

NAIST-IS-MT9651100

修士論文

DSP を用いた移動カメラ画像からの移動物体輪郭の
実時間抽出・追跡

松岡 隆志

1998年2月13日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学)授与の要件として提出した修士論文である。

松岡 隆志

指導教官： 横矢 直和 教授
千原 國宏 教授
竹村 治雄 助教授

DSP を用いた移動カメラ画像からの移動物体輪郭の 実時間抽出・追跡*

松岡 隆志

内容梗概

自律移動ロボットの実現には移動カメラ画像から実時間で移動物体抽出を行うことが必要となってくる。しかし、カメラが移動する場合、撮像された画像上において背景などの静止している物体にも見かけの動きが生じるため、一般に移動物体の抽出や追跡は困難となる。このため、移動カメラ画像から移動物体抽出を実時間で安定に行う手法の報告例は少ない。本論文では移動カメラ画像からの移動物体抽出を実時間で安定に行う手法について報告する。具体的には、従来より提案されていたロバスト推定による背景位置合わせと分裂統合型輪郭モデルを組み合わせて背景の動きを分離し移動物体を抽出、追跡する手法を提案する。本手法は分裂統合型輪郭モデルの採用により、複数移動物体の安定な抽出、追跡が可能である。実際に複数の DSP(Digital Signal Processor) ボードを用いたシステムに提案手法を実装し実時間での処理が可能であることを確認した。

キーワード

移動カメラ画像, 移動物体抽出・追跡, 動的輪郭モデル, DSP

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9651100, 1998年2月13日.

Realtime extraction and tracking of the moving object's contour from the moving camera using multi DSP boards*

Takashi Matsuoka

Abstract

Tracking moving objects from a sequence of images captured by a moving camera is necessary for realizing a vision system for an autonomous robot. However, the problem is difficult because the static background apparently moves. In this thesis, reported is a method of tracking moving objects from a sequence of images captured by a moving camera in real-time. First, the motion of the movement of the background is estimated by using robust estimation, and then split-and-merge contour models are used to extract and track moving objects. Split-and-merge contour models make it possible to extract and track multiple moving objects. The proposed algorithm is implemented on a image processing system which uses multi DSPs. The system successfully runs in real-time.

Keywords:

moving camera, extraction and tracking moving objects, active contour model, DSP

*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9651100, February 13, 1998.

目次

1. はじめに	1
2. 移動カメラ画像の背景位置合わせ	4
2.1 オプティカルフローの算出	4
2.2 背景の動きパラメータ推定	7
2.2.1 幾何学的変換	8
2.2.2 ロバスト推定	8
2.3 背景位置合わせの従来手法	10
2.4 本研究の背景位置合わせ手法	11
3. 動的輪郭モデルによる移動物体輪郭の抽出と追跡	13
3.1 動的輪郭モデルの概要	13
3.2 分裂統合型輪郭モデル	14
4. 実時間処理のためのアルゴリズム簡略化と DSP システムへの実装	16
4.1 実装に利用したオプティカルフロー算出アルゴリズム	16
4.2 実装に利用した背景の動きパラメータ推定アルゴリズム	19
4.3 実装に利用した動的輪郭モデルのアルゴリズム	20
4.4 実装に利用したシステム構成と実装した処理	21
5. 実験と結果	25
5.1 実験 1. 実画像による実験 - 移動物体が 1 つの場合	25
5.2 実験 2. 実画像による実験 - 移動物体が複数の場合	28
5.3 考察	28
6. おわりに	31
謝辞	33
参考文献	34

付録	37
A. 動画像処理用 DSP システム	37
A.1 DSP	37
A.2 DSP ボード	38
A.3 VIDEO ボード	42

図目次

1	背景位置合わせの概念図	5
2	ブロックマッチング法の概念図	7
3	特徴点抽出処理の流れ図	18
4	システム構成図	22
5	各 DSP の BC バス利用方法	24
6	背景位置合わせの処理の様子	26
7	輪郭モデルによる抽出結果	27
8	移動物体が 2 つの場合の輪郭モデルによる追跡結果	29
9	DSP ボードの構成	39
10	DSP ボードのメモリマップ	41

表目次

1	オプティカルフローの転送フォーマット	23
2	実験において利用したパラメータ	28
3	C40 での浮動小数点の取扱	37

1. はじめに

近年, 計算機の処理能力の飛躍的向上によって動画像を対象とした研究が盛んに行われるようになってきた. 動画像を対象とした研究の進展により, 自律移動ロボットや自動車の自動運転に画像センサから入力される動画像を利用する試みが発見可能なものとなってきている. 画像センサから入力される動画像を利用して突然視界に入ってくる移動する障害物に対処するためには取得される画像から移動物体を認識する必要がある. 従来, 静止カメラ画像に対して画像上の移動物体を抽出, 追跡する試みは数多くあり [1, 2, 3, 4], 実時間処理を可能としているものも見られる [5, 6, 7]. しかし, 自律移動ロボットなどへの応用を考慮する場合, 取得画像は移動カメラ画像となる. 移動カメラ画像では移動物体以外の領域である背景にも見かけの動きが生じるため, 静止カメラ画像に対する手法をそのまま適用した場合, 背景も移動物体として抽出されてしまう問題が発生する.

このため, 背景の見かけの動きと移動物体を分離する必要がある. 背景の見かけの動きと移動物体を分離することで, 静止カメラ画像に対する移動物体抽出手法を適用でき, 移動カメラ画像からの移動物体抽出を行うことができる. 背景の見かけの動きを分離する手法として, 画像各点での動きベクトルであるオプティカルフローを画像全体で算出し, そのオプティカルフローベクトルの示す方向からカメラの移動パラメータを推定し, 推定されたカメラ移動パラメータとオプティカルフローの照合を行うことで移動物体の動きを分離する手法 [8, 9] や, カメラが人工環境を移動すると仮定し, 人工環境に多い垂直エッジを利用してカメラ移動パラメータの推定を行い, 推定されたカメラ移動パラメータを利用して移動物体の検出を行う手法 [10] などがある. これらの手法は正確なオプティカルフローが得られない場合や移動物体がカメラと同じ方向に運動している場合には背景の見かけの動きの分離は困難となる. これに対して背景位置合わせを用いて背景の見かけ動きを分離する手法 [11, 12, 13] が提案されている. 背景位置合わせとはフレーム間での背景の動きを幾何学的変換で近似できると仮定し, 推定された変換パラメータを基に背景の動きを分離, 除去する手法である. 背景位置合わせにより背景の見かけの動きを分離する手法はカメラと同じ方向に移動する移動物体に対しても有効な抽出手法である. これらの従来手法は 2 段階のアフィン変換を行う

[11], もしくは M-推定 [14] および遺伝的アルゴリズムを用いて精密化を行う [12] など, 計算量の多い処理を伴うため, 実時間での処理を行うには至っていない.

静止カメラ画像からの移動物体抽出手法には様々な手法があるが, 動的輪郭モデル [1, 2, 17] や, アクティブネット [3, 4] などのエネルギー最小化による手法は抽出洩れが発生しにくく, また種々の画像特徴を総合的に利用可能であるといった点から有効なものである. この中で輪郭モデルを高次元に拡張したモデルであるアクティブネットは輪郭モデルと比較して計算量が多く, 並列化などの手法を用いないと実時間処理は難しい. 輪郭モデルでは一般に初期輪郭の設定が必要である. 初期輪郭を必要としない輪郭モデルとして, 輪郭モデルの分裂や統合を行い, 画像上のノイズに捕獲された輪郭モデルを分離, 消滅させることで初期輪郭に依存せず領域を抽出可能な手法が提案されている [2, 17]. 具体的にはそれぞれ, 三角格子を画像に重ねて各格子の属性を利用することで輪郭モデルを分裂・統合する手法 [17], および制御点間の交差を基準に輪郭モデルを分裂・統合する手法 [2] である. 前者の手法は後者の手法と比較して格子の属性を保持するための記憶領域が余分に必要となり, また格子の情報と制御点の情報を扱う必要となる. これに対し制御点間の交差を基準に分裂・統合する手法では制御点の情報のみを扱えばよく, 記憶領域と処理時間の双方の点から有利な手法である.

これらの手法を実時間で処理するためには並列分散処理や, 処理の簡略化などが必要となる. 動画の並列分散処理に有効な方法として動画処理用 DSP ボード [15] が提案されている. 従来, DSP ボードを利用してオプティカルフロー算出 [16] や前方の車との車間距離計測 [15] が行われている. また, 静止カメラ画像からの移動物体抽出を実時間で行う手法 [7] が提案されているが, 移動カメラ画像からの移動物体抽出への応用例は報告されていない.

本研究では移動カメラ画像からの移動物体抽出を実時間で安定に行う手法の開発を目的とする. このため, まず, 移動カメラ画像からの移動物体抽出処理を大きく背景の動きの分離と移動物体の抽出の 2 段階に分けることにする. 背景の動きの分離には移動物体の移動方向によらず安定に分離可能な背景位置合わせ手法を提案し利用する. また, 移動物体の抽出には制御点間の交差判定により分裂・統合する分裂統合型輪郭モデルを用いることで初期輪郭の制限をなくし, 複数移動

物体の抽出を可能とする。実時間処理のため、分散処理として動画像処理用 DSP ボード [15] を採用し、処理の簡略化として背景位置合わせにおける背景の動きパラメータの算出の際、パラメータ算出に用いる参照点数を減らす。これにより生じる誤差は、輪郭モデルを用いることで吸収し、安定性を確保する。

以下、第 2 章で移動カメラ画像を処理する場合において問題となる背景の動きを除去する背景位置合わせ手法について述べ、第 3 章で本研究で移動物体の抽出、追跡に利用した輪郭モデルの詳細について述べる。続いて第 4 章で DSP 上への実装方法について述べ、第 5 章で実装したシステム上で行った実験とその結果を示す。最後に第 6 章で本研究のまとめと今後の課題について述べる。

2. 移動カメラ画像の背景位置合わせ

本章では移動カメラ画像での背景位置合わせの手法について概観し、次に、本研究で利用する手法について述べる。背景位置合わせとは移動カメラ画像において画像上で最も大きい領域を占める背景領域の画像上での位置を連続するフレーム間で合わせることで背景の見かけの動きを除去する手法である。背景位置合わせの概念図を図1に示す。背景位置合わせとは具体的には、図1に示されるようにフレーム間で背景の動きを近似することのできる幾何学的変換パラメータの推定を行い、推定された変換パラメータで前フレームを変換し、次フレームでの背景の位置と合わせることで背景の見かけの動きを除去する手法である。以下、背景の位置合わせに利用するオプティカルフローの算出手法、実際の背景の動きパラメータの推定手法、背景位置合わせの従来手法、本研究での背景位置合わせ手法について順に述べる。

2.1 オプティカルフローの算出

オプティカルフローとは動画像において連続するフレーム間の対応する点を画像上で結ぶベクトルのことである。このオプティカルフローの算出には大きく2種類の方法がある。これらは、それぞれ勾配法 [18]、ブロックマッチング法と呼ばれている。以下にこれらの手法の概要を示す。

1. 勾配法

まず、連続する2フレーム間で物体の輝度変化が微小であると仮定する。このとき物体の輝度を $I(x, y, t)$ ((x, y) は画像上の点の座標, t は時刻) とすると以下の式が成り立つ。

$$I(x, y, t) \approx I(x + \delta x, y + \delta y, t + \delta t)$$

右辺を Taylor 展開すると

$$I(x, y, t) \approx I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + O^2$$

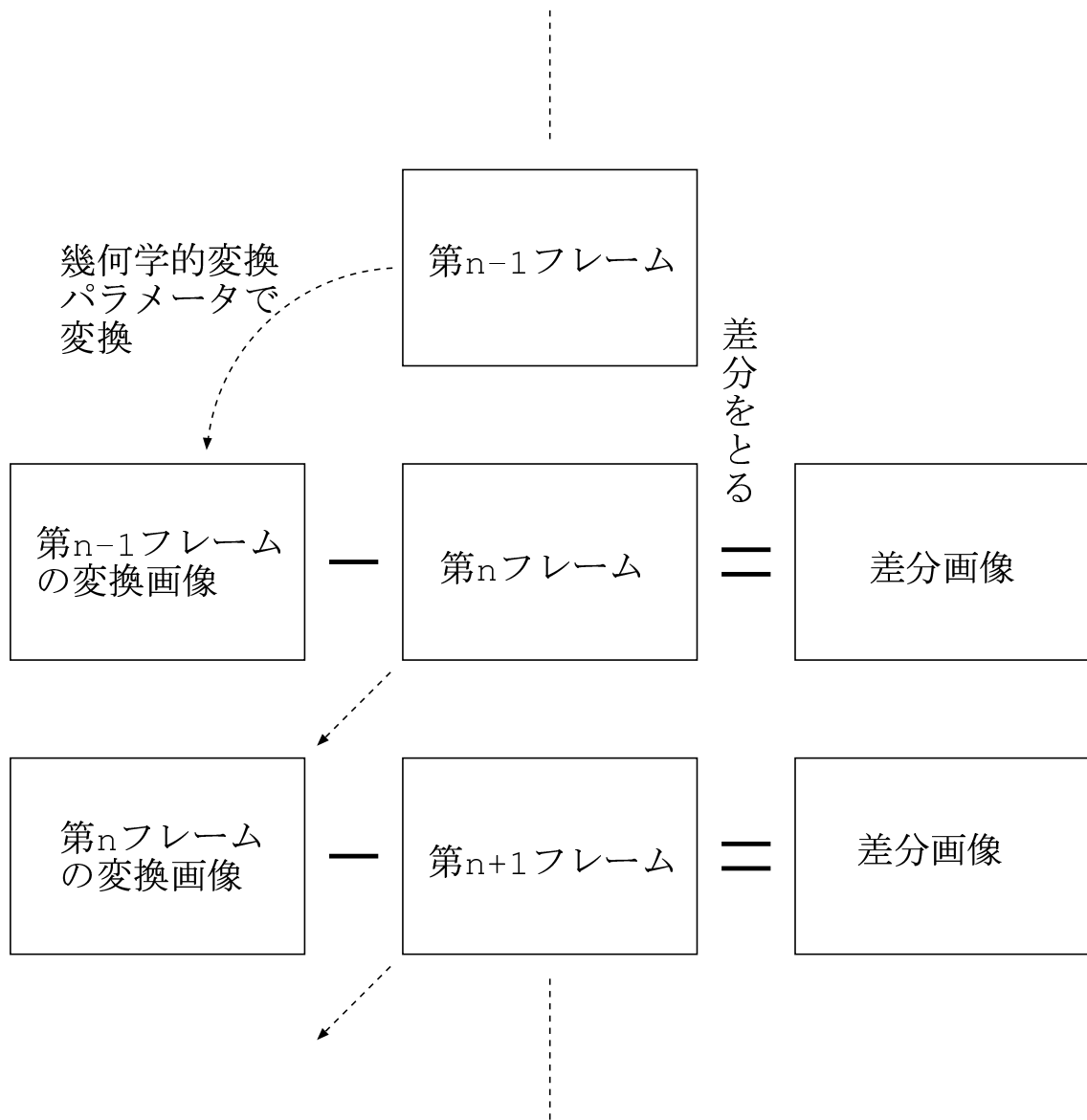


図 1 背景位置合わせの概念図

となる. ここで O^2 は I の 2 次以上の微分の項である. 両辺から $I(x, y, t)$ を引き, 2 次以上の項 O^2 は微小であることから無視すると

$$\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0$$

と書ける. ここで $(\partial I / \partial x, \partial I / \partial y)$ は求めるオプティカルフローベクトルであることから, この拘束式を利用することでオプティカルフローを算出する手法である. しかし, この場合拘束式が 1 種類しか得られない. そこでこの解決法として, 画像に 2 種類のフィルタリングを行い複数の拘束式を成立させる手法 [7] や画像の色情報を利用して複数の拘束式を構成する手法 [19] などが提案されている. 勾配法には,

- 計算コストが低く, 実時間処理においても密なフローを得やすい.
- 精度が確保できれば容易にサブピクセル単位でオプティカルフローを算出できる

という利点がある. しかし逆に

- 照明条件によっては輝度不変が成り立たない
- 微分を利用するため時間的空間的に高い解像度が必要

という欠点があり, 現実には高い精度を得ることは困難である.

2. ブロックマッチング法

連続する 2 フレーム間で対応する点をテンプレートマッチングを用いて探索することでオプティカルフローを算出する手法である. 具体的にはまず, 最初のフレームで画像上に適当な大きさの矩形領域 (テンプレート) をオプティカルフローを算出したい点 (参照点) を中心として設定する. 次に, 次のフレーム上に参照点を中心とした探索領域を設定し, 設定したテンプレートと類似度の高い矩形領域の探索を行う. そして, もっとも類似度の高い矩形領域の中心 (対応点) が参照点の移動先であるとみなし, 参照点と対応点を結ぶベクトルとしてオプティカルフローを算出する手法である. ブロックマッチングで利用される類似度としてはテンプレートを構成する各点で

の輝度差の絶対値の総和や正規化相互相関などが挙げられる。ブロックマッチングによるオプティカルフロー算出の概念図を図2に示す。ブロックマッチング法は勾配法と比較して次のような利点がある。

- 照明条件に左右されにくい
- 時間的空間的解像度に対する要求が低い

ブロックマッチング法は一般に勾配法と比較すると正確なフローが得られる手法である。しかし同時に計算コストが高く、実時間処理への応用には工夫が必要となるという欠点がある。

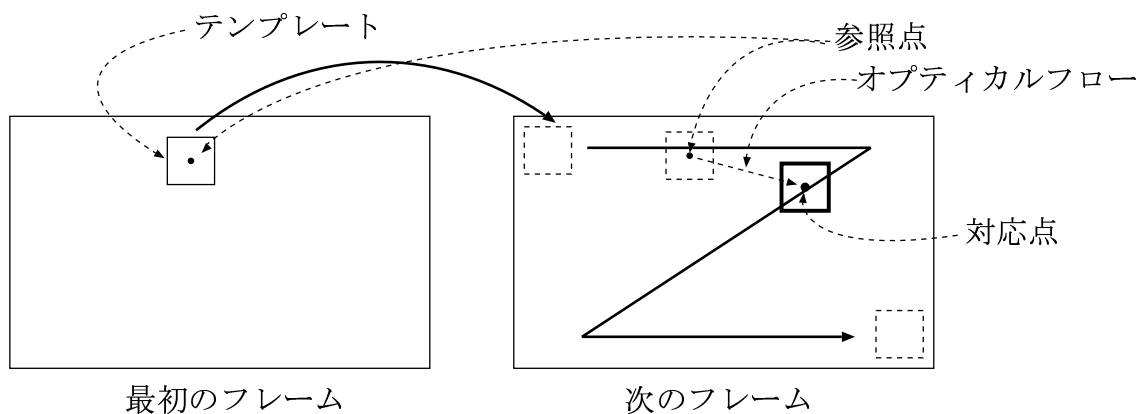


図2 ブロックマッチング法の概念図

2.2 背景の動きパラメータ推定

一般に背景位置合わせ手法では背景の動きを幾何学的変換で近似できると仮定することが多い。以下で幾何学的変換および背景の動きパラメータ推定に利用したロバスト推定について順に説明する。

2.2.1 幾何学的変換

背景位置合わせにおいて利用される幾何学的変換とは、画像の位置 (x, y) における情報 (輝度など) を任意の多項式 $f(x, y), g(x, y)$ を用いて新しい画像 (X, Y) に変換する式 (1) で表されるような変換のことである。

$$\begin{aligned} X &= f(x, y) \\ Y &= g(x, y) \end{aligned} \quad (1)$$

画像処理に用いられる代表的な幾何学的変換としてアフィン変換や射影変換が知られている。これらはそれぞれ式 (2), 式 (3) で表されるものである。

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (2)$$

$$\frac{1}{c_1x + c_2y + d} \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (3)$$

オプティカルフローを利用した背景位置合わせ手法に用いられる場合, (x, y) にはオプティカルフローを求める際に使用した参照点を利用し, (X, Y) はこの参照点の座標と推定されたオプティカルフローベクトル (u, v) との和 $(x + u, y + v)$ で表される対応点を利用する。

2.2.2 ロバスト推定

幾何学的変換パラメータを推定するにあたって、求めたオプティカルフローを全て満足するパラメータを推定することはできない。そこで、統計的手法によってパラメータ推定を行う。統計的手法でパラメータの推定を行う場合、最小2乗法を利用するのが一般的である。しかし、背景位置合わせにおいては最小2乗法を利用すると移動物体上の特徴点のオプティカルフローも含めて背景の動きパラメータの推定を行うこととなり、推定精度が低下する。この解決法として、移動物体上の特徴点から得られたオプティカルフローを外れ値とみなし除去する手法であるロバスト推定 [14] を利用して背景の動きパラメータ推定を行う方法がある。

ロバスト推定では得られたデータに外れ値が半分以上含まれている場合、正しい値の推定はできないが、背景位置合わせ手法では背景の動きが支配的であると仮定しており、外れ値となる移動物体上の参照点は半分以下であるためロバスト推定の適用が可能である。

ロバスト推定とは得られたデータの中に全体の傾向と全く異なった外れ値と呼ばれるデータが含まれている場合にこのような外れ値を除外したデータの傾向を検討するために用いられる手法である。ロバスト推定には以下のようなものがある。

(1) LMedS 推定

$$\text{Minimize}\{\text{med}(|r_i|)\}$$

残差 r_i (実測値と推定値の差) の絶対値の中央値が最小になるようなパラメータを最適なパラメータとして推定を行う手法。ただし残差の絶対値 $|r_i|$ の代わりに残差の 2 乗 r_i^2 が用いられることもある。一般に中央値は平均値よりも外れ値に対するロバスト性が高いため、中央値を最小化することで外れ値に対してロバストな推定結果を得ることが可能である。

(2) LTS 推定

$$\text{Minimize} \sum_{i=1}^h r_{i:n}^2$$

まず、残差 r_i を小さいものから順にソートし、ソート結果を $r_{i:n}$ とする。次にこのソート結果を 2 乗したものを h 個 (ただし h は $0 \leq h < n$ を満たす任意定数) 合計したものが最小となるパラメータを最適なパラメータとして推定を行う手法。外れ値は一般に残差が大きくなるため、残差の大きい値を除外して推定を行うことで外れ値に対してロバストな推定結果を得ることが可能である。

(3) M-推定

$$\text{Minimize} \sum_{i=1}^n \rho(r_i)$$

まず、 $\rho(-t) = \rho(t)$ を満たし、 $t = 0$ で唯一の最小値を持つ任意の関数 $\rho(x)$ を設定する。次に残差 r_i を関数 $\rho(x)$ へ代入し合計する。この合計が最小と

なるパラメータを最適なパラメータとして推定する手法. 関数 $\rho(x)$ は統計処理の対象に応じて選択される. M-推定には最小 2 乗法も含まれるが, 一般に M-推定という場合, より高次の多項式を利用した推定法を指す. 一般に再帰的計算により推定を行うこととなり, 初期値の設定が必要となる.

ここで 3 番目の手法の M-推定法は再帰的計算の必要性から計算量が多く実時間処理には適していない. 次に LMedS 推定と LTS 推定であるが, いずれも残差のソーティングが必要である. しかし, LTS 推定の場合さらに 2 乗処理が必要であるため計算量が若干増加する上, LMedS 推定と比較して特に推定誤差が改善されるわけではない.

2.3 背景位置合わせの従来手法

背景位置合わせを用いた移動物体抽出を行う場合, カメラの運動に関する情報は取得画像を除いて利用しない. 従って, カメラの運動情報を利用した背景の動き推定は不可能である. この場合, 一般には背景の動き推定は不可能となる. そこで背景位置合わせを利用する場合, 一般に背景の動きに対して以下のような仮定をおく.

仮定:背景の動きは画像上で支配的な動きである

これは画像上で背景の占める割合が画像全体の 50%を越えていることを表している. 移動カメラ画像のみから移動物体を抽出する研究においては一般的に使われている仮定である. 一般的に移動カメラ画像においては移動物体領域の合計が 50%を越えることは少なく, 実用的な範囲において背景位置合わせ手法の適用範囲を制限するものではないといえる.

背景位置合わせによる移動カメラ画像からの移動物体抽出の従来手法として, 寺久保らの 2 段階アフィン変換を行う手法 [11] や, 朝岡らのロバスト統計を利用する手法 [12] がある. これらはそれぞれ以下のような手法である.

2 段階アフィン変換を利用する手法

移動カメラ画像の背景の動きをアフィン変換で近似する手法である. まず, 画像上に粗く参照点を設定し, 設定した参照点におけるオプティカルフロー

を算出する。次に、算出したオプティカルフローをもとに、利用した参照点の組み合わせすべてについてアフィン変換のパラメータを計算する。さらに、それぞれのアフィン変換パラメータで前のフレームに対してアフィン変換を実行し、現在フレームとの輝度差分の平均を評価値として最適と思われるアフィンパラメータを選択する。最後に、選択したアフィンパラメータを用いて前のフレームに対しアフィン変換を実行し、現在のフレームとの差分情報をもとに移動物体を抽出する。

ロバスト統計を利用する手法

移動カメラ画像の背景の動きを射影変換で近似する手法である。まず、画像上に粗く参照点を設定し、設定した参照点におけるオプティカルフローを算出する。さらに、算出したオプティカルフローをもとに遺伝的アルゴリズムを利用した LMedS 推定によって射影変換パラメータを推定する。次に求めた射影変換パラメータを初期値として M-推定法による精密化を図ることでより精度の高い射影変換パラメータを推定する。最後に、推定されたパラメータを基に前のフレームに対し射影変換を行い、変換画像と現在のフレームとの差分を生成し、ブロック単位で差分を検証することで移動物体を抽出する。

これらの手法はいずれも精度を重視しており、速度に関してはあまり重視されていない。

2.4 本研究の背景位置合わせ手法

本研究では実時間処理を目指すため、まず速度を重視し、次に精度を考慮する。以下に本研究で利用した背景位置合わせ手法について述べる。

まず、本研究では背景の動きに関して次のような仮定をおく。

仮定 1:背景の動きは画像上で支配的な動きである

仮定 2:背景の動きはアフィン変換で近似できる

背景の動きをモデル化するには様々な数学的表現が可能である。表

現する式を複雑にすればより多くの種類の動きを正確に近似することが可能となるが、処理に必要な時間は増える。従って、どの程度の近似を行うかが重要となる。現在モデル化しようとしている背景の動きの原因はカメラの運動にある。一般にカメラが運動する場合、連続するフレーム間では視線の変化は微小である。ここで、背景の奥行き方向の変化が微小であるか、視点から背景までの距離が視点の移動距離に比して十分大きいならば、連続するフレーム間で起こり得る背景画像の変形は平行移動、回転、拡大縮小などのアフィン変換で近似できるものばかりであり、高次の多項式変換などのより複雑な式を利用する必然性は少ない。また、計算量の点からもアフィン変換は線形な変換であり高い計算コストを要求しない。また、背景の奥行き方向の変化を無視できない場合は、背景の動きは局所的な動きの集合となり、大局的に背景の動きをモデル化することは困難である。すなわち、現在の計算機の能力で実時間処理は難しい。従って、現状の計算機能力を仮定する場合、背景の動きを表現するにあたって背景の動きをアフィン変換で近似することは必要十分と考えられる。

この仮定を用いて次のように背景の動きを分離し、差分画像を生成する。

1. 連続する2フレーム間でブロックマッチングにより参照点でのオプティカルフローを算出する。これは第4章で述べるように、勾配法では移動物体抽出が可能な精度の背景位置合わせが行えなかったためである。
2. 算出されたオプティカルフローをもとに背景の動きパラメータ(フレーム間のアフィン変換パラメータ)をロバスト統計の中でも処理速度の点から有利である LMedS 推定を利用して推定する。
3. 推定された背景の動きパラメータにより前のフレームを変換する。
4. 生成した変換画像と現在入力されたフレームとの差分をとる。

3. 動的輪郭モデルによる移動物体輪郭の抽出と追跡

本章では背景の見かけの動きがない場合に移動物体を追跡する手法である分裂統合型輪郭モデル [2] について述べる. 分裂統合型輪郭モデルとは動的輪郭モデル (Snakes)[1] を基礎として複数物体の抽出, 追跡へ応用したものである. 以下で, 動的輪郭モデル, 分裂統合型輪郭モデルについて順に説明する.

3.1 動的輪郭モデルの概要

動的輪郭モデルとは正則化の手法を利用した領域分割手法の 1 種であり, 以下のエネルギー関数を最小にするパラメータ表現された閉曲線 $\mathbf{v}(s) = (x(s), y(s)) (0 \leq s < 1)$ に収束するモデルとして表される.

$$E_{snakes} = \int_0^1 (E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))) ds \quad (4)$$

以下で各項について順に説明する.

- 内部エネルギー $E_{int}(\mathbf{v}(s))$

輪郭の滑らかさを表すエネルギー. 一般に式 (5) で表される $E_{spline}(\mathbf{v}(s))$ が用いられることが多い.

$$\begin{aligned} E_{int}(\mathbf{v}(s)) &= E_{spline}(\mathbf{v}(s)) \\ &= \frac{1}{2} (\alpha |\mathbf{v}_s(s)|^2 + \beta |\mathbf{v}_{ss}(s)|^2) \end{aligned} \quad (5)$$

ここで $\mathbf{v}_s(s), \mathbf{v}_{ss}(s)$ はそれぞれ $\mathbf{v}(s)$ の 1 次, および 2 次微分である. また α, β は重み係数である,

- 画像エネルギー $E_{image}(\mathbf{v}(s))$

輪郭モデルを特徴的な画素値を持つ領域へ収束させるエネルギー. 画像の輝度 $I(\mathbf{v}(s))$ を用いた式 (6) が用いられることが多い. この場合, 輪郭モデルは画像上で輝度変化の大きい部分へ収束しようとする.

$$E_{image}(\mathbf{v}(s)) = E_{edge}(\mathbf{v}(s))$$

$$\begin{aligned}
&= -\frac{1}{2}\gamma|\nabla I(\mathbf{v}(s))|^2 \\
&= -\frac{1}{2}\gamma\left(\left(\frac{\partial I(\mathbf{v}(s))}{\partial x(s)}\right)^2 + \left(\frac{\partial I(\mathbf{v}(s))}{\partial y(s)}\right)^2\right) \quad (6)
\end{aligned}$$

- **外部エネルギー** $E_{con}(\mathbf{v}(s))$

内部エネルギーおよび画像エネルギー以外に輪郭モデルに与えるエネルギーのこと。輪郭モデルにさまざまな特徴を付加する場合に利用する。本研究では次節で述べる面積項がこれに相当する。

このように定義される動的輪郭モデルは次のような特徴を持つ。

- 滑らかな抽出輪郭線が得られる
- ある特徴を持った領域の抽出が可能

3.2 分裂統合型輪郭モデル

前節で説明した動的輪郭モデルには、予め対象となる物体の近傍に初期輪郭を設けないと物体の抽出に失敗するという欠点がある。このことは本研究においては常に入力される動画像を人間が監視し、画像に応じて初期輪郭を設定しなければならないということになる。しかし、例えば1度抽出に失敗し、再び初期輪郭を設定しなければならない場合などにおいては、この人間の介在によって実時間性が損なわれるという問題がある。そこで、本研究では実時間処理において問題となるこの人間の介在をなくすため、初期輪郭の制限を解決した従来手法である分裂統合型輪郭モデル [2] を用いることにした。分裂統合型輪郭モデルとは輪郭モデルを次のように分裂または統合することで複数物体の抽出を可能とする手法である。

- エネルギー関数に面積項を導入する、面積項の影響を強くすることで輪郭モデルがより強く収縮しくびれた部分に自己交差が発生する。この輪郭モデルの自己交差を検知して分裂させる。
- 異なる輪郭モデルが衝突した場合に相互交差を検知して輪郭モデルを統合する。

以下でそれぞれについて説明する.

まず分裂手法についてであるが, エネルギー関数に式 (7) 表される輪郭モデルの内部面積 E_{area} を導入する.

$$E_{area} = w_{area} \int_s (\mathbf{v}(\mathbf{s}) \times d\mathbf{v}(\mathbf{s})) \quad (7)$$

この項を導入することにより複数の物体が存在する場合に輪郭モデルが物体間の空間へ侵入することが可能となる. この面積項の重みを大きくすると物体間の空間で輪郭モデルが自己交差を発生する. そこでこの自己交差を判定し, 交差部分で2つの輪郭モデルへ分割することで輪郭モデルの分裂を実現している.

次に, 統合手法であるが, 前述の分裂によって複数の輪郭モデルが画面上に存在することとなる. 従って, 抽出されている2つの移動物体が接近した場合これらを抽出している輪郭モデルは衝突し, 輪郭モデル間の相互交差が発生する. この相互交差を判定し, 相互交差が起こった場合に2つの輪郭モデルを包含する輪郭モデルを新たに定義し, 新しい輪郭モデルで衝突した2つの輪郭モデルを置き換えることで統合を実現している.

分裂統合型輪郭モデルによる移動物体の追跡は以下のように行う. まず, 抽出物体の移動を考慮し, 前フレームでの輪郭モデルによる抽出結果を拡大する. 次に拡大した輪郭を初期輪郭として輪郭モデルの収縮を行う. この時拡大する量はフレーム間での移動量を参考に決定する.

これを差分画像に適用することで本研究では入力される動画像にかかわらず常に同じ初期輪郭を用いることができ, 人間の介入の必要の少ない安定した実時間移動物体輪郭抽出を目指し. さらに, 同時に複数の移動物体輪郭の抽出が可能となる.

4. 実時間処理のためのアルゴリズム簡略化と DSP システムへの実装

前章までに述べたアルゴリズムを実時間で処理するためには分散処理や処理の単純化が必要である。本研究ではアルゴリズムの簡略化を行い(株)富士通製動画処理用 DSP システム [15] 上に実装し、実時間処理を実現した。利用した DSP システムの詳細は付録 A を参照のこと。以下オプティカルフロー算出、背景の動きパラメータ推定、分裂統合型輪郭モデルそれぞれのシステムへの実装および構築した DSP システムの構成について説明する。

4.1 実装に利用したオプティカルフロー算出アルゴリズム

第 2.1 節で述べた手法を実際に DSP 上へ実装する。本研究では、まず速度を重視することから勾配法によってオプティカルフローを算出する手法について検討し予備実験を行ったが、十分満足できる結果を得ることはできなかった。この理由として、勾配法を利用する場合、現状のシステムで可能なフレームレートでは勾配法の要求する時間解像度を満足できないことが挙げられる。そこで、本研究ではブロックマッチング法を用いてオプティカルフローを算出することとした。しかし、ブロックマッチング法には計算コストが高いという欠点がある。そこでオプティカルフローを算出する点の数を極力減らすことで実時間性を確保することとした。これにより背景の位置合わせ精度が若干低下することになるが、アフィン変換のパラメータ数は少ないため数多くのオプティカルフローを利用した場合でも計算コストと比較して誤差が大きく改善されないこと、および輪郭モデルによる移動物体抽出を行うことから大きな問題とならないと判断した。オプティカルフロー算出に利用する参照点には画像上で参照点の周囲の輝度分散の大きい特徴点を選択した。特徴点の選択は以下のように行った。

1. 画像上からランダムに 1 点を選択する
2. 選択した点のまわり $T_s \times T_s$ ピクセルのテンプレートの輝度分散を算出する

3. 算出した輝度分散が閾値以上となる点を特徴点とし 5. へ処理を移す. 輝度分散が閾値以下の場合は 1. へ処理を移す.
4. 輝度分散が閾値以上となる点が N_{max} 回を越えても得られなかった場合には, N_{max} 回のうち最も輝度分散が大きかった点を特徴点として使用する.
5. 1. ~ 4. を N_f 回繰り返し画像上から特徴点を N_f 個抽出する.

処理の流れ図を図 3 に示す. 輝度分散の算出には式 (9) を利用した. 特徴点選択時の輝度分散の閾値は, 高いと実時間処理において十分な速度を得ることが困難であり, 低いとオプティカルフロー計算の精度が低下することから, 実験的に選択した.

$$V_I = \sum_{x,y \in D} (I_{x,y} - Ave.(I_{x,y}))^2 \quad (8)$$

D : 参照点を中心とした正方形領域

$I_{x,y}$: 点 (x, y) における輝度

$Ave.(I_{x,y})$: 領域 D 内部の輝度 I の平均値

このようにして抽出した特徴点でのオプティカルフローをブロックマッチングにより算出した. 対応点探索領域は特徴点を中心とした正方形領域とした. 探索領域は可能な限り大きい方が良いが, 処理時間との兼ね合いもあるため実験的に大きさを決定した. 探索時のテンプレートの類似度としては各点での輝度差の絶対値の総和を用いた. さらに残差逐次検定法 (SSDA 法) を利用することで高速化を図っている. 残差逐次検定法とは以下のような手法である.

1. 輝度差の絶対値の総和の最小値を保持しておく.
2. 輝度差の絶対値の総和を計算する過程で算出される部分和を最小値と比較する.
3. 部分和が最小値を越えない場合次の輝度差を加えて部分和を更新する
4. 部分和が最小値を越えた場合計算を打ち切る

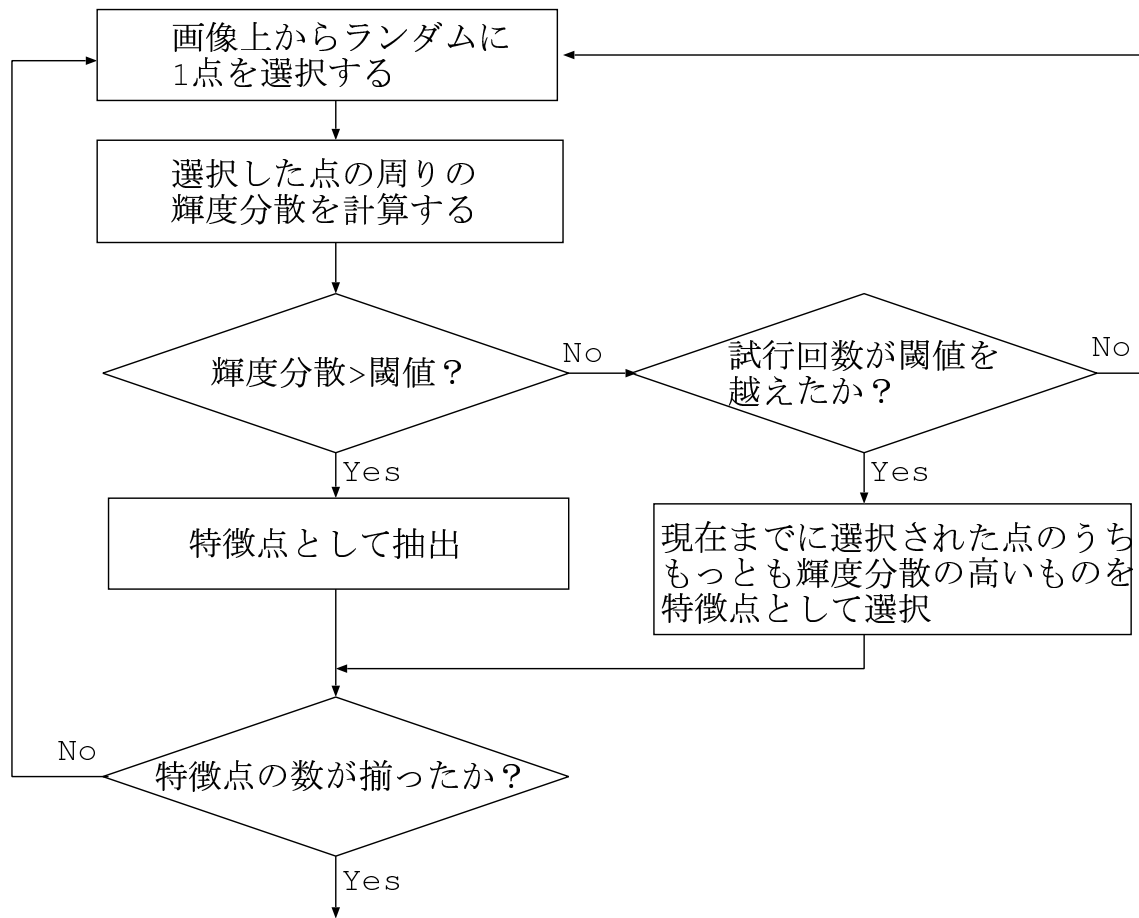


図 3 特徴点抽出処理の流れ図

4.2 実装に利用した背景の動きパラメータ推定アルゴリズム

背景の動きパラメータ推定は前節で算出したオプティカルフローを用いて LMedS 推定によって行う。具体的な手順は次の通りである。

1. 抽出した特徴点からランダムに N_s 点を選択する
2. 選択した N_s 点を正しく変換するアフィンパラメータを算出する
3. 算出したアフィンパラメータを用いて抽出した特徴点全てを変換する
4. 変換後の点と実際に算出されたオプティカルフローとの差分ベクトルを計算する
5. 差分ベクトルの大きさのメディアンを計算する。
6. 1. ~ 5. を k 回繰り返して差分ベクトルの大きさのメディアンが最小となる時に利用したアフィンパラメータを背景の動きパラメータとする。

N_s は 3 以上の定数であり、 N_s が 4 以上の時は一般に算出に最小 2 乗法を用いる。今回は速度を重視する点から N_s を 3 としてアフィンパラメータを算出した。

今、画像中で移動物体の占める領域の割合を ϵ 、対応点が正しく求まる確率を q とすると、以上の操作を k 回繰り返した場合に正しく背景の動きパラメータが推定される確率は

$$p(\epsilon, q, k) = 1 - \left(1 - ((1 - \epsilon)q)^3\right)^k \quad (9)$$

と表される。(9) 式から $\epsilon = 0.3, q = 0.7$ の場合、 $k = 40$ とすることで $p \approx 0.993$ となる。本研究では実時間性との兼ね合いから k をあまり大きくすることはできない。もし、 $p \approx 0.993$ であれば、秒 7 フレームの場合、背景の動きの推定誤りは 20 秒に 1 フレームとなり実用上ほぼ十分な精度であること、および、現在のシステムにおいては処理速度の関係から k をこれ以上にすることは困難であるため、 k を 40 と設定した。

変換パラメータにより前フレームを変換し現在のフレームとの差分をとり差分画像を生成する。差分画像の生成は実験的に閾値を設定し、閾値処理により 2 値化を行った。

4.3 実装に利用した動的輪郭モデルのアルゴリズム

式(4)の動的輪郭モデルのエネルギー関数の定義において, パラメータ s は連続的な量として定義されている. しかし, 実装においてはこれを離散的なものとする必要がある. そこで動的輪郭モデルを離散的な点の集合として実装する. まず連続的な量であるパラメータ s を離散的な量 i , (ただし $0 \leq i < n$, n は輪郭モデルを構成する点の数) で置き換えた. これに伴って $\mathbf{v}(s) = (x(s), y(s))$ は $\mathbf{v}_i = (x_i, y_i)$ に置き換えられ, $\mathbf{v}(s)$ の 1 次および 2 次微分は次のようになる.

$$\begin{aligned}\mathbf{v}_s(s) &= \mathbf{v}_i - \mathbf{v}_{i-1} \\ \mathbf{v}_{ss}(s) &= \mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}\end{aligned}$$

従って, 第 3 章で述べたエネルギー関数の各項は次のように書ける.

$$\begin{aligned}E_{int}(\mathbf{v}(s)) &= E_{spline}(\mathbf{v}(s)) \\ &= \frac{1}{2} \left(\alpha |\mathbf{v}_i - \mathbf{v}_{i-1}|^2 + \beta |\mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1}|^2 \right) \\ E_{image}(\mathbf{v}(s)) &= E_{edge}(\mathbf{v}(s)) \\ &= -\frac{1}{2} \gamma |\nabla I(\mathbf{v}(s))|^2 \\ &= -\frac{1}{2} \gamma \left(\left(\frac{\partial I(\mathbf{v}_i)}{\partial x_i} \right)^2 + \left(\frac{\partial I(\mathbf{v}_i)}{\partial y_i} \right)^2 \right) \\ E_{area} &= w_{area} \sum_i (x_i(y_{i+1} - y_i) - y_i(x_{i+1} - x_i))\end{aligned}$$

本研究では前節で生成した 2 値化差分画像に輪郭モデルを適用する. この時, 画像エネルギー E_{image} は更に簡単に以下のように書ける.

$$E_{image}(\mathbf{v}(s)) = \begin{cases} -\frac{1}{2} \gamma & (\text{輝度 } I \neq 0 \text{ の時}) \\ 0 & (\text{輝度 } I = 0 \text{ の時}) \end{cases}$$

この離散化したエネルギー関数を利用し輪郭モデルを構成する各点を収束させることで移動物体輪郭を抽出する. 収束にはグリーディアルゴリズム [20] を用いた. これは輪郭モデルの各構成点で

1. 近傍探索領域を設定する

2. 点を近傍探索領域のいずれかの点へ移動する
3. 移動後のエネルギーを算出する
4. 近傍探索領域に含まれる点のうち最もエネルギーが低くなる点を新たな輪郭モデルの構成点とする

という処理を繰り返し行うことでエネルギー最小化を目指す方法である。

本研究で利用した DSP ボードでの並列処理の場合、次段のボードでの収束結果を利用することはできない。このため、各ボードで一定回数収縮させることとし、収束判定は行っていない。従って、各ボードでは各フレームで常に直前のフレームにおける各ボードでの収縮結果が初期輪郭として利用されることになる。また、各フレームで前フレームの輪郭モデルの収縮結果を拡大したものを初期輪郭として利用するが、この時の拡大方法は次の通りである。

- \mathbf{v}_{i-1} と \mathbf{v}_{i+1} を結ぶ直線の法線ベクトルを求める (この時新しいベクトルの向きは $\overrightarrow{\mathbf{v}_{i-1}\mathbf{v}_{i+1}}$ を反時計周りに 90 度回転した方向である)
- 求めた法線ベクトルの単位ベクトルを定数倍し、 \mathbf{v}_i に足す

ただし \mathbf{v}_i は輪郭モデルを構成する各点であり、添字 i は輪郭モデルを構成する点を時計周りに数えた時の番号である。

4.4 実装に利用したシステム構成と実装した処理

実装のために構築した DSP システムの構成を図 4 に示す。以下に各ボードでの処理を簡単に述べる。

- DSP1 ～ DSP3
特徴点を 6 点ずつ設定し、対応点を探索することでオプティカルフローを計算する。
- DSP4
背景の動きパラメータ (アフィン変換パラメータ) を推定する。

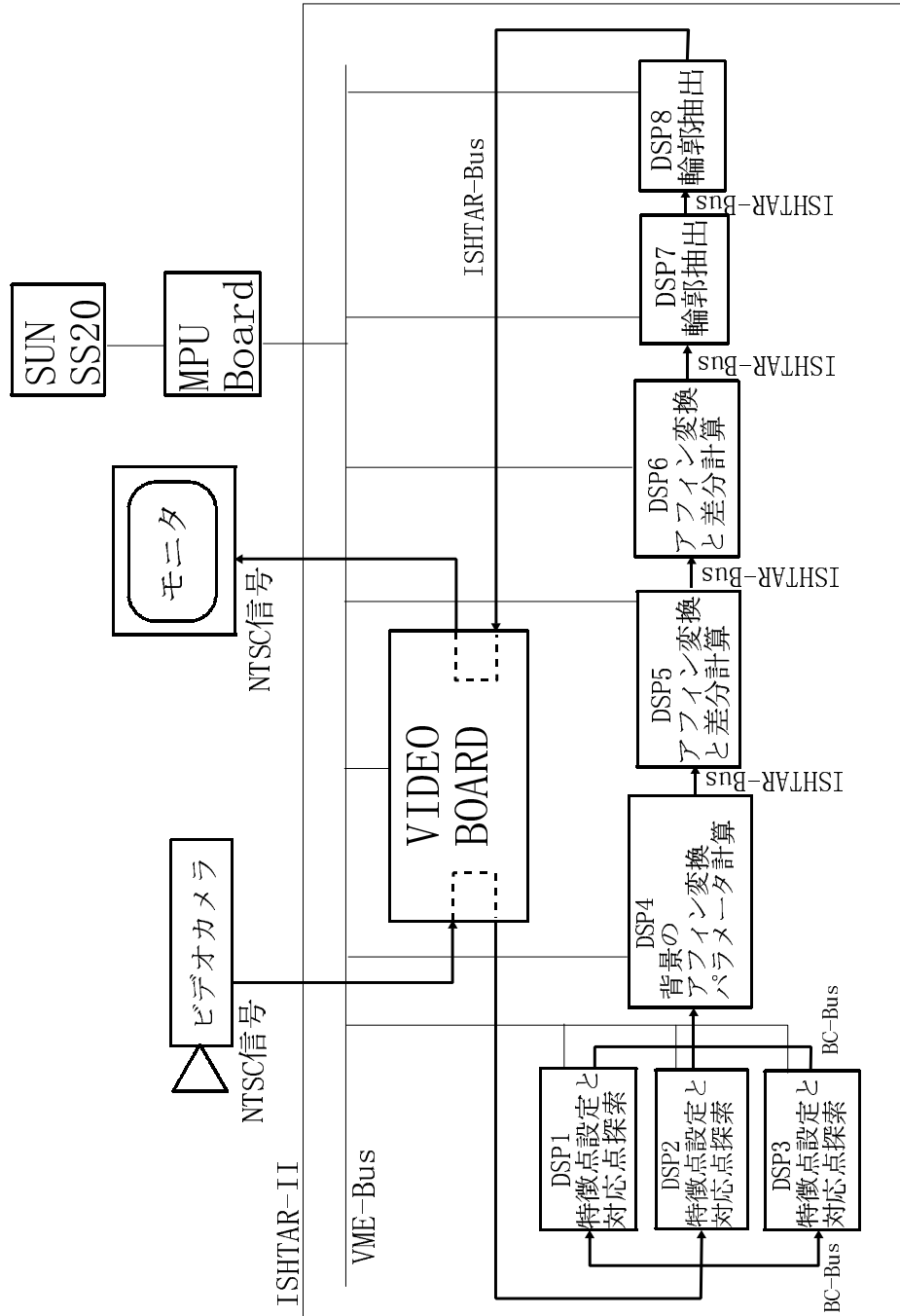


図 4 システム構成図

- DSP5 ～ DSP6

推定したパラメータをもとに前フレーム画像を変換し次フレーム画像との差分をとる.

- DSP7 ～ DSP8

差分画像に分裂統合型輪郭モデルを適用し, 結果を実画像へ重ねて表示画像を生成する.

本システムで秒 7.5 フレームで移動物体の抽出, 追跡を行っている. DSP7, DSP8 における輪郭モデルの収縮回数はそれぞれ 5 回, 4 回である. また, パイプライン処理の各段階で 1/7.5 秒ずつ遅れが生じるため, 画像の入力から出力までの遅れは 0.8 秒である.

続いて, 図 4 におけるボード間のデータの転送フォーマットについて以下に述べる. まず, ボード全体を通して画像データが最初に 32bit 整数形式で送られる. 画像データ以外のデータは画像データに続いて以下に述べる形式で送られる. DSP1 ～ DSP3 で算出されたオプティカルフローデータは次のフォーマットは DSP4 へ送られる.

1byte 目	特徴点の X 座標
2byte 目	特徴点の Y 座標
3byte 目	オプティカルフローの X 成分+128
4byte 目	オプティカルフローの Y 成分+128

表 1 オプティカルフローの転送フォーマット

本 DSP システムでは付録 Aにあるように, 並列処理の際には BC バスと呼ばれるバスを利用する. BC バスは 32bit 幅であるが, 8bit ずつ 4 本のバスとして利用することが可能となっている. この BC バスを利用して DSP1 ～ DSP3 は図 5 のフォーマットで DSP4 へデータの転送を行う.

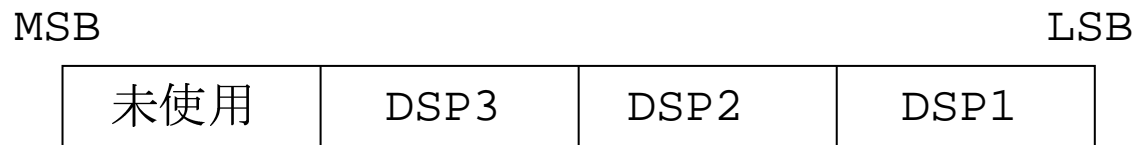


図 5 各 DSP の BC バス利用方法

DSP4は推定したアフィン変換パラメータを $a_{11}, a_{12}, b_1, a_{21}, a_{22}, b_2$ の順に 32bit 浮動小数点形式で DSP5 へ送る。DSP5 ~ DSP7 は画像データのみが送られる。DSP7 から DSP8 へは輪郭モデルの数, 各輪郭モデルの制御点数, 各制御点の座標が 32bit 整数形式で送られる。

5. 実験と結果

本システムの有効性を示すため、第 4.4 節に示したシステムを利用して実験を行った。

5.1 実験 1. 実画像による実験 - 移動物体が 1 つの場合

実験画像として人物が画面に向かって左から右へ歩いているのをカメラをパンしつつ撮像した動画像を利用した。まず、オプティカルフロー算出および背景位置合わせの有効性を検証するためにシステムの各段階での処理結果を表示した。図 6 に結果を示す。図 6(c) から、背景上に乗っている正しく算出されたオプティカルフローを利用して変換パラメータを推定していることがわかる。さらに、図 6(f) からわかるように正しく背景位置合わせが行われ、移動している人物のみが差分として抽出されていることがわかる。図 6(f) の画像上部のノイズは画像全体の背景とみなせる草原部と比較して遠距離にある柵によるものであり、また、画面左部のノイズは背景部分と比較して近距離にある樹によるものである。これらはオプティカルフローの算出精度やアフィン変換の近似精度などの複合による背景の見かけの動きの近似誤差によるものである。

次に利用した動画像の途中の連続する 3 フレームについての入力画像とそれを本システムで処理した結果を図 7(a) ~ (c) に示す。図 7(b) の差分の結果から、各フレームではほぼ正しい背景位置合わせが行われていることがわかる。また、図 7(b) 中央および右の画像においては駐輪場に相当する部分に若干ノイズが乗っているが、これはこの部分が全体の背景である道路部分と比較して近距離にあるためアフィン変換による近似が良好に行えなかったためと考えられる。しかしながら図 7(c) から分かるように、これらのノイズは最終的な輪郭モデルによる移動物体抽出には影響しておらず、各フレームにおいて正しく移動物体のみが抽出、追跡されていることがわかる。

実験において利用したパラメータを表 2 に示す。



(a) 前フレーム画像



(d) 次フレーム画像



(b) 抽出された特徴点とそのフロー



(e) 前フレームの変換画像



(c) 変換に使用した3点とそのフロー



(f) 差分画像

図 6 背景位置合わせの処理の様子



(a) 原画像



(b) 背景位置合わせ後のフレーム間差分画像



(c) 輪郭モデルによる移動物体抽出・追跡結果

図 7 輪郭モデルによる抽出結果

画像サイズ	160 × 120
テンプレートサイズ	$T_s = 5$
探索領域	21 × 21
抽出した特徴点	$N_f = 18$
試行打ち切り回数	$N_{max} = 80$
滑らかさの係数	$\alpha = \beta = 1.0$
エッジ係数	$\gamma = 3.0$
面積項の重み	$w_{area} = 1.0$

表 2 実験において利用したパラメータ

5.2 実験 2. 実画像による実験 - 移動物体が複数の場合

最後に、本研究で用いた分裂統合型輪郭モデルでは同時に複数の物体を抽出することが可能であるので、動画像中に複数の移動物体がある場合についても実験を行った。図 8 は 2 人の人物が同一方向へ歩いており、一方が他方を追い越していく様子をカメラをパンしつつ撮像した画像を本システムで処理した結果である。利用したパラメータは実験 2. と同じく表 2 のものである。図 8 に示されるように、図 8(a) においては 2 つの移動物体は別の移動物体として抽出され、図 8(b) において 2 つの移動物体を抽出している輪郭モデルが衝突し、図 8(c) においてまとまった 1 つの移動物体として統合され抽出されている。さらに図 8(d) で移動物体間の距離が離れるに従って輪郭モデルは広がり、移動物体の間に侵入していく。そして図 8(e) において輪郭モデルは自己交差を起こし、この部分で分裂することにより最終的に図 8(f) において 2 つの移動物体として再抽出されていることがわかる。

5.3 考察

以上に示されるように本手法は移動カメラ画像から移動物体を抽出、追跡する有効な手法であると言える。本システムの特徴は次のように言える。



(a) 第 0 フレーム



(d) 第 54 フレーム



(b) 第 6 フレーム



(e) 第 94 フレーム



(c) 第 7 フレーム



(f) 第 97 フレーム

図 8 移動物体が 2 つの場合の輪郭モデルによる追跡結果

1. カメラが運動している場合に移動物体輪郭を秒 7.5 フレームで抽出, 追跡できる.
 2. 図 7(b) の樹や柵に示されるような背景の動きが部分的にアフィン変換で近似できない場合においても, そのような領域が少ない限り最終的な抽出, 追跡には影響しない
 3. 同時に複数の移動物体輪郭を抽出, 追跡できる.
1. は背景位置合わせの精度を若干犠牲にすることで高速化を図ったためである. しかし, 2. で挙げたように最終的な移動体の抽出および追跡には影響を与えていないことがわかる. すなわち, 背景位置合わせの精度を落して輪郭モデルを用いると高い安定性を保ったまま高速化が望めると言える. また分裂統合型輪郭モデルを用いたことでさらに 3. に見るような複数移動物体への拡張も行うことができた. 現状では人間の歩行に伴って足の部分に抽出洩れが発生しやすい. これは歩行時に人間の接地している片足の移動量が少ないため, 差分によって抽出される領域が少なく, 分裂統合型輪郭モデルによる抽出において足の部分がノイズとして分離・除去されてしまうためであり, 移動量を基準に抽出を行う限り回避は困難であるといえる. また, オプティカルフローの算出にあたり特徴点をランダムに選択しているため, 同じ入力画像であっても処理の開始のタイミングなどにより複数移動物体の分離抽出などの失敗が発生することがある. オプティカルフロー算出に利用する選択点の数が少ない場合画像上で特定の点を常に利用するよりもランダムに輝度分散の大きい点を選択する方が精度的に有利である. このことからより安定な複数移動物体抽出のためには処理のさらなる高速化によって選択点数を増加することが有効な解決策であると言える.

6. おわりに

本研究では移動カメラ画像から実時間で移動物体を抽出する手法について検討し、その DSP 上への実装を試みた。本手法は背景位置合わせと分裂統合型輪郭モデルの利用により高速かつ安定な移動カメラ画像からの移動物体抽出手法である。また、動画像処理用 DSP ボードを複数利用したシステムへの実装により実際に実時間で複数の移動物体輪郭を抽出・追跡できることを確認した。本システムは以下のような特徴を持つ。

1. 移動物体輪郭を秒 7.5 フレームで抽出, 追跡できる
2. 複数の移動物体輪郭を同時に抽出, 追跡できる
3. 背景の動きにアフィン変換で近似できない領域が存在してもそれが少なければ移動物体の抽出, 追跡に影響しない

このような特徴から、本手法は突然現れる障害物などを認識して回避等が可能なロボットや、歩行者を検知して安全に目的地を目指す自動車への応用などが可能である。

今後の課題としては次のようなものが挙げられる。

1. 現在メモリ量による問題で不可能である 3 つ以上の移動物体の抽出, 追跡
2. 画面外から新たに侵入する移動物体の検出および追跡
3. 移動物体が複数の場合にすれ違い前後での移動物体の対応付け
4. さらなる高速化

まず、3 つ以上の移動物体の抽出および追跡であるが、これは輪郭モデルの実装における制限である。現在、輪郭モデルの実装において非効率なメモリ利用が残っており、この解消により 3 つもしくは 4 つの移動物体の抽出が可能である。また、より搭載メモリ量の多い DSP ボードが利用可能となれば、さらに多くの移動物体を抽出、追跡が可能である。

次に、画面外から新たに侵入する移動物体の検出および追跡であるが、これは画面の外枠に常時侵入検出用の輪郭モデルを置くことで解決できる。

さらに、移動物体が複数の場合の対応付けであるが、これは輪郭モデル内部の特徴量(輝度分散)等を保持しておき、すれ違い前後で対応させることで可能である。最後のさらなる高速化であるが、

- より多くの DSP ボードを利用して並列化を進める。
- より高速化が望めるアルゴリズムの検討および実装
- より高速な DSP を利用する

などにより可能である。3番目のより高速な DSP の利用であるが、すでに我々が利用した DSP より高速なものは市販されており、このような高速な DSP を利用して、本研究で利用した DSP システムと同様のシステムを構築することで高速化可能である。また、現状では Intel 社の x86 系列や DEC 社の Alpha 系列などの CPU においても今回用いた DSP 同様の高速な浮動小数点演算や多命令並列実行などの機能を備えてきており、このような CPU の利用によっても同様の高速処理を得ることができるようになってきている。これらの CPU はより優れた開発環境を持ち、一般に入手しやすいことから本研究で利用した手法を容易に実現可能であり、自律移動ロボットや自動車の自動運転の実用化へ向けて画像センサを利用することも困難でなくなるといえる。

謝辞

本研究を進めるにあたり、終始暖かく御指導を頂きましたソフトウェア基礎講座横矢 直和教授 に厚く御礼申し上げます。また、本学での研究におけるあらゆる面で多大な御助言、御協力を賜りましたことを深く感謝致します。

本論文をまとめるにあたり、有益な御助言を頂きました像情報処理学講座千原 國宏教授、並びにソフトウェア基礎講座 竹村 治雄助教授に深く感謝致します。

本研究への助言や研究環境の維持整備に御尽力頂きましたソフトウェア基礎講座 岩佐 英彦助手、山澤 一誠助手に深く感謝致します。

本研究を通して有益な御助言を頂きました松下電器産業 (株) 荒木昭一氏に深く感謝致します。

また、日々の研究室の活動を支えて下さったソフトウェア基礎講座事務補佐員 福永 博美女史、並びにソフトウェア基礎講座元事務補佐員村上 和代女史に感謝します。

最後に、本研究の多様な局面においておしめない御協力を頂きましたソフトウェア基礎講座諸氏に感謝します。

参考文献

- [1] M.Kass, A.Witkin and D.Terzopoulos: “SNAKES:Active contour models”, Proceedings of 1st International Conference on Computer Vision, pp.259-268, 1987.
- [2] 荒木 昭一, 横矢 直和, 岩佐 英彦, 竹村 治雄: “交差判定により分裂・統合する輪郭モデルによる複数移動物体の追跡”, 電子情報通信学会技術報告, PRMU96-21, 1996.
- [3] 坂上 勝彦, 山本 和彦: “動的な網のモデル Active Net とその領域抽出への応用”, テレビジョン学会誌, Vol.45, No.10, pp.1155-1163, 1993.
- [4] 吉野 和芳, 守田 了, 川嶋 稔夫, 青木 由直: “アクティブネットの分裂による複数物体の追跡”, 画像の認識 理解シンポジウム講演論文集 (MIRU'92), pp.145-152, 1992.
- [5] J.Segen and S.Pingali: “A Camera-Based System for Tracking People in Real Time”, Proceedings of 13th International Conference on Pattern Recognition, pp.63-67, 1996.
- [6] I.J.Cox and S.L.Hingorani: “An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking”, IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol.18, No.2, pp.138-150, Feb.1996.
- [7] S. Yamamoto, Y. Mae, Y. Shirai and J. Miura: “Realtime Multiple Object Tracking Based on Optical Flows”, Proceedings of International Conference on Robotics and Automation, pp.2328–2333, 1995.
- [8] 太田 直哉: “信頼性情報を持ったオプティカルフローからの形状復元とその移動物体検出への応用”, 電子情報通信学会論文誌 (D-II), Vol.J76-D-II, No.8 pp.1562–1571, Aug.1993.

- [9] 二宮 芳樹, 太田 充彦: “オプティカルフローによる移動物体の検知”, 電子情報通信学会技術報告, PRMU97-28, 1997.
- [10] 八木 康史, 浅田 稔, 辻 三郎: “移動カメラで撮像した環境内の運動物体の検出”, 電子情報通信学会論文誌 (D), Vol.J69-D, No.5, pp.794-801, May.1986.
- [11] 寺久保 敏, 横矢 直和, 岩佐 英彦, 竹村 治雄: “2 段階アフィン変換を利用した移動カメラ画像からの動物体の抽出”, 電子情報通信学会技術報告, PRU95-181, 1995.
- [12] 朝岡 忠, 横矢 直和, 竹村 治雄, 山澤 一誠: “ロバスト統計を用いた背景位置合わせに基づく移動カメラ画像からの移動物体抽出”, 電子情報通信学会技術報告, PRMU96-145, 1996.
- [13] 松岡 隆志, 荒木 昭一, 山澤 一誠, 竹村 治雄, 横矢 直和: “移動カメラ画像からの実時間移動体抽出の DSP による実装の検討”, 情報処理第 55 回全大講演論文集, No.2K-1, Sep.1997.
- [14] P.J.Rousseeuw and A.M.Leroy: Robust Regression and Outlier Detection, John Wiley & Sons, NewYork, 1987.
- [15] 此島 真喜子, 瀬川 英吾, 塩原 守人, 佐々木 繁: “動画像処理システム ISHTAR による前方車間距離計測”, 電子情報通信学会技術報告, PRU95-48, 1995.
- [16] M. Shiohara, H.Egawa, S. Sasaki: “Real-Time Optical Flow Processor ISHTAR”, Proceedings of 1st Asian Conference on Computer Vision, pp.790-793, 1993.
- [17] T.McInerney and D.Terzopoulos: “Topologically Adaptable Snakes”, Proceedings of 5th International Conference on Computer Vision, pp.840-845, 1995.
- [18] B.K.P. Horn and B. Schunck: “Determining optical flow”, Artificial Intelligence, Vol.17, pp.185-203, 1981.

- [19] V.Markandey and B.E.Flinchbaugh “Multispectral Constraints for Optical Flow Computation” Proceedings of 3rd International Conference on Computer Vision, pp.38-41, 1990.
- [20] D.J.Williams and M.Shah: “A fast algorithm for active contours”, Proceedings of 3rd International Conference on Computer Vision, pp.592-595, 1990.
- [21] TMS320C40 User’s Guide, TEXAS INSTRUMENTS, 1993.
- [22] 動画像処理システム ISHTAR-II ユーザーズマニュアル, 株式会社富士通研究所 メディア処理研究部, 1995.

付録

A. 動画処理用 DSP システム

以下, 本研究で利用した DSP システムについて説明する [21, 22].
まず, DSP について, 次に DSP を用いて構成された DSP ボードについて, 最後に, DSP システムへの画像の入出力を行う VIDEO ボードについて述べる.

A.1 DSP

本研究で利用した DSP ボードで用いられている DSP は TI 社製の TMS320C40 (50MHz) (以下 C40 と表記) である. 一般に DSP は浮動小数点演算を高速に行うことが可能であり, C40 の場合浮動小数点の加減乗除を 1 クロックで行うことが可能である. また, 浮動小数点の扱いは表 3 のようになっている.

	内部レジスタ	内部レジスタ以外 (メモリ, 外部バス出力等)
実数部	32bit(符号 bit 込み)	24bit(符号 bit 込み)
仮数部	8bit	8bit

表 3 C40 での浮動小数点の取扱

これらの浮動小数点データは内部レジスタへのロードおよび内部レジスタからのストアの際に自動的に型変換される. また, 乗算と加算の同時実行やレジスタ-メモリ間のデータ転送を 2 つ同時に実行可能となっており, 実装時にこれらを考慮することでより高速な処理を得ることが可能となっている.

次に C40 の特徴について説明する. C40 には高速処理のために次のようなハードウェアが実装されている.

- **インストラクションキャッシュ**

これは C40 がいったん実行した命令を C40 内部に蓄えておくためのキャッ

シユである。これにより短いループの実行などの場合、命令の取り込みのための外部バスへのアクセスが最初の 1 回のみとなり高速な処理の実現が可能である。

- **内部 RAM**

C40 内部に 1024word×2 用意された高速なメモリである。2 枚それぞれに独立してバスが割り当てられているために相互に干渉することがなく、C40 が一方のデータを処理している間に次に述べる DMA を利用して他方へデータを転送することで外部バスへのアクセスを減らし、高速動作を可能とする。

- **DMA**

C40 内部に 6 チャンネル用意された C40 とは独立に動作するメモリ-メモリ間データ転送ユニットである。C40 でのデータ処理中に新しい画像データなどを高速なメモリに転送することで C40 が低速なバスへアクセスする回数を減らし、高速動作を可能とする。

- **コミュニケーションポート**

別の C40 とメモリ等を介さずに直結でき、C40 の持つその他のバスと独立に動作するポート。2 ポート搭載されており、データの入力と出力が行える。

以上の機構によって各 C40 は高速な動作を可能としている。

A.2 DSP ボード

図 9 に本研究で使用した DSP ボードの構成図を示す。以下で各部分について説明する。

まず、各 DSP には 2 種類の外部バスがある。それぞれ次のようなものである。

- **グローバルバス**

このバスにはフレームメモリのみが接続されている。前段のボードからの入力は一時的にグローバルメモリに保持されるので一般にデータの入力に使用される

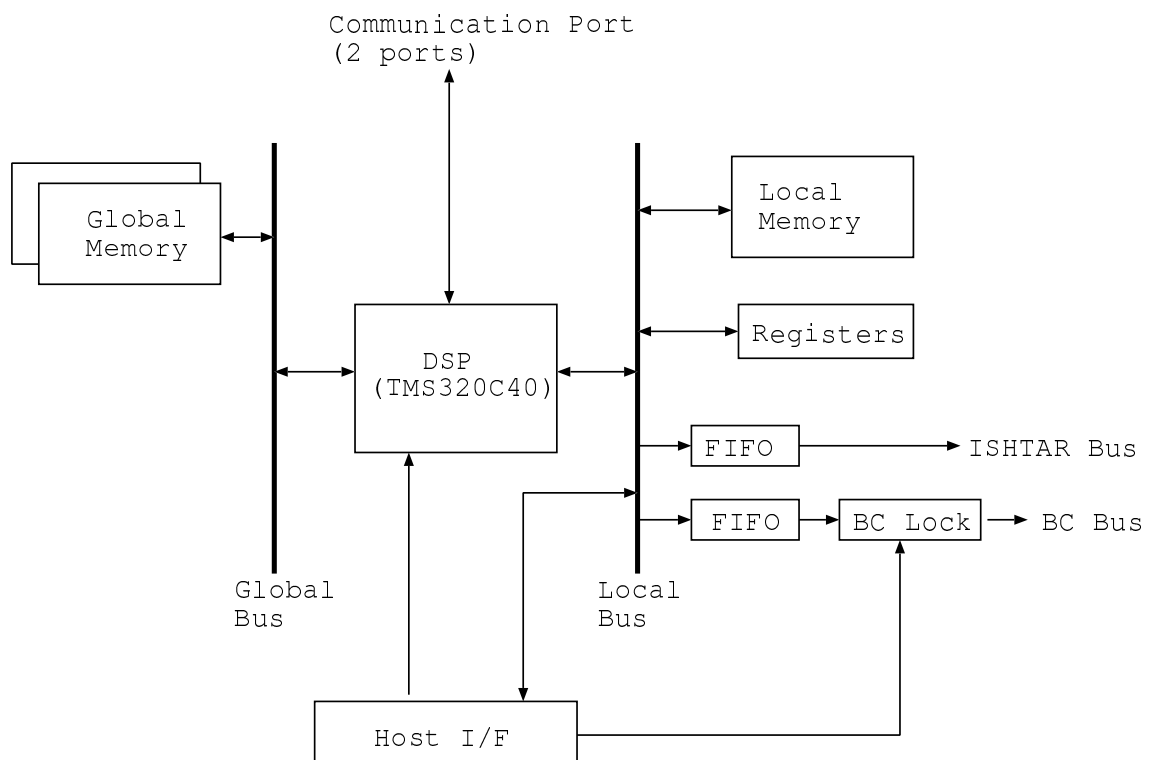


図 9 DSP ボードの構成

- ローカルバス

このバスにはローカルメモリ、各種外部レジスタ、次段のボードへ転送するデータを保持する FIFO が接続されている。データの処理や、結果の次段への転送に使用される。

次に、各 DSP ボードに搭載されているメモリについて説明する。まず、DSP から見たメモリマップを図 10 に示す。図 10 に示すように、各 DSP ボードには 256Kword のフレームメモリが 2 枚と 256Kword のローカルメモリが 1 枚搭載されている。また DSP 内部に搭載された内部 RAM も一般メモリと同様にメモリマップ上に載せられている。各メモリの特徴を以下に示す。

- フレームメモリ

前段からのデータが保持されるメモリ。2 枚存在し、1 枚は前段からのデータ入力、もう 1 枚は DSP からのデータアクセスに使用される。このダブルバッファリングの機能によって各 DSP は前段からのデータ入力を意識することなくデータの処理が可能となっている。ただし、DSP ボード上からのデータアクセスには 1wait 入るため、高速化のためには処理するデータをこのフレームメモリから次に述べるローカルメモリへ転送して利用する必要がある。

- ローカルメモリ

DSP での処理を行う場合に実際に利用するメモリ。DSP の実行するプログラムや処理に利用するデータが保持される。wait は 0 であり、フレームメモリより高速であるが、先に述べた内部 RAM より若干低速である。容量が大きいため内部 RAM に入りきらない画像データなどはローカルメモリ上で処理を行う。

最後に DSP ボード間でデータを転送する際に利用する外部バスについて説明する。各 DSP ボードには 2 種類の外部バスが搭載されており次のような特徴を持つ。

- ISHTAR Bus

ボード間を 1 対 1 で接続し、32bit のデータの転送を可能にするバス。32bit

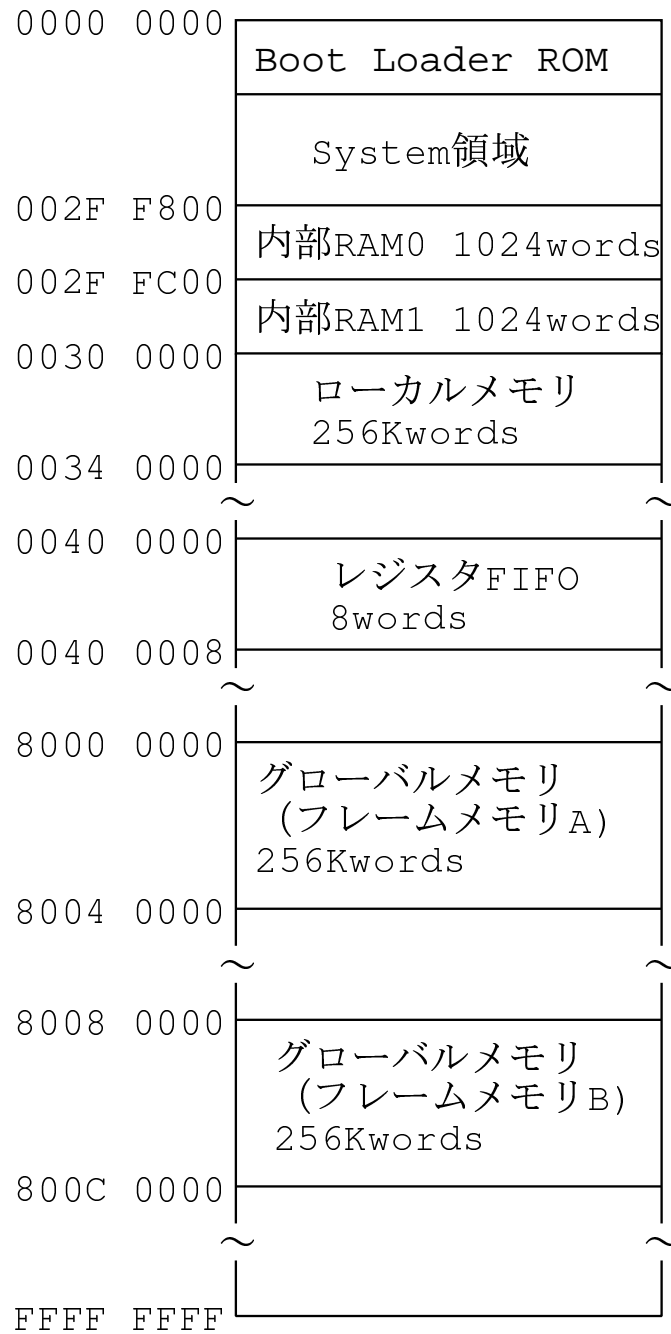


図 10 DSP ボードのメモリマップ

をまとめて扱う必要があり, 1 方向のみのデータ転送が可能である. パイプライン接続の際に利用する.

- **BC Bus**

1 対多, 多対多の接続を可能にするバス. バス幅は 32bit であり, 各ボードはこの 32bit を 8bit 毎に分割して, それぞれ入出力することが可能である. 8bit 毎に利用可能であるため, 並列処理の際に利用する. 並列処理を行う場合, 同時に複数の DSP ボードから BC バスへの出力が行われる可能性がある. この時各 8bit 毎にどの DSP ボードからデータを出力するかは制御はホストコンピュータで行う.

それぞれのボードではこれらのバスからの入力をフレームメモリに蓄積する. この際, 先に説明したダブルバッファリングにより DSP の行っている処理には影響を及ぼさない. また, 各 DSP ボードがどのバスから入力を行うかは各 DSP ボード上に搭載された外部レジスタによって選択する. 逆に DSP がこれらのボードへデータを出力する場合いったんローカルバスに接続された FIFO へデータを出力し, 実際の各外部バスへの出力は FIFO を通して自動的に独立して行われるため同期などを意識する必要はない. このように DSP ボードにおいてはデータ転送の際のオーバーヘッドを軽減し, 結果として高速な処理を可能とする構成となっている.

A.3 VIDEO ボード

本研究で利用した DSP ボードへの画像の入出力は VIDEO ボードによって行う. VIDEO ボードの機能を以下に示す.

- NTSC 信号を最高秒 30 フレームでデジタル画像化する
- 1 フレーム時間の中に奇数, 偶数フィールドの順に次段のボードへ出力
 - 各フィールドの画像サイズは 640×240
 - 1 ピクセルのサイズは 32bit

- システム全体の同期のための同期信号の出力
- DSP ボードの処理結果を NTSC 信号へ変換する

ただし、NTSC 信号からデジタル画像への変換は色情報を保存して行えるが、DSP ボードの処理結果を NTSC 信号へ変換する場合にはモノクロ NTSC 信号への変換しか行うことができない。また、VIDEO ボードは次段への出力を間引いて行うことが可能であり、この機能を利用してシステム全体の処理速度を変更することが可能である。