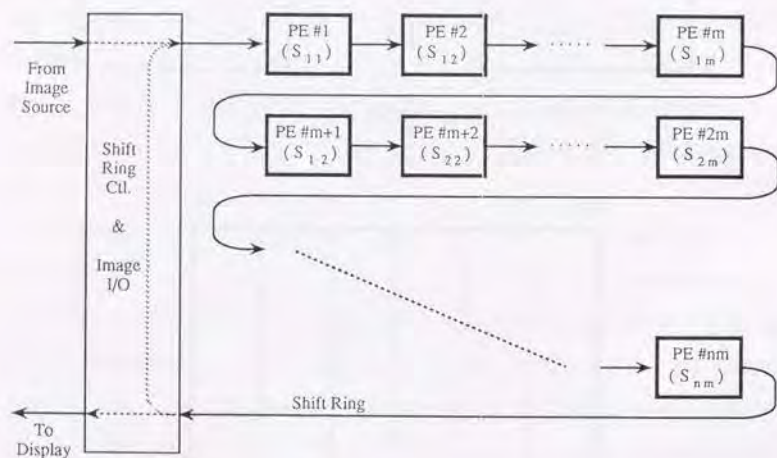


図4-13 2次元領域分割モードにおけるPUの接続関係

(a) 領域分割モードにおけるオーバーラップを持つ部分画像のロード

画像データは図4-2の画像入出力インタフェース(Image I/O)を介して、外部からラスタスキャンで送られて来るものとする。

- (1) 全てのPUのフラッグラッチをリセットする。
- (2) 全てのPEには、『ローカルメモリに対し、画像を書き込むべき先頭アドレスから順にインクリメントするアドレスを発生しつつ、シフトリングからのデータをローカルメモリに書き込むことを繰り返す』プログラムをロー

ドして起動しておく

- (3) 1番目のPUのフラッグラッチのみをセットし、画像データを Image I/O からシフトリングに流し込む
- (4) 次々とデータラッチ上をシフトして行く画像データは、1番目のPUのローカルメモリのみ書き込まれる。この間はモード信号は、ホールドモードにして置き、全てのフラッグラッチは変化させない
- (5) 2番目のPUが受け取るべき最初の画素データが2番目のPUに到達するタイミングで、モード信号をコピー1モードにして再びホールドモードにする

コピー1モードは、左側のフラッグラッチが1であるときのみ、そのフラッグラッチを1にするという意味である。これにより(1番目のフラッグラッチは1のまま)、2番目のフラッグラッチが1となる。この状態で画像データのシフトを続行すると、1番目と2番目のローカルメモリへの書き込みが並行して行なわれる。すなわち、オーバーラップ分の書き込みが同時に実行される。そして

- (6) その1ラスタ中で、1番目のローカルメモリに書き込むべきデータの最後の画素データが、1番目のPUを通過するタイミングで、モード信号をコピー0モードにする

コピー0モードは、左側のフラッグラッチが0であるときのみ、そのフラッグラッチを0にするという意味である。これにより、1番目のフラッグラッチは0となり、2番目のフラッグラッチは1を保つ。この状態で画像データのシフトを続行すると、2番目のローカルメモリのみに書き込みが行なわれる。以後、(5)(6)の「1番目」を「 $i-1$ 番目」、「2番目」を「 i 番目」と読み替えて

- (7) i を進めつつ、 i 番目と $i-1$ 番目のフラッグラッチを、上記と同様に(5)と(6)を繰り返して、 m 番目のPUまでの1ラスタ分の書き込みが行なわれたとき、再び最初の状態(3)に戻る

こうして、次々と後続のラスタの画像データの書込みが行なわれる。1次元の領域分割の場合には、このようにして全ての部分画像の分配が終了するが、2次元の領域分割の場合には、縦方向にも上記の横方向の制御と同じようなモード信号の制御を行なう。

(b) 領域分割モードにおける非正常部の置換

- (1) 全PEには、『ローカルメモリのアドレスを更新しつつリードを行ない、次いでライトを行なうことを繰り返す』プログラムをロードし起動して置く
- (2) モード信号によって一度全フラッグラッチを1にセットして、全PEにローカルメモリからの読出しを行なわせる
- (3) 再びモード信号によって全フラッグラッチを0に戻して(PEのライト動作を引き延ばして置き)、シフトリングを所定回数(PUの論理的な2次元配列と論理的な2次元の転送方向によって決まる数)だけシフトする
- (4) 再度全フラッグラッチを1にセットする

これで転送されたデータが、PEのライト動作によってローカルメモリに書き込まれる。

こうして1画素分の置換が完了する。以後

- (5) (2)に戻って、同じ動作を所定回数(オーバーラップの幅と部分画像の1辺の長さの積として求まる画素数、図4-11を参照)だけ繰り返す。

ここまでの、1方向分の画素データの置換が終了する。

- (6) シフトリングのシフト回数(3)で用いる)を変更して、(2)~(5)をもう一度実行する。

これで2方向分の置換が終了するが、3方向転送型で実際に必要な転送は全て

完了しており、3回目の転送は不要である。何故ならば、(左への転送の次に上への転送を行なうとして)1回目の置換時に図4-8で示した $S_{i,j+1}$ の正常データは、 S_{ij+1} へと転送されており、これを2回目の転送で S_{ij+1} 中の正常データと合わせて S_{ij} へ転送できるからである。

以上の説明で分かるように、領域分割モードにおける非正常部の置換は、シフトリングを用いることによって、複数のPUの全てに対して並列的に行なうことができるため、PUの数に依存せず短時間で終了する。

複数のPUへの同一画像の一斉ロードも、同様の制御方法によれば、画像データを一度シフトリングに載せるだけで全てのPUに配送されるため、極めて高速に実行することができる。

4.4 GPIPの性能の評価

ここでは、GPIPにおいて最もオーバーヘッドが大きくなる恐れのある領域分割モードについて、その性能を定式的に評価する。

4.4.1 領域のオーバーラップによる性能低下

画像を部分画像に分割するときを持たせるオーバーラップの幅を d とし、2次元の領域分割を考える(図4-7~図4-9を参照)。原画像のサイズを $X \times Y$ 画素、PUの数を $m \times n$ 個とすると、1PU当たりの処理画素数は $(\frac{X}{m} + 2d)(\frac{Y}{n} + 2d)$ であるから、1画素を処理するのに要する時間を t_p とすると、全画面の処理時間 T_{mp2} は

$$T_{mp2} = (\frac{X}{m} + 2d)(\frac{Y}{n} + 2d)t_p \quad (1)$$

となる。これに対し、1PUだけで全画面を処理する場合の処理時間 T_{sp} は

$$T_{sp} = XY t_p \quad (2)$$

である。この両者の比 α_2 をGPIPの計算能力であると考えると

$$\alpha_2 = \frac{T_{sp}}{T_{mp2}} = \frac{mnXY}{(X+2md)(Y+2nd)} \quad (3)$$

となる。

1次元の領域分割の場合には、同じ条件でmn個のPUを横に並べるとすると、1PU当りの処理画素数は

$$T_{mp1} = \left(\frac{X}{mn} + 2d\right) Y t_p \quad (4)$$

となる。このときのGPIPの計算能力 α_1 は

$$\alpha_1 = \frac{T_{sp}}{T_{mp1}} = \frac{mnX}{X+2mnd} \quad (5)$$

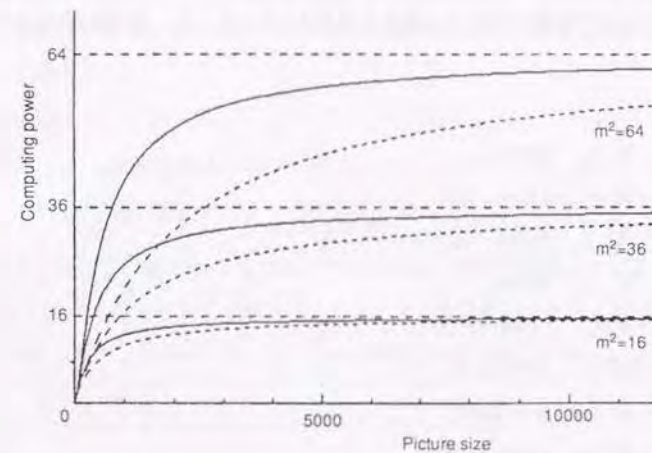
である。

原画像が正方形(X=Y)とし、PUも縦横同数配置する(m=n)として、いくつかのケースについて計算能力 α_1 と α_2 をプロットしたものを図4-14に示す。图中、横軸は画像サイズXであり、実線は α_2 を、破線は α_1 を示し、上から順にPU数 m^2 が64、36、16の場合について示している。尚、オーバーラップ幅dについては16とした。これは3×3画素の大きさのマスクで16回の繰返しフィルタリング処理、(あるいは32×32画素のテンプレートとのマッチング etc.)に対応できるオーバーラップ量であり、実用的な代表値として選定した。

図4-12から分かるように、原画像の大きさが十分に大きいときには、オーバーラップ部分の処理に要する時間は無視できるため、GPIPの計算能力はPUの台数分に等しくなる。又、2次元の領域分割の方が1次元の領域分割よりもはるかにPUの利用効率が高く、この傾向はPUの数が増加する程大きくなる。前者の特性は、オーバーラップ部の画素数が部分画像中に占める比率に起因するものである。後者の特性については、画像を升目に分割するか、短冊に分割するかの差異によるもので、短冊状の分割ではPU数に対する画像サイズが小さすぎると、部分画像が細長くなりすぎる結果として生じるものである。

具体的な性能の指標として、オーバーヘッドが10%となる画像サイズを調べてみる

と、表4-3のようになる。現実のA4サイズの文書処理(2次元領域分割モードを多用する)では、1辺の長さが4000画素程度、地図画像処理(線の追跡などが多く、1次元領域分割モードを使う)では10000画素程度の大きさの画像を扱う。従って、オーバーヘッドを10%以下とするには、前者では64PU、後者では36PU程度となる。このような点から、今回の試作では64個のPUを用意することとした。



実線: 2次元分割処理の能力 α_2
破線: 1次元分割処理の能力 α_1

図4-14 画像サイズに対するGPIPの計算能力

表4-3 領域分割モードにおいて10%のオーバーヘッドとなる画像サイズ

PU数	画像の1辺の長さ(画素)	
	1次元領域分割	2次元領域分割
16	4068	2366
36	10368	3549
64	18432	4733

4.4.2 非正常部の置換による性能低下

GPIPの領域分割モードにおいては、フィルタリング処理等の後に生じる非正常部を、隣接PUからの正しい画像データで置換することが必要であり、これをシフトリングのシフトによって実行する。このデータ転送が、画像処理の間に入ることによる性能低下を調べる。1つのPUに対し隣接するPUからのデータ転送を行なうのに要する1画素当りの時間は、ローカルメモリの読出し、書込み時間をそれぞれ t_r 、 t_w とし、シフトリングの1回のシフトに要する時間を t_s としたとき、論理的な2次元空間上の方向により

$$\begin{aligned} \text{右} &: t_r + t_s + t_w \\ \text{左} &: t_r + (nm - 1)t_s + t_w \\ \text{上} &: t_r + (nm - n)t_s + t_w \\ \text{下} &: t_r + nt_s + t_w \\ \text{右上} &: t_r + (nm - n + 1)t_s + t_w \\ \text{右下} &: t_r + (n + 1)t_s + t_w \\ \text{左下} &: t_r + (n - 1)t_s + t_w \\ \text{左上} &: t_r + (nm - n - 1)t_s + t_w \end{aligned}$$

となる。ここで n 、 m は先と同じようにPUの横と縦の数であり、その他についても先と同じノテーションを用いる。

これらと図4-7から分かるデータ転送量を用いて、非正常部の置換に要する時間を求めると、2次元の領域分割で8方向転送の場合

$$T_{ig} = 2d \left(\frac{X}{m} \right) (t_r + t_w + m^2 t_s) \quad (6)$$

となる。但し、式を簡単にするため $n=m$ 、 $Y=X$ とした。これに対し、3方向転送の場合の所要時間は

$$T_{ig} = d \left(\frac{X}{m} + 2d \right) \{ t_r + t_w + (m + 1)t_s \} \quad (7)$$

となる。シフトリングのシフト時間 t_s の係数が8方向転送では m^2 となっている。すなわち8方向転送では、全PU数 m^2 に比例する回数だけのシフトリングのシフトが必要なのに対し、3方向転送では、その平方根+1の回数だけのシフトで済むことが分かる。両者の処理結果には差がないので、GPIPでは3方向転送を採用する。

次に非正常部の置換のための転送時間の処理時間に対する比を求める。処理時間は式(1)及び式(2)として求められている。これらの式において、1画素を処理する時間 t_p は、ローカルメモリのリード・ライトに要する時間と、演算に要する時間 t_{pp} の和であり

$$t_p = k_r t_r + k_w t_w + t_{pp} \quad (8)$$

と表される。ここで k_r 、 k_w は1画素を求めるために必要なメモリのリード・ライト回数である。GPIPのPEとして用いるDSP-iでは、データ演算とローカルメモリへのリード・ライトは並行して実行できる上、データ演算の方がはるかに速いので、 t_{pp} は0とみなすことができる。又、 $t_r = t_w$ が成立するから、これをあらためて t_a と書き、 $K = k_r + k_w$ として

$$t_p = K t_a \quad (9)$$

と書ける。これと $Y=X$ 、 $n=m$ の条件を用いると式(1)は

$$T_{mp2} = \left(\frac{X}{m} + 2d \right)^2 K t_a \quad (10)$$

一方式(7)は同じ条件で

$$T_{ig} = d \left(\frac{X}{m} + 2d \right) \{ 2t_a + (m + 1)t_s \} \quad (11)$$

となり、式(10)と式(11)の比をデータ転送による性能低下 β と定義すると

$$\beta = \frac{md \{ 2t_a + (m + 1)t_s \}}{K (X + 2dm) t_a} \quad (12)$$

となる。具体例として

PU数: $m^2 = 64$
 オーバラップ量: $d = 16$
 メモリアクセス回数: $K = 16$
 シフトリングの速度: $t_f = 0.05(\mu s)$
 ローカルメモリのアクセス時間: $t_a = 0.2(\mu s)$

としたとき、画像サイズ(1辺の画素数) X に対して、性能低下 β がどう変化するかを図4-15に示す。画像サイズが500画素以上であれば、性能低下は5%以下であり、サイズが2000画素ではわずか1.5%となるので、実用上ほとんど問題とならないことが分かった。実際に1024×1024画素の画像を64PUで処理した場合の処理時間を表4-4に示すが、例えばフィルタサイズが3×3のフィルタリングの場合で2%、画像全体を90度回転するために、PU間で入れ替えるデータ量が最も多い、2次元のFFTの場合でも19%のオーバーヘッドに納まっている。また、コサイン変換や文字フォントの描画では、画像のオーバーラップは不用であるため全く問題は生じない。

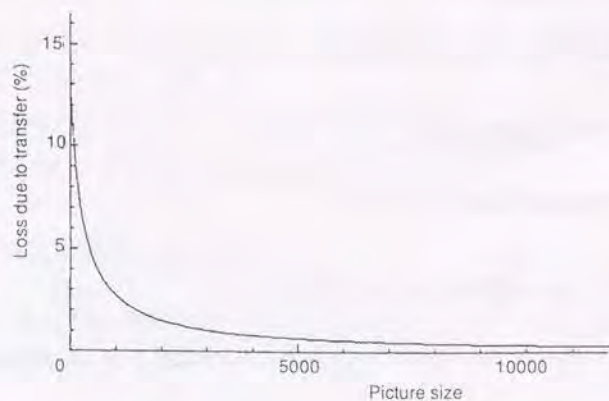


図4-15 画像サイズに対する、GPIPの非正常部の置換に伴う性能低下

表4-4 GPIPの処理時間の例

処理内容	対象画像	演算時間	データ転送	オーバーヘッド
3×3 フィルタリング	16ビット画像 1024×1024	14 ms	0.34 ms	2%
8×8 DCT (コサイン変換)		19.2 ms	0 ms	0%
2-D FFT		153 ms	29 ms	19%
16×16 マッチング	二値画像	70 ms	3.9 ms	5%
文字の輪郭線描画	1024×1024	1 ms / 文字	0 ms	0%

4.5 結果の検討

ハードウェア規模を十分小さく抑えながら、高効率でマルチプロセッサを動作させて、1GOPS(Giga Operation Per Second)をこえる超高速汎用イメージプロセッサを実現する方式の検討を行ない、シフトリングで結合されたマルチプロセッサ・アーキテクチャを開発した。このアーキテクチャでは、PU(プロセッサユニット)間が物理的に1次元接続であるが、これを論理的な2次元配列とみなして制御することによって、2次元の構造を持つ画像を効果的に処理することができる。又、シフトリングを有効に働かせて、画像処理で必要とする各種機能に柔軟に対処できるように、4つの動作モードを用意した。

上記のモードの内、特に領域分割モードにおける制御方式について詳しく検討した。その結果次のことが明らかとなった。

- (1) 原画像を部分領域に分割するとき、互いにオーバーラップを持たせることが必要な場合がある。
- (2) 画像処理の内容によっては、部分画像の周辺部に正しくないデータ(非正常部)が生ずることがある。
- (3) 非正常部は隣接PUからの正しいデータで置換することができる。
- (4) 隣接PUからのデータの置換は、8方向からと3方向からの2方式が考えられるが、前者はPU数(m^2)に比例する時間がかかるのに対し、後者はその平方根(m)に比例する時間で済む。

(5) シフトリングを複数のPUの全てから並列的に利用する制御方式を開発したので、(4)との相乗効果でデータ転送時間の大幅短縮が可能となった。

又、プロセッサ間のデータ交換等によるオーバーヘッドに起因する性能低下は、領域分割モードにおいて、最も大きいと考えられる。そこで、これらについて数式を導出し、具体的な数値データを用いて評価し次の結果を得た。

- (6) 2次元の領域分割の方が1次元の場合よりもはるかにPUの利用効率が高い。
- (7) 部分画像にオーバーラップを持たせることによる性能低下は10%程度である。
- (8) PU間のデータの置換のために費やす時間による性能低下は5%以下である。

今回開発したGPIPでは、64個の信号処理プロセッサ(DSP-i)をPEとして用いることによって、1GOPSの画像処理能力を達成することを目指した。DSP-iの演算能力は、20MOPS(Mega Operation Per Second)であるので、前述の各種要因によるマルチプロセッサのオーバーヘッドが合計で15%程度に抑えられると算定されたことから、GPIPは1GOPSの性能を達成できたと考えられる。

GPIPシステムでは、プロセッサエレメントの数を自由に増減することが可能で、より高性能が必要な場合には、最大で120 GOPS (3639 プロセッサ)の構成が可能である。一方、TV画像のパターンマッチングを数十ms程度で行うには、250 MOPS (8 プロセッサ)以下の能力でよいので、そのような用途の場合にはプリント基板1枚のサイズにまで小型化できる。このようなサイズ変更ハードウェア、ソフトウェア共に極めて簡単に対応できるように設計した。本システムはこの柔軟さに加え、処理の内容をマイクロプログラムにより自由に記述できるので、従来は、装置規模の制約あるいは機能面での不十分さから、画像処理を導入することができなかった分野にも適用を広げるものと期待される。もちろんマルチプロセッサ構成のGPIPで開発したソフトウェアを、そのまま1チップのDSP-iで利用することも可能である。従って、今後の個人用のマルチメディアパソコンに、さらにはインテリジェント・スキャナのような機器にも、GPIPで開発されたプログラムを載せたDSP-iが組み込ま

れ、その高機能化に貢献できると考える。

また、このGPIPのマルチプロセッサ方式はコンパクトであることから、複数のプロセッサとメモリをワンチップに搭載するLSIに発展する可能性がある。そうすれば今回の試作でも問題となったPEからローカルメモリへのアクセス速度の制限や、シフトリングのシフトクロック周期の制限、すなわちメモリ・アクセス・ネックによる性能低下の問題も解消されるので、ワークステーションやパソコンのアーキテクチャでは到底達成できないような超高性能を実現できると考えられる。

4.6 結言

OAを始めとする広い分野に適用可能で、要求性能に合わせてプロセッサユニット(PU)数を任意に増減可能な、マルチプロセッサ構成の超高速汎用イメージプロセッサ(GPIP)の方式の検討を行ない、次の結果を得た。

- (1) コンパクトなマルチプロセッサ・アーキテクチャを開発し、1ボードに8PUを搭載可能とした。
- (2) 物理的にはPUを1次元のシフトリング結合とすることによって回路規模を抑え、論理的な2次元の制御を行なうことによって自由度を大きく取れる。
- (3) PUのデータ交換の速度は、シフトリング上に並列的にデータを載せることによって大幅に向上できる。
- (4) 以上により、マルチプロセッサの並列動作のオーバーヘッドを15%程度に抑えることができる。
- (5) GPIPの性能は、そのPUに64個のDSP-iを用いて、2GOPS(Giga Operation Per Second)を達成できた。

参考文献

- [4-1] K.E.Batcher: "Design of Massively Parallel Processor", IEEE Trans. on Comp., pp. 836-840 (1980)
- [4-2] M.Onoe, 他: "Real time parallel computing -- Image Analysis", Plenum Press

(1981)

- [4-3] M.J.B.Duff: "Computing Structure for Image Processing", Academic Press (1983)
- [4-4] T.Sudo: "An LSI Adaptive Array Processor", ISSCC 82, pp. 122 (1987)
- [4-5] 佐々木, 他: "構造可変型ビデオレート画像処理システム「韋駄天」", コンピュータビジョン, 37-1 (昭60)
- [4-6] 森下, 他: "TIP-3システム構成", 信学全大 1428 (昭59)
- [4-7] P.Moller-Nielsen, 他: "Saturation in a Multiprocessor", IFIP Information Processing 83, pp. 383-388 (1983)
- [4-8] H.Ueda, K.Kato, H.Matsushima, K.Kaneko and M.Ejiri, "A Multiprocessor System Utilizing Enhanced DSP's for Image Processing," Proc. of Int'l Conf. Systolic Arrays, pp.611-620, 1988.
- [4-9] K.Kaneko, T.Nakagawa, A.Kiuchi, Y.Hagiwara, H.Ueda and H.Matsushima, "A 50ns DSP with Parallel Processing Architecture," Proc. of ISSCC, pp.158-159, 1987.

第5章 結論

第1章では本論文の序章として、本研究の背景およびその位置づけを明かにした。本研究は、対話的マルチメディアハンドリングにおいて、画像処理技術と認識技術を活用することによって、ユーザとシステムの対話を、より高度化することを目指し、開発してきたものである。第2章では図形のハンドリングを取り上げて、マルチメディアにおけるマンマシン対話における課題と、その解決の方向を明かにしようとした。ここでは会議用のスライド等の図をカラー画面上に創作するための、新しい対話型図形編集装置方式について提案した。本研究の動機の一つは、画素単位の処理に徹した図形編集方式の利点を追及することであった。従来のグラフィクスが、「線」を図形構成の基本要素としていたのに対し、ここでは、図形を「面」として定義し、それを操作する新しい図形編集方式を提案した。また、この図形編集方式を、マイクロコンピュータを主体とした卓上型装置として試作した結果について述べ、多様な図が効果的に構成できることを示した。

図の構成要素である図形を、座標とパラメータの入力に対し、論理値を出力する計算体形として統一的に定義する方式により、少数の基本図形から、その論理接続によって任意の複合図形が定義できるようになる。又、図をパラメータ群から成る図形コマンドの順序を持った集合として定義したので、これを逆方向に探索する処理によって、図形の削除や挿入などが自然な形で実行できた。それ故、人間にとって理解しやすく、またその操作方法も親しみ易い対話的図形ハンドリングが可能となった。

対話的図形ハンドリング実現上の問題点は、パラメータの数の多さであり、図形コマンドを誰もが簡単に入力できる方法を見出すことにあった。本装置では、動的仮想キー方式を採用し、更にスペースキーを押した場合には、自動的にデフォルト値が入る等の方式により、まず描いてみてその結果で、大きさや位置、色などを所望のものに変更するという作図法を開発した。これによって、いちいち操作法を覚えなくてよい上、大まかな図形の配置のあと、細かな修飾をするという利用方法が可能となり、極めて効率のよい図形編集ができることが確認された。

もう一つの問題点は、ユーザの操作性の向上や編集操作に対する画面変化の違和

感の削除であった。これについては本システムでは、図形を座標駆動とパラメータ駆動の両形式で操作できる構造と、図形コマンド列の逆方向探索とによって、画面上の図形の編集を、操作者の意図どおりに(不自然な画面を出さずに)進行させられるようになった。

これらによって、図形編集における対話性は大きく向上した。しかしながら、マイクロコンピュータによる処理では、おのずと速度性能に限界があり、通常の文字を主体とした簡単なスライド程度ならよいが、図形の数が増える複雑な説明用の画面の場合には快適性の面で問題があるということも分かった。

第3章では動画像を中心とするマルチメディアの自由な編集・作成を行うシステムについて述べた。ここでは、ユーザの創造的活動を支援する環境を実現するための新しいアプローチを提案し、プロトタイプを試作結果について報告した。このアプローチの特徴は画像処理・認識技術を応用して、動画像情報の時間・空間構造を一貫性のある形で視覚化して、それをダイレクトマニピュレーションできるようにすることにある。これによりユーザは対話的に映像素材を編集できる。具体的にはビデオ情報を解析して、その構造の自動記述を行なうための機能とこれらの機能によって得られるビデオの時間空間構造をビジュアライゼーションする新しい手法を提案した。

ビデオ情報の解析機能としては、カット分割、カメラと被写体の動きの解析、被写体の軌跡や輪郭線の抽出、特定の被写体の存在の判定及びその存在期間の記述等を開発した。そしてこれらの機能により動画アイコンを用いた動画像の構造の視覚化を実現し、その上でこれら動画アイコンのダイレクトマニピュレーションによる時間軸編集を実現した。また、画像中の対象物の切り出しについても実験的な機能を組み込み、動画像の合成など、ユーザにとってのメリットと今後の研究課題を明かにした。

更に、被写体の存在判定結果を用いて、同じ被写体はビデオ全体を通じて自動的にリンクされるようにした。このようにして得られるリンクをオブジェクトリンクと名付けた。オブジェクトリンクによるナビゲーションと画像解析結果によってサポートされるビジュアライゼーション手法により、ユーザが生のビデオデータのみ

ならず、ビデオ情報の構造、そしてこれに付加される記述内容を、ダイレクトマニピュレーション(直接操作)によって自由にブラウジングしたり、編集したりできることを示した。

本研究では実験結果を用いてこれらのことを示し、画像処理・認識技術を応用して、対話的なマルチメディアのハンドリングにかかわるユーザの創造的活動をサポートすることが非常に有効であることを明かにした。

従来のビデオ編集では、ビデオの情報構造が頭の中にはっきりと構築されるまで、ユーザは早送りと巻き戻しを繰り返す必要があった。そしてこの作業は非常に時間がかかるものであった。しかし、本プロトタイプシステムでは、画像認識技術を応用することによって、この時間のかかるステップが、ほとんど不要となった。そして動画アイコンのダイレクトマニピュレーションによるカットの取捨選択と並べ替えによる実質的な編集に、いきなり取り掛かれるようになった。またユーザは、これらのビジュアライゼーションとオブジェクトリンクによるナビゲーションの機能を使用して、直感的にシーンの構造とビデオのストーリーの流れを概括したり、詳細なカットの記述を自由にブラウジングしたりできるようになった。

但し、このプロトタイプでもワークステーションの性能不足が顕在化した。カットの自動分割や被写体の存在判定に関しては、アルゴリズムを工夫することにより、リアルタイム処理を可能としたが、動きの解析や対象物の切り出し等の処理を精度良く実行するためには、オフライン処理とせざるを得なかった。NTSCビデオ方式でも数十倍の高速化が、今後普及すると考えられるハイビジョンでは、200~300倍の高速化が必要である。

第4章では、これまでの章で明かとなったパソコンやワークステーションのマルチメディア処理能力の不足をカバーするためのイメージプロセッサの研究について述べた。このイメージプロセッサは前処理から構造解析までの広い画像処理に適用可能とすることを目指した超高性能の並列プロセッサである。ここでは信号処理用プロセッサを画像処理用に拡張して新たに筆者らが開発したLSIをプロセッサユニット(PU)として用いた、マルチプロセッサ構成の超高速汎用イメージプロセッサ(GPIP)の方式を提案し、その試作結果について述べた。

ここで提案した方式は、物理的にはPUを1次元のシフトリング結合(データをシフトクロックに同期してPU間で巡回させ、プロセッサもこのクロックに同期してデータを読み書きするバス構造)とすることによって回路規模を抑えつつ、論理的な2次元の制御を行なうものである。また、画像処理で必要とする各種機能に柔軟に対処できるように、4つのシフトリング動作モードを用意した。これにより、2次元の構造を持つ画像を効果的に処理することができる。また、このアーキテクチャはプリント基板1枚に8PUを搭載可能とするようなコンパクトなマルチプロセッサ・アーキテクチャである。

以上のような設計により、マルチプロセッサの並列動作のオーバヘッドを常に20%程度に抑えることができた。また、シフトリング上に並列的にデータを載せる高速多重転送方式によって、PUのデータ交換の速度を大幅に向上できた。同時にGPIPは画像のサイズ、データ形式(2値、多値、カラー)や処理内容などに制限がなく、PU数も自由に変えられるフレキシブルなシステムとなった。64個のDSP-iを用いて試作したGPIPの総合性能は、2GOPS(Giga Operation Per Second)を達成した。

GPIPシステムでは、プロセッサエレメントの数を自由に増減することが可能で、より高性能が必要な場合には、最大で120 GOPS (3639 プロセッサ)の構成が可能である。一方、TV画像のパターンマッチングを数十ms程度で行うには、250 MOPS (8 プロセッサ)以下の能力でよいので、そのような用途の場合にはプリント基板1枚のサイズにまで小型化できる。本システムはこの柔軟さに加え、処理の内容をマイクロプログラムにより自由に記述できるので、従来は、装置規模の制約あるいは機能面での不十分さから、画像処理を導入することができなかった分野にも適用を広げるものと期待される。

また、このGPIPのマルチプロセッサ方式はコンパクトであることから、複数のプロセッサとメモリをワンチップに搭載するLSIに発展する可能性がある。ワークステーションやパソコンのアーキテクチャでは到底達成できないような超小型超高性能のマルチメディアプロセッサを実現できると考えられる。

以上に述べたように、本研究では対話的マルチメディアハンドリングにおける画像認識技術、特に図形や映像を編集するシステムのソフトウェアとハードウェア技

術の開発を行った。本研究で開発したソフトウェア技術はプロトタイプによる実用性評価を終え、映像ハンドリングシステムとして(株)日立製作所において製品化を検討中である。またハードウェア技術の一部は画像処理向き信号処理プロセッサLSIとして製品に活かされている。今後はより高度な、すなわち知的協調といえるようなシステムとユーザの対話環境の実現に向けて、研究を進めていきたい。

謝辞

本論文は、昭和56年から平成5年にかけて(株)日立製作所中央研究所において行ったマルチメディアハンドリングのための画像認識技術に関する研究の総合報告である。この研究を遂行するにあたり、また、本論文をまとめるにあたり、多くの方々からご指導、ご支援を賜わった。ここではその内容を紹介することで謝意に変えさせて頂きたい。

本論文をまとめるにあたっては、画像処理の権威であられる東京大学生産技術研究所高木幹雄教授にご指導を賜わった。極めてご多忙中であるにもかかわらず、丁寧なご指導とご助言を幾度となく頂き、心から感謝する次第である。また、貴重なご助言を賜わった、東京大学工学部羽鳥光俊教授、原島 博教授、石塚 満教授、生産技術研究所坂内正夫教授にお礼を申し上げる。

本研究は(株)日立製作所中央研究所と、その米国分室ともいうべき存在であったHISL Inc. において行われたものである。特に、同研究所の江尻正員博士、烏野 武博士、松島 整氏には、著者の上司として、また共同研究者として、研究の目標設定や研究成果のまとめ方等に関して多くのご指導を頂いた。江尻正員博士と烏野 武博士は、実用的な画像認識技術の重要性を早くから提唱され、著者が入社して以来今日にいたるまで、高所からの研究のご指導を賜わった。企業における研究に対する両博士の考え方が著者の研究の指針となり、両博士の親身なご助言とご支援が研学生活の励みとなった。

江尻正員博士には、特に第2章で述べた対話型図形編集システムの研究において、細部にわたるご指導を賜わった。研究に対する合理的なアプローチ、企業における研究でやや失われがちな理論的検討を行う態度等を博士から学ばせて頂いた。烏野 武博士と松島 整氏には、実時間画像処理アルゴリズムとそのハードウェア化技術に関してご教示を頂いた。第4章で述べた画像処理用高速プロセッサの研究は、このお二人のご指導に負うところが大きい。

(株)日立製作所中央研究所に筆者が入社した昭和48年からの歴代第6部長の沼倉俊郎博士、川崎 淳博士、猪瀬文之博士、大西淑弘氏、千葉常世博士、堤 善治氏、角田義人博士、長島重雄博士、そして HISL Inc. の山田弥彦社長、また知能システム

部部長の杉江 衛博士、研究室のリーダーであった加茂善久氏、大島義光氏には、本研究の実施や論文執筆の便宜を与えていただくと同時に、暖かいご支援を賜わった。

さらに、本研究に関する試作、評価、検討では多くの方々にご協力を頂いた。

第2章で述べた対話型図形編集システムの開発では、装置の試作において、米国 Dirersity West 社の John F. Binns 氏と Charlene Ament 女史にご協力頂いた。

第3章で述べた対話型映像編集システムの開発では、システムの試作と評価において、共同研究者である宮武孝文氏、吉沢 聡氏、炭野重雄氏、長坂晃朗氏、谷口勝美氏らのご協力を頂いた。またマサチューセッツ工科大学メディアラボの Glorianna Davenport 助教授からはシステムコンセプト等について貴重なご討論を頂いた。

第4章の画像処理用高速プロセッサの開発では、LSI と装置の試作において、共同研究者である金子憲二氏、加藤寛次氏らのご協力を頂いた。

恩師である大阪大学工学部の滑川敏彦教授（当時）、森永規彦教授、現在京都工芸繊維大学の笠原正雄教授には、研究生活の端緒を開いて頂くとともにその後の研究に必要な情報工学の基盤を与えて頂いた。本論文の研究を遂行するにあたり、ここでご紹介した方々以外の方々からもご指導とご支援を頂いた。これらの方々も含め、改めて深謝の意を表す。

平成6年早春

上田博唯

著者発表論文目録

1. 主執筆論文

1.1 国内一般論文

- [1] 上田博唯, 藤方健二, 江尻正貞, "マイクロコンピュータによる素片波形接続型音声合成方式とその応用," 信学論(D), Vol. J63-D, No. 11, pp931-938 (1980).
- [2] 上田博唯, 江尻正貞, "画素単位の処理アルゴリズムに基づく対話型図形編集方式," 信学論(D), Vol. J65-D, No. 6, pp742-749 (1982).
- [3] 上田博唯, 宮武孝文, 吉澤 聡, "認識技術を応用した対話型映像編集方式の提案," 信学論(D), Vol. J75-D-II, No. 2, pp. 216-225 (1992).
- [4] 上田博唯, 宮武孝文, 炭野重雄, 長坂晃朗, "動画像解析に基づくビデオ構造の視覚化とその応用," 信学論(D), Vol. J76-D-II, No. 8, pp. 1572-1580 (1993).

1.2 国内解説論文

- [1] 上田博唯, "インテリジェント映像編集技術," テレビ誌, Vol. 47, No. 1, pp. 24-29 (1993)
- [2] 上田博唯, "FRIEND21 プロジェクト," 情処誌, Vol. 34, No. 7, pp. 892-901 (1993)
- [3] 上田博唯, "マルチメディアのハンドリング," テレビ誌, Vol. 47, No. 11, pp. 5-8 (1993)

1.3 国内研究会講演論文

- [1] 上田博唯, "インタラクティブな動画像編集方式の提案," 信学技報, IE90-6, pp. 39-46 (1990).
- [2] 上田博唯, "Laboratory Review—FRIEND 21 研究センター," 計測自動制御学会, 第6回ヒューマン・インタフェース・シンポジウム, pp. 37-38 (1990).
- [3] 上田博唯, 比田井 裕, 宇山政志, "ユーザ情報を積極的に蓄積・利用するHIアーキテクチャー—エージェンシーモデル—," 信学技報, HC91-, pp. 23-30 (1991).
- [4] 上田博唯, 比田井 裕, 宇山政志, "エージェンシーモデル—ダイナミックに外見群とマッピングするアーキテクチャー," 計測自動制御学会, 第7回ヒューマン・インタフェース・シンポジウム, pp. 531-538 (1991).

- [5] 上田博唯, "画像認識技術を応用した対話型動画編集方式," 信学会, ハイパーメディアと信号処理箱根フォーラム'91, pp.217-225 (1991).
- [6] 上田博唯, "ハイパーメディア/知的オーサリングシステムの実現へ向けて," 信学会, 第4回情報伝送と信号処理ワークショップ (1991).
- [7] 上田博唯, "インテリジェント映像編集技術," テレビジョン学会, インテリジェント映像メディアフォーラム'92 (1992).
- [8] 上田博唯, "FRIEND21プロジェクト," 平成4年電気情報関連学会連合大会 (1992).

1.4 国内雑誌寄稿論文

- [1] 上田博唯, "21世紀のヒューマンインタフェースを目指すFRIEND 21研究センター," 画像ラボ, 日本工業出版 (1990).
- [2] 上田博唯, "マルチメディア/ハイパーメディアとコンピュータ," コンピュータロール誌, コロナ社, pp.44-50 (1991).
- [3] 上田博唯, "次世代のユーザ・インタフェース," Industrial DESIGN, pp. 22-28 (1992).
- [4] 上田博唯, "画像認識技術で支援された映像情報オーサリングシステム," 映像インダストリアル, 産業開発機構 (1992).
- [5] 上田博唯, "画像認識技術を応用した映像編集支援システム," 画像ラボ, 日本工業出版 (1992).

1.5 国際会議招待講演

- [1] Ueda, H., Miyatake, T. and Yoshizawa, S., "IMPACT: An Interactive Natural-Motion-Picture Dedicated Multimedia Authoring System," フランス視聴覚研究所, Imagina, Proc. of IMAGINA'92, pp. II-25~II-33 (1992).
- [2] Ueda, H., "Agency Model as a Human Interface Architecture in the FRIEND21 Project," Interface Agents Symposium, MIT Media Lab. (1992)

- [2] Ueda, H., "Agency Model as a Human Interface Architecture in the FRIEND21 Project," Interface Agents Symposium, MIT Media Lab. (1992)

1.6 国際会議講演論文

- [1] Ueda, H. and Ejiri, M., "An Interactive Graphics Editor Handling Characters and Shapes in Arbitrary Size and Orientation," SPIE, JAPAN DISPLAY '83, Kobe, pp.530-533 (1988)
- [2] Ueda, H., et al., "A Multiprocessor System Utilizing Enhanced DSP's for Image Processing," IEEE International Conference on Systolic Arrays, San Diego, CA, pp.611-620 (1988)
- [3] Ueda, H., Miyatake, T. and Yoshizawa, S., "IMPACT: An Interactive Natural-Motion-Picture Dedicated Multimedia Authoring System," ACM CHI'91, pp.343-350 (1991).
- [4] Ueda, H., "FRIEND21 (Aiming for the Personalized Information Environment)," SIGGRAPH VIDEO REVIEW, ISSUE 79, CHI'92 Special Video Program, No.10, ACM (1992).
- [5] Ueda, H., Miyatake, T., Sumino, S. and Nagasaka, A., "Automatic Structure Visualization for Video Editing," ACM & IFIP INTERCHI'93, pp.137-141 (1993).
- [6] Ueda, H., Miyatake, T. and Yoshizawa, S., "IMPACT: Interactive Motion Picture Authoring system for Creative Talent," INTERCHI'93, Video Program, ACM (1993)

2. 共同執筆論文

2.1 国内一般論文

- [1] 藤方健二, 上田博唯, 江尻正員, "動的仮想キー方式漢字選択方式とその和文ワードプロセッサへの応用," 電学論(C), Vol. 101-C, No. 1, pp.1-8 (1982).

- [2] 江尻正真, 上田博唯, "知識に基づいた質問応答形個人識別方式の基本特性," 電学論(C), Vol. 102-C, No. 5, pp.117-124(1982).
- [3] 加藤寛次, 上田博唯, 柏岡誠治, 酒匂裕, 小畑修, "電子部品検査向きの小型画像処理装置SBIP," 日立評論, Vol.67, No.9, pp.71-74 (1982).

2.2 国内解説論文

- [1] 黒須正明, 上田博唯, "パーソナル情報環境一人にやさしいインタフェースの枠組み一," 電学論(D), Vol.111, No.12, pp.103 (1991).

2.3 研究会講演論文

- [1] 吉澤 聡, 上田博唯, 宮武孝文, "高精細画像を用いた動画像カットの生成法," 信学会 1990年秋季全国大会, D-337, p 6-339 (1990).
- [2] 宮武孝文, 吉澤 聡, 上田博唯, "フレーム相関係数の変化率に着目したカットの自動検出法," 信学会1990年秋季全国大会, D-299, p 6-301 (1990).
- [3] 炭野重雄, 上田博唯, 宮武孝文, 吉澤 聡, "実写画像とCGの合成に関する支援手法," 信学会1991年春季全国大会, D-451, p 7-163 (1991).
- [4] 宮武孝文, 吉澤 聡, 上田博唯, "ビデオ編集のためのシーン自動記述の一考察," 信学会1991年春季全国大会, D-499, p 7-211 (1991).
- [5] 比田井 裕, 上田博唯, 宇山政志, "エージェンシーモデルにおける個人情報管理の一考察," 人工知能学会研究会, SIG-HICG-9003-2 (2/15), pp.11-20 (1991).
- [6] 炭野重雄, 宮武孝文, 上田博唯, "動画像編集支援における個人適応化モデルの検討," ヒューマンインタフェース 44-21, pp. 157-164 (1992).
- [7] 岡田和久, 宇山政志, 上田博唯, "タスク意図同定による個人適応に関する一考察," ヒューマンインタフェース 44-20, pp. 149-156 (1992).
- [8] 長坂見朗, 宮武孝文, 上田博唯 "動画像情報リンク付けの一手法," テレビ学技報, Vol.16, No.10, pp.25-30 (1992)

(他多数、整理中)

2.4 国際会議講演論文

- [1] Nonogaki, H. and Ueda, H., "FRIEND21 Project: A Construction of 21st Century Human Interface," CHI'91, pp.407-414 (1991).
- [2] Nonogaki, H. and Ueda, H., "FRIEND21 Project: A Construction of 21st Century Human Interface," 15th COMPPUSAC (1991).

2.5 国際雑誌寄稿論文

- [1] Ejiri, M., Kashioka, S. and Ueda, H., "The Application of Image Processing Technology to Industrial Automation," Computers in Industry 5, Noth-Holland pp.107-113 (1984).

