

## オプティカルフロー演算ハードウェアと 移動体追跡への応用

平位隆史      笹川耕一      黒田伸一

三菱電機(株)産業システム研究所

本報告ではオプティカルフロー演算を実時間で実行するためのH/Wの構成法について述べる。オプティカルフローは物体の移動ベクトルを抽出するための有力な手段であるが、繰り返し演算と浮動小数点演算を含むため計算量が非常に多く、実時間でのオプティカルフローの抽出は困難であった。筆者らは上記の問題を解決するため、オプティカルフローの繰り返し演算がパイプライン的に処理できることを示し、さらに浮動小数点を用いない整数型の演算で実行できることを示す。その結果、市販の画像処理用LSIとEPLD(Erasable Programmable Logic Device)によりコンパクトなH/Wを開発したので報告する。本H/Wによる移動体追跡への応用例も示す。

## Hardware of Optical Flow Calculation and Its Application to Tracking of Moving Object

Takashi Hirai, Kouichi Sasakawa and Shin-ichi Kuroda

Industrial Electronics and Systems Lab.

Mitsubishi Electric Corp.

8-1-1, Tsukaguchihonmachi, Amagasaki, Hyogo 661, JAPAN

This paper deals with the architecture to calculate the optical flow in real time. Although the optical flow is the powerful method to extract the vector of moving objects, the computational effort that is caused by the iteration and floating point operations prevents from real time processing. In this paper, we show that the optical flow equations can be transformed into the forms those are suitable for pipeline processing. Moreover we show that those equations can be carried out by integer operations. Examples of application to tracking of moving objects are also shown.

## 1 はじめに

本報告ではオプティカルフロー演算の H/W 化とその応用例について述べる。我々は [2], [3] において Horn and Schunck の勾配法によるオプティカルフロー [1] を用いた微小移動物体検出法を提案しているが、オプティカルフローの演算に時間がかかるのが欠点であり、実時間でオプティカルフロー演算を行える H/W の実現が応用上必須である。

オプティカルフローの演算を H/W で実現しようとするときの問題点は次の 2 点である。まず、オプティカルフローの演算には繰り返し演算が必要であり、全画面にわたってオプティカルフローを計算するときに繰り返しの回数に比例した計算時間がかかるという問題がある。次に繰り返し演算の途中に浮動小数点の積和演算と除算が含まれているため、そのまま H/W 化しようすると DSP などの浮動小数点演算用の LSI を用いても規模が相当大きくなるという問題がある。

本報告ではオプティカルフロー方程式を変形することにより、オプティカルフローの演算がパイプライン処理で行えることを示し、第 1 の問題を解決した。パイプライン処理は画像処理装置に採用されている高速処理手法の一つであり、筆者らは可変パイプライン構造を持った高速画像処理装置を開発済である [4]。次にデータ型について検討した結果、内部データは 8bit 程度の整数であれば出力の精度には影響しないことが確認できた。内部データが 8bit 程度であれば除算は ROM を用いて高速に実行することができる。さらに出力データも 8bit 程度であればオプティカルフローの強度、方向を算出するのに十分な精度である。これにより、第 2 の問題である H/W 規模の問題も解決することができた。

以上の考察の結果、市販の画像処理用 LSI と EPLD(Erasable Programmable Logic Device) により、繰り返し演算のための係数演算部を VME ダブルハイトボード 1 枚、その係数を用いて行う繰り返し演算部を繰り返し 1 回につき 1 枚のボードで実現することができ、リアルタイムのオプティ

カルフロー抽出が可能になったので報告する。また、本ボードを用いた移動体追跡についても簡単に述べる。

## 2 オプティカルフロー

本章ではオプティカルフローの原理と、オプティカルフローを用いた微小移動物体検出法 [2], [3] について簡単に述べる。

### 2.1 オプティカルフローの原理

オプティカルフローは観測者と物体の間の相対的な運動によって生じる画像上の見かけの速度分布を表すものであり、Horn ら [1] により次のように基本方程式が導かれた。いま時刻  $t$  において画像上のある画素  $(x, y)$  における輝度値を  $I(x, y, t)$  とし、第 1 の仮定として「物体の輝度は時間的に不変である」とすると次式が成立する。

$$I_x u + I_y v + I_t = 0 \quad (1)$$

ここで、 $u, v$  は  $x$  方向、 $y$  方向の速度成分、 $I_x, I_y$  は  $x$  方向、 $y$  方向の空間輝度勾配、 $I_t$  は時間輝度勾配である。次に第 2 の仮定として「速度の変化は滑らかである」とし、次の誤差関数

$$E^2 = \iint (E_b^2 + \alpha^2 E_c^2) dx dy \quad (2)$$

を最小にする  $u, v$  がオプティカルフローである。 $E_b$  は第 1 の仮定による誤差、 $E_c$  は第 2 の仮定による誤差であり、 $\alpha$  は適当な重み係数である。観測値に雑音が多く信頼できるときには  $\alpha$  は小さくし、逆に観測値が信頼できない場合には第 2 の仮定による平滑化効果を上げるために  $\alpha$  は大きくする。(2) 式を最小にする  $u, v$  は次の繰り返し式で与えられる。

$$u^{(n+1)} = \bar{u}^{(n)} - \frac{I_x(I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t)}{\alpha^2 + I_x^2 + I_y^2} \quad (3)$$

$$v^{(n+1)} = \bar{v}^{(n)} - \frac{I_y(I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t)}{\alpha^2 + I_x^2 + I_y^2} \quad (4)$$

ここで、 $\bar{u}^{(n)}, \bar{v}^{(n)}$  は  $u^{(n)}, v^{(n)}$  の 8 近傍の平均、 $n$  は繰り返し回数である。

## 2.2 微小移動物体検出法

我々は [2], [3] において、オプティカルフローを用いた微小移動物体検出法を提案している。本手法は時間的に異なる 2 枚の画像上に微小な物体が存在するときに、物体の存在位置の画素の 8 近傍のフローが図 1 に示すようなパターンを持つことに着目し、微小移動物体を検出する方法である。

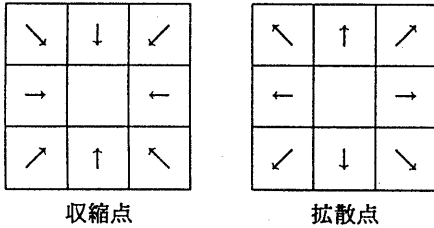


図 1: 収縮点と拡散点

ここで収縮点は時刻  $t-1$  における物体の存在位置、拡散点は時刻  $t$  における物体の存在位置である。このようなオプティカルフローの性質は物体がカメラの光軸に沿って遠ざかる場合や接近する場合、すなわち 2 次元画像上では物体が縮小あるいは拡大する場合の極限状態のフローの振る舞いであると考えることができる。また物体の移動量が少なく 2 フレーム間でお互いに重なりを持って移動している場合には物体の存在位置の 8 近傍には図 2 のようなパターンが現れる。この場合には移動体が存在する可能性を判定することができる。

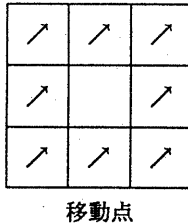


図 2: 移動点のフロー

しかし (3), (4) 式からオプティカルフローを計算するためには繰り返しの回数に比例した計算時間が必要である。また浮動小数点の積和演算や除

算が含まれるため回路規模の面から H/W 化が困難であった。我々はこれらの問題を処理のパイプライン化と演算の整数化によって解決し、コンパクトな H/W を開発したので報告する。

## 3 H/W 化の概要

本章ではオプティカルフロー方程式がパイプライン処理に適した形に変形できることを示すとともに、内部演算が整数型のデータで行えることを示す。

### 3.1 パイプライン処理

(3), (4) 式において、輝度勾配  $I_x, I_y, I_t$  はある画素とその空間隣接画素および時間隣接画素の  $2 \times 2 \times 2$  の 8 画素の和と差であるので、時間的に異なる 2 枚の画像がラスタースキャンで入力されると、近傍画素演算で計算される。したがって輝度勾配の計算はパイプライン処理が行える。輝度勾配が求まると (3), (4) 式からオプティカルフローが計算できるが、(3), (4) 式を変形すると、

$$u^{(n+1)} = \frac{(\alpha^2 + I_y^2)\bar{u}^{(n)} - I_x I_y \bar{v}^{(n)} - I_x I_t}{\alpha^2 + I_x^2 + I_y^2} \quad (5)$$

$$v^{(n+1)} = \frac{(\alpha^2 + I_x^2)\bar{v}^{(n)} - I_x I_y \bar{u}^{(n)} - I_y I_t}{\alpha^2 + I_x^2 + I_y^2} \quad (6)$$

となる。したがって

$$a = \frac{\alpha^2 + I_y^2}{\alpha^2 + I_x^2 + I_y^2}, \quad b = \frac{I_x I_y}{\alpha^2 + I_x^2 + I_y^2}, \quad (7)$$

$$c = \frac{\alpha^2 + I_x^2}{\alpha^2 + I_x^2 + I_y^2}, \quad d = \frac{I_x I_t}{\alpha^2 + I_x^2 + I_y^2}, \quad (8)$$

$$e = \frac{I_y I_t}{\alpha^2 + I_x^2 + I_y^2} \quad (9)$$

とおくと

$$\begin{cases} u^{(n+1)} = a\bar{u}^{(n)} - b\bar{v}^{(n)} - d \\ v^{(n+1)} = c\bar{v}^{(n)} - b\bar{u}^{(n)} - e \end{cases} \quad (10)$$

となる。すなわち、 $u^{(n+1)}, v^{(n+1)}$ は係数 $a, b, c, d, e$ と $\bar{u}^{(n)}, \bar{v}^{(n)}$ との積和演算で計算されることがわかる。係数 $a, b, c, d, e$ は定数 $\alpha$ と輝度勾配 $I_x, I_y, I_t$ から計算されるので、2枚の画像が入力され各画素毎に輝度勾配を計算した時点で決定され、オプティカルフローの繰り返し演算とは無関係である。また $\bar{u}^{(n)}, \bar{v}^{(n)}$ は $u^{(n)}, v^{(n)}$ の8近傍の平均値であるので、第 $n$ 回目の繰り返しによるオプティカルフローの近傍演算となる。したがってこの部分もパイプラインで処理することができる。図3にパイプライン処理の概略図を示す。図よりオプティカルフローの演算は、画像が入力されたときに輝度勾配と係数を演算する部分と、平均値、積和演算をする部分に分けることができ、後者の部分を複数用いることにより繰り返しの回数を制御できる形に帰着できることがわかる。

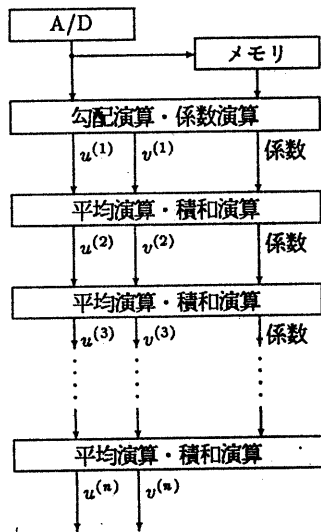


図3: オプティカルフローのパイプライン処理

### 3.2 演算の整数化

次に(7) - (9)式について考察する。明らかに $|a| < 1, |b| < 1, |c| < 1$ は成立する。また $d, e$ については輝度勾配の計算をする $2 \times 2 \times 2$ の8画素の値と定数 $\alpha$ との関係で1を越える場合もある

が、これは1枚の画像で隣接画素の輝度値が大幅に変わり、かつ2枚の画像で同じ空間座標の画素の輝度値が大幅に変わるようなときにしか起こらず、実画像ではほとんど起こり得ない組み合わせである。したがって $|d| < 1, |e| < 1$ としても実用上問題は無いと考えられる。

いま、以下のように(7) - (9)式の各値を $2^k$ 倍し、整数化する。

$$A = 2^k a, B = 2^k b, C = 2^k c, \quad (11)$$

$$D = 2^k d, E = 2^k e \quad (12)$$

$$U^{(n)} = 2^k u^{(n)}, V^{(n)} = 2^k v^{(n)} \quad (13)$$

このとき(10)式は次のようになる。

$$\begin{cases} U^{(n+1)} = \frac{1}{2^k} (A\bar{U}^{(n)} - B\bar{V}^{(n)}) - D \\ V^{(n+1)} = \frac{1}{2^k} (C\bar{V}^{(n)} - B\bar{U}^{(n)}) - E \end{cases} \quad (14)$$

ここで(7) - (9)式のビット数について考察する。まず入力画像は8bitの正数であるから輝度勾配 $I_x, I_y, I_t$ の取り得る範囲は-255から255となり符号付き9ビットの整数で表現できる。定数 $\alpha$ が8bitの整数で与えられるとすると符号を考慮して $\alpha^2 + I_x^2, \alpha^2 + I_y^2$ は17bit、 $\alpha^2 + I_x^2 + I_y^2$ は18bit、 $I_x \times I_y, I_x \times I_t, I_y \times I_t$ はそれぞれ17bitになる。しかし、(7) - (9)式では比が求まればよく、またそれぞれ8bit程度の精度があればよいことから分子、分母とも同程度の精度であればよい。したがって各画素毎に分子、分母が8bitになるようにシフトし、これらの分子、分母を入力アドレスとするROMに除算結果を $2^k$ 倍したものを書いておくことにより、内部のデータ量を減らすことができるとともに除算も高速に行うことができる。

次にデータシフトによる整数化に伴う誤差の影響について考察する。上記のように(7) - (9)式の分子、分母を各8bitになるようにシフトすると、定数 $\alpha$ に対して物体と背景の輝度差が非常に小さい場合、輝度勾配が反映されない。例えば $\alpha = 32(6\text{bit})$ としたとき、 $I_x^2 + I_y^2 \leq 7, I_x I_t \leq 7, I_y I_t \leq 7$ となるような輝度勾配に対して、(7) - (9)式の分

子、分母のうち $\alpha$ の含まれる項は2進数表示では10000000xxx、 $\alpha$ の含まれない項は00000000xxx、 $x = 0$  or  $1$ となる。これらに前述のシフトを行うと、桁落ちの結果 $a = c = 1, b = d = e = 0$ となり、繰り返しの初期値を $u^{(0)} = v^{(0)} = 0$ としたときオプティカルフローの値は常に0になる。したがってコントラストの低い画像では $\alpha$ の値を小さくする必要がある。また、適切な $\alpha$ を与えた場合でも、画面上で輝度勾配の小さい部分ほどデータシフトによる誤差の影響が大きくなる。しかし、輝度勾配の小さい部分は空間的に平坦な部分や時間的に変化しない部分、すなわち背景であり、そのような部分からはオプティカルフローは発生しない。また、背景が微妙に変化しても画像中の移動物体に比較するとオプティカルフローの強度は非常に小さくなる。このようなフローは誤差を多く含むが、最終的にある閾値より強度の小さいフローは0にする処理を行っているため、データシフトに伴う誤差の影響は無視してもよいと考えられる。

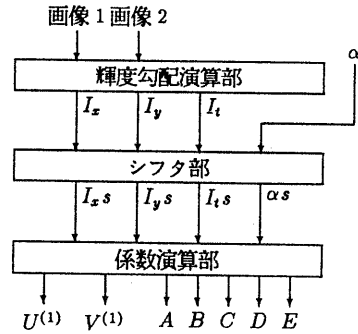


図 4: 係数演算ボードブロック図

## 4 各ボードの詳細

### 4.1 係数演算ボード

図 4 に係数演算ボードのブロック図を示す。本ボードは輝度勾配演算部、シフト部、係数演算部で構成される。また、図 5 は係数演算ボードの外観写真である。

#### 4.1.1 輝度勾配演算部

輝度勾配演算部では2系統の画像入力バスから入力される画像より、空間輝度勾配 $I_x, I_y$ および時間輝度勾配 $I_t$ を計算する。この回路はLSI ロジック社のビデオシフトレジスタとマルチビットフィルタにより構成した。

#### 4.1.2 シフト部

3.2節で述べたように、(7) - (9) 式は分子・分母とも8bitになるようにシフトしてから除算を行

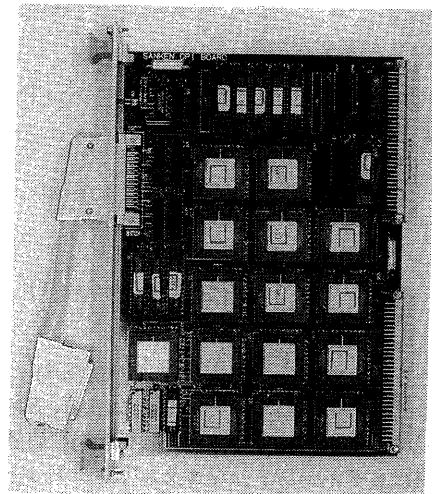


図 5: ボードの外観

が、分子・分母とも $\alpha, I_x, I_y, I_t$ の積和演算であるから、積和演算を行ってからビット数を低減することは、あらかじめ $\alpha, I_x, I_y, I_t$ のビット数を低減してから積和演算を行うことと同値である。そこでシフト部では $\alpha, I_x, I_y, I_t$ のうち、絶対値が最大のものが4bitになるようにシフト量を決定し、そのシフト量に応じて各値をシフトして出力する。

#### 4.1.3 係数演算部

係数演算部では前述のシフト回路の出力の積和演算により、(7) - (9) 式の分子分母を計算する。その後それぞれを入力アドレスとするROMに除算結果を書いておくことにより、除算を高速に行うことができる。

### 4.2 繰り返し演算ボード

図6は繰り返し演算ボードのブロック図である。本ボードは平均値演算部、係数遅延部、積和演算部、特徴点抽出部で構成される。

#### 4.2.1 平均値演算部

平均値演算部では次式のように $U^{(n)}, V^{(n)}$ の平均値を計算する。

$$\bar{U}^{(n)}(i, j) = \frac{1}{8} \sum_{(k, \ell) \in S} U^{(n)}(i - k, j - \ell) \quad (15)$$

$$\bar{V}^{(n)}(i, j) = \frac{1}{8} \sum_{(k, \ell) \in S} V^{(n)}(i - k, j - \ell) \quad (16)$$

ここで $U^{(n)}(i, j), V^{(n)}(i, j)$ は画素 $(i, j)$ におけるオプティカルフローの $x, y$ 方向の成分、 $S$ は画素 $(i, j)$ の8近傍である。本回路は輝度勾配演算部と同様にビデオシフトレジスタとマルチビットフィルタで構成される。

#### 4.2.2 係数遅延部

繰り返し演算ボードへは2系統の画像入力バスから $U^{(n)}, V^{(n)}$ 、係数バスからは5種類の係数A-Eが入力される。このうち $U^{(n)}, V^{(n)}$ は前述の平

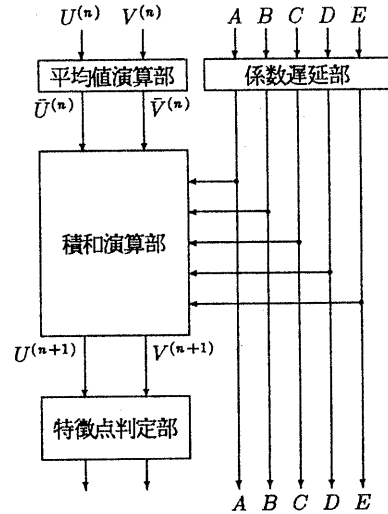


図6: 繰り返し演算ボードブロック図

均値演算部において近傍画素演算を行うので、平均値 $\bar{U}^{(n)}, \bar{V}^{(n)}$ は係数バスから入力された係数に対して遅延を含んでいる。このため、係数遅延部では係数を一旦ラインバッファに書き込み、平均値演算部での遅延に相当する遅延を係数に与えて読み出す。これにより次段の積和演算部への入力データの同期をとる。

#### 4.2.3 積和演算部

積和演算部はadderとmultiplierで構成し、(14)式により、 $U^{(n+1)}, V^{(n+1)}$ を計算する。

#### 4.2.4 特徴点抽出部

繰り返し演算ボードの最終段には特徴点抽出部を設ける。ここでいう特徴点とは2.2節で述べた微小移動物体に対する拡散点および移動点である。本ボードが繰り返しの途中であれば前述の積和演算回路の出力である $U^{(n+1)}, V^{(n+1)}$ をそのまま画像出力バスに送出し、次段の繰り返し演算の入力とする。本ボードが繰り返しの最終段であれば、オプティカルフローの強度、方向を算出し、方向

データから拡散点や移動点を検出し画像バスに送出する。このとき、強度に対する閾値を設定し、微小なフローは無視できるように構成した。またフローの強度、方向も画像バスに送出するためのセクタを設けたので、微小な移動体の検出だけでなく、通常の移動体追跡への応用も可能である。

## 5 移動体追跡への応用

移動体追跡への応用例としてに道路上に設置したカメラ画像からの車両抽出について述べる。図7は使用した連続画像中の1シーンである。図8には図7とその前フレームから抽出されたオプティカルフローの強度を示す。図は抽出されたフロー強度のある閾値以上のものだけを8bitに量子化し、256階調で表示している。この場合カメラは固定されているため、移動車両のみのフローが抽出される。図より車両のみが良好に抽出されていることがわかる。出力としてはフロー強度以外にフローの方向を8方向に量子化したものも得られるので、これらを利用し移動車両の切り出し、移動方向の判定等が可能である。

## 6 おわりに

オプティカルフローの実時間演算のためのH/W構成法と、応用例として固定カメラの画像からの車両抽出について述べた。他の応用としては車載カメラによる前走車追尾等も考えられるが、この場合にはカメラぶれによる背景のフローの除去や、車両切り出し法等の後処理を検討する必要がある、今後の課題である。

## 参考文献

- [1] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.
- [2] 平位, 笹川, 黒田, "オプティカルフローを用いた微小移動物体の検出," 情報処理学会研究報告, 90-CV-65, 1990.

- [3] T. Hirai, K. Sasakawa, S. Kuroda and S. Ikebata, "Detection of Small Moving Object by Optical Flow," *Proc. 11th ICPR*, Vol. II, pp. 474 - 478, 1992.
- [4] 黒田, 笹川, 池端, "可変パイプライン型高速画像処理装置 (PICIP) の開発," 画像電子学会第110回研究会資料, 98-01-07, pp. 37 - 42, 1989.

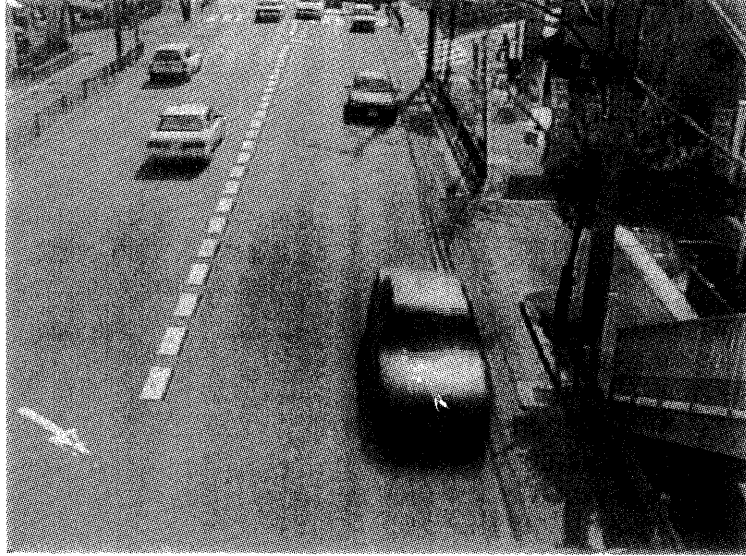


図 7: 原画像

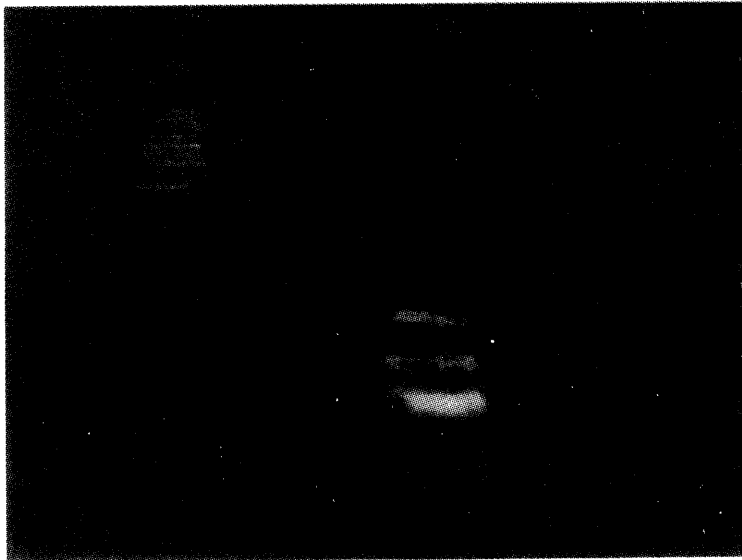


図 8: オプティカルフローの強度