高速物体用オプティカルフロー検出法

山際　勝也*・井上　哲理*・関　　靖夫*

Optical Flow Detection for High Speed Objects

Katsuya YAMAGIWA, Tetsuri INOUE and Yasuo SEKI

Abstract

In this paper, a method to detect optical flow of objects moving at high speed are proposed. Although several methods have been proposed, they have difficulty for high speed objects. In the method proposed, to cope with this difficulty, original images are processed with Gaussian filter before estimating optical flows by the gradient-based detecting method. The blurred images resulted from this filtering are expected to enable the gradient-based method to detect optical flow of high speed objects. This paper also describes a method to detect optical flows using only edge points on images to improve processing speed. Experiments are conducted to evaluate the ability of the proposed method to detect optical flow of high speed objects, and to examine effects of blurring parameter which corresponds to the width of Gaussian filter. Experimental results show that the proposed method is superior to the conventional gradient-based method in detecting optical flows of objects moving at high speed.

1. はじめに

本研究では画像上で高速に移動する物体のオプティカルフローを検出する方法を提案し，生成画像を用いてその有効性を検討した。

オプティカルフローはフレーム間での画像上の各点の移動方向と移動距離を示す動きベクトル場である。オプティカルフローには物体の3次元形状や3次元運動に関する情報が含まれているので，オプティカルフロー検出は動画像処理の基本処理のひとつとなっている。

オプティカルフローの検出法はこれまでにいくつか提案されているが，画像上での移動量が大きい高速移動物体のオプティカルフロー検出が困難であるという問題を持っている。

本研究で提案する方法は，オプティカルフロー検出法の一つであるグラディエント法（勾配法）を拡張して，高速移動物体のオプティカルフローの検出を可能としたものである。具体的には処理に用いる画像に対して，前処理としてボカシ処理を施し，このボカシ画像をもとにオプティカルフローの算出を行う方法である。本研究ではさらに，グラディエント法の持つ処理時間の問題に対して，処理をエッジ線上の点にのみ限定するように改良することにより，高速化を図る方法についても検討を行った。

さらに生成画像を用いて従来の方法と今回提案する方法でオプティカルフロー検出能力の比較を行い，提案した方法の有効性を検証するとともに前処理における画像のボケ量などのパラメータについて検討を加えた。

1995年9月21日受理

* 情報工学科
** 情報工学科専攻修士学生

2. ボカシを用いたオプティカルフロー検出法

2.1 グラディエント法[1)]

はじめに本方法の基礎となっているグラディエント法によるオプティカルフロー検出法について述べる。

グラディエント法は画像上の各画素における明るさの空間的勾配および時間的勾配の関係を用いてオプティカルフローを求める方法である。以下に基本式を

示す。

ある時刻tにおける画像上座標(x, y)における濃淡値を$f(x, y, t)$，標本化された値を$f(i, j, k)$とする。時間tにおいて，座標(x, y)にある濃淡パターンがδt時刻経過した時の座標$(x+dx, y+dy)$に移動したとき，濃度値分布が変化しない（輝度保存条件）とすると，

$$f(x, y, t)=f(x+\delta x, y+\delta y, t+\delta t) \tag{1}$$

が成立する。式(1)の右辺をテイラー展開し，2項以上の項を無視すると次式が導かれる。

$$\frac{\partial f}{\partial x}\delta x+\frac{\partial f}{\partial y}\delta y+\frac{\partial f}{\partial t}\delta t=0 \tag{2}$$

式(2)の両辺をδtで割り，次式を得る。

$$\frac{\partial f}{\partial x}\frac{dx}{dt}+\frac{\partial f}{\partial y}\frac{dy}{dt}+\frac{\partial f}{\partial t}=0 \tag{3}$$

さらに，

$$\frac{\partial f}{\partial x}u+\frac{\partial t}{\partial y}v+\frac{\partial f}{\partial t}=0 \tag{4}$$

ただし，$u=\frac{dx}{dt}$, $v=\frac{dy}{dt}$（オプティカルフローに対応）

式(4)は動画像の濃淡値の時間・空間に関する偏微分（勾配）とオプティカルフロー速度(u, v)を関係づける式となる。

実際にオプティカルフローの2成分u, vを求めるには，式(4)以外に拘束条件が必要となる。これまでにいくつかの拘束条件が提案されているが，ここでは本研究に用いた離散化された場合の空間的大域最適化法について述べる。

離散化されたある点(i, j, k)とその近傍点におけるオプティカルフローを用いることにより滑らかさの逸脱を表す量$S_{i,j,k}$は次式で表される。

$$\begin{aligned}S_{i,j,k}=\frac{1}{4}[&\{u(i+1, j, k)-u(i, j, k)\}^2\\&+\{u(i, j+1, k)-u(i, j, k)\}^2+\{v(i+1, j, k)\\&-v(i, j, k)\}^2+\{v(i, j+1, k)-v(i, j, k)\}^2]\end{aligned} \tag{5}$$

一方，オプティカルフロー拘束方程式の誤差$C_{i,j,k}$は次式で表せる。

$$C_{i,j,k}=\{u(i, j, k)f_x+v(i, j, k)f_y+f_t\}^2 \tag{6}$$

ここで，式(5)と(6)から

$$e\equiv\sum_i\sum_j(S_{i,j,k}+\lambda C_{i,j,k}) \tag{7}$$

を定義し，このeを最小にする$\{u(i, j, k), v(i, j, k)\}$を求めることで，オプティカルフローを求める。これは，式(7)のeを$u(i, j, k)$および$v(i, j, k)$で微分し，その値が0となる条件より次式が導かれる。

$$(1+\lambda f_x^2)u(i, j, k)+\lambda f_xf_yv(i, j, k)=\bar{u}(i, j, k)-\lambda f_xf_t \tag{8}$$

$$\lambda f_xf_yu(i, j, k)+(1+\lambda f_y^2)v(i, j, k)=\bar{v}(i, j, k)-\lambda f_xf_t \tag{9}$$

さらに，変形して以下の式を得る。

$$\begin{aligned}&\{1+\lambda(f_x{}^2+f_y^2)\}u(i, j, k)\\&\quad=(1+\lambda f_y^2)\bar{u}(i, j, k)-\lambda f_xf_y\bar{v}(i, j, k)-\lambda f_xf_t\end{aligned} \tag{10}$$

$$\begin{aligned}&\{1+\lambda(f_x^2+f_y^2)\}v(i, j, k)\\&\quad=-\lambda f_yf_x\bar{u}(i, j, k)+(1+\lambda f_x^2)\bar{v}(i, j, k)-\lambda f_xf_t\end{aligned} \tag{11}$$

ここで，

$$\bar{u}(i, j, k)\equiv\sum_{(i', j')\in N_4(i, j)}u(i', j', k)/4 \tag{12}$$

$$\bar{v}(i, j, k)\equiv\sum_{(i', j')\in N_4(i, j)}v(i', j', k)/4 \tag{13}$$

ただし，$N_4(i, j)$は点(i, j)の4近傍の集合である。

式(10), (11)は$u(i, j, k)$および$v(i, j, k)$に関する連立方程式と考えることができ，これを解くことで次の繰り返し解法（反復法）が導かれる。

$$\begin{aligned}u^{l+1}(i, j, k)=&\bar{u}^l(i, j, k)\\&-\frac{\lambda\{f_x\bar{u}^l(i, j, k)+f_y\bar{v}^l(i, j, k)+f_t\}}{1+\lambda\{f_x^2+f_y^2\}}f_x\end{aligned} \tag{14}$$

$$\begin{aligned}v^{l+1}(i, j, k)=&\bar{v}^l(i, j, k)\\&-\frac{\lambda\{f_x\bar{u}^l(i, j, k)+f_y\bar{v}^l(i, j, k)+f_t\}}{1+\lambda\{f_x^2+f_y^2\}}fy\end{aligned} \tag{15}$$

ただし，lは反復回数であり，また

$$\begin{aligned}f_x\equiv[&\{f(i+1, j, k)+f(i+1, j+1, k)+f(i+1, j, k+1)\\&+f(i+1, j+1, k+1)\}-\{f(i, j, k)+f(i, j+1, k)\\&+f(i, j, k+1)+f(i, j+1, k+1)\}]/4\end{aligned} \tag{16}$$

$$\begin{aligned}f_y\equiv[&\{f(i, j+1, k)+f(i+1, j+1, k)+f(i, j+1, k+1)\\&+f(i+1, j+1, k+1)\}-\{f(i, j, k)+f(i+1, j, k)\\&+f(i, j, k+1)+f(i+1, j, k+1)\}]/4\end{aligned} \tag{17}$$

$$f_t \equiv [\{f(i,j,k+1)+f(i,j+1,k+1)+f(i+1,j,k+1)+f(i+1,j+1,k+1)\}-\{f(i,j,k)+f(i,j+1,k)+f(i+1,j,k)+f(i+1,j+1,k)\}]/4 \quad (18)$$

式(14),(15)を画像上の各画素に関して反復計算することで，オプティカルフロー $u(i,j,k)$ および $v(i,j,k)$ を求めることができる。

2.2 高速移動物体のオプティカルフロー

式(14),(15)によりオプティカルフローを求める際に，式(16)〜(18)よりオプティカルフローの速度を画素の中間点，つまり連続するフレームの中間点で推定していることがわかる。この処理の考え方を図1(a)の1次元の場合について簡単に述べる。図1(a)において，対象画像が時刻 t から $t+1$ で画素位置が a から a' へと移動する。このとき，図のようにベクトル f_t（時間的勾配）と角度 f_x（空間的勾配）からベクトル u を予想するのがグラディエント法の考えかたである。このベクトル u がオプティカルフローであり，図1(a)では実際の移動した画素 a' の位置と一致するので正しく検出されたことになる。

一方図1(b)のような移動量が大きい高速移動物体の場合は，時間的勾配 f_t と空間的勾配 f_x から予想されるベクトル u と実際の移動した量（$a \to a'$）が一致しないので，フローの検出が正しくできないことになる。従って，従来の方法では高速移動物体のオプティカルフロー検出は困難となる。

2.3 ボカシを用いたオプティカルフロー検出法

従来の方法では物体の移動量が大きいと空間的勾配と時間的勾配からオプティカルフローを予測することが難しいことが問題であった。この点を改善する目的で，本研究では画像に対してボカシ処理を施すことを考えた。図1(b)に対してボカシ処理を施した場合の様子を図1(c)に示す。

原画像にボカシ処理を施すことで画像が滑らかになり，空間的勾配と時間的勾配からオプティカルフローを予測することが可能となることがわかる。

画像にボカシ処理を施すのに本研究では次式(19)の2次元ガウス関数を対象画像の各画素に施す。

$$G(x,y;\sigma)=\frac{1}{2\pi\sigma^2}\exp\{-(x^2+y^2)/2\sigma^2\} \quad (19)$$

ここで，σ^2 はボカシの度合いを表すことになる。

画像関数 $f(x,y,t)$ に対してボカシ処理を施した後の画像関数を $g(x,y,t)$ とすると，

$$g(x,y,t)=\sum_{(i',j')\in I} G(x-i',y-j';\sigma)f(i',j',k) \quad (20)$$

ただし，I は $1 \leqq i' \leqq$（x 方向の画素サイズ）と $1 \leqq j' \leqq$（y 方向の画素サイズ）を範囲とするすべての (i',j') の集合。また $G(x,y;\sigma)$ は式(19)のガウス関数。

式(20)より明るさの空間的勾配および時間的勾配は次のようになる。

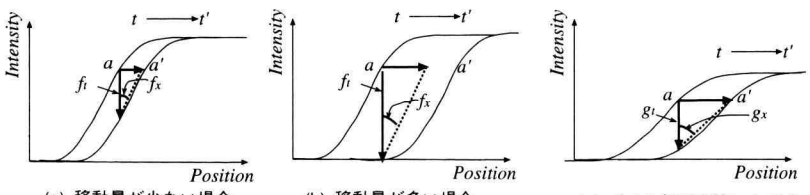
（x 方向の空間的勾配）

$$g_x=\sum_{(i',j')\in I}\left\{\frac{\partial G(x-i',y-j';\sigma)}{\partial x}\right\}f(i',j',k)$$
$$=-\left(\frac{1}{\pi\sigma^2}\right)\sum_{(i',j')\in I}(x-i')G(x-i',y-j';\sigma)f(i',j',k) \quad (21)$$

（y 方向の空間的勾配）

$$g_y=\sum_{(i',j')\in I}\left\{\frac{\partial G(x-i',y-j';\sigma)}{\partial y}\right\}f(i',j',k)$$
$$=-\left(\frac{1}{\pi\sigma^2}\right)\sum_{(i',j')\in I}(y-i')G(x-i',y-j';\sigma)f(i',j',k) \quad (22)$$

（時間的勾配）



(a) 移動量が少ない場合　(b) 移動量が多い場合　(c) ボカシ処理を施した場合

図1．グラディエント法によるオプティカルフロー検出

$$g_t = g(x, y, t+1) - g(x, y, t)$$
$$= \sum_{(i', j') \in I} G(x-i', y-j'; \sigma)\{f(i', j', k+1) - f(i', j', k)\} \quad (23)$$

式(21)〜(23)は $x-w \leqq i' \leqq x+w$ および $y-w \leqq j' \leqq y+w$ を範囲とする整数対 (i', j') の集合 $W(x, y; w)$ で成立すると考えてもよいので次式を得る。

$$g_x = -\left(\frac{1}{\pi\sigma^2}\right) \sum_{(i', j') \in W(x,y;w)} (x-i') G(x-i', y-j'; \sigma) f(i', j, k) \quad (24)$$

$$g_y = -\left(\frac{1}{\pi\sigma^2}\right) \sum_{(i', j') \in W(x,y;w)} (y-j') G(x-i', y-j'; \sigma) f(i', j, k) \quad (25)$$

$$g_t = \sum_{(i', j') \in W(x,y;w)} G(x-i', y-j'; \sigma)\{f(i', j, k+1) - f(i', j', k)\} \quad (26)$$

これら式(24)〜(26)を式(16)〜(18)の代わりにオプティカルフロー成分を求める反復式(14),(15)に用いると，次式となる。

$$u^{l+1}(i, j, k) = \bar{u}^l(i, j, k) - \frac{\lambda\{g_x \bar{u}^l(i, j, k) + g_y \bar{v}^l(i, j, k) + g_t\}}{1 + \lambda\{g_x^2 + g_y^2\}} g_x \quad (27)$$

$$v^{l+1}(i, j, k) = \bar{v}^l(i, j, k) - \frac{\lambda\{g_x \bar{u}^l(i, j, k) + g_y \bar{v}^l(i, j, k) + g_t\}}{1 + \lambda\{g_x^2 + g_y^2\}} g_y \quad (28)$$

式(27)と(28)を用いることで，原画像にボカシ処理を施した場合のオプティカルフローを求めることが可能となる。

3. エッジ線上に限定した処理

グラディエント法によるオプティカルフロー検出では式(14),(15)（あるいはボカシ処理を用いた式(27),(28)）による反復計算が画像上の各画素に対して行われる。そのために処理時間がかかるという問題が起こる。ところで，動画像処理においてオプティカルフローから画像認識を行う際には一般に画像全体のオプティカルフローを知る必要は必ずしもない。多くの場合，対象物体の輪郭やエッジ線上のオプティカルフローの情報で十分である。

このような観点から，実用性を失わないように高速化する方法として対象物体のエッジ線上の画素のみを用いてオプティカルフローを求めるように式(14),(15)（あるいは式(27),(28)）の拡張を行うことを考えた。

まず，式(14),(15)でも用いられていた4近傍局所平均，(式(12),(13))の代わりに次式の8近傍局所平均を用いる必要がある。

$$\bar{u}^l(i, j, k) \equiv \sum_{(i', j') \in N_8(i,j)} u^l(i', j', k)/8 \quad (29)$$

$$\bar{v}^l(i, j, k) \equiv \sum_{(i', j') \in N_8(i,j)} v^l(i', j', k)/8 \quad (30)$$

ただし，$N_8(i, j)$ は点 (i, j) を中心とした8近傍点全体の集合。

これは，エッジ点のみを用いる場合，4近傍では画素が無い可能性があるためである。（なお，式(29),(30)が式(14),(15)で適応可能であることは付録1に示す）

式(29),(30)を用いてエッジ線上に限定したオプティカルフローを算出する方法を以下に導出する。

まず，速度の平滑性として速度の8近傍の差の2乗和をとると，次式となる。

$$e = \sum_{(i,j) \in I_{\infty e}} [\lambda\langle u(i, j, k) f_x + v(i, j, k) f_y + \{f(i, j, k+1) - f(i, j, k)\}\rangle^2 + \sum_{(i,j) \in N_{8e}(i,j,k)} \langle\{u(i', j', k) - u(i, j, k)\}^2 + \{v(i', j', k) - v(i, j, k)\}^2\rangle]/\{2 \cdot n_n(i, j, k)\} \quad (31)$$

ただし，

$I_{\infty e} = \{(i, j) \mid (i, j)$ はエッジ点またはコーナ点，i, j は整数$\}$

$N_{8e}(i, j, k) \equiv \{(i', j') \mid\mid i'-i \mid \leq 1 \wedge \mid j'-j \mid \leq 1 \wedge not(i'=i \wedge j'=j) \wedge (i', j')$ はエッジ点またはコーナ点$\}$

であり，エッジ点やコーナ点を含む線は細線化してあるものとする。

上式を，$e = \lambda e_0 + e_1$ と表す。e_0 は輝度保存，e_1 は移動ベクトルの平滑性の誤差を示す。

e_1 は，

$$e_1 \equiv \sum_{(i,j) \in I_{\infty e}} \frac{1}{2 \cdot n_n(i, j, k)} \sum_{(i', j') \in N_{8e}(i,j,k)} [\{u(i', j', k) - u(i, j, k)\}^2 + \{v(i', j', k) - v(i, j, k)\}^2]$$
$$= \sum_{(i,j) \in I_{\infty e}} \frac{1}{2 \cdot n_n(i, j, k)} \sum_{(i', j') \in N_{8e}(i,j,k)} [\{u(i, j, k) - u(i', j, k)\}^2 + \{v(i, j, k) - v(i', j', k)\}^2] \quad (32)$$

となり，これを $u(i, j, k)$ および $v(i, j, k)$ で偏微分すると次の式を得る。

$$\frac{\partial e_1}{\partial u(i,j,k)}=\sum_{(i',j')\in N_{8e}(i,j,k)}\left\{\frac{1}{n_n(i,j,k)}+\frac{1}{n_n(i',j',k)}\right\}\{u(i,j,k)-u(i',j',k)\} \tag{33}$$

$$\frac{\partial e_1}{\partial v(i,j,k)}=\sum_{(i',j')\in N_{8e}(i,j,k)}\left\{\frac{1}{n_n(i,j,k)}+\frac{1}{n_n(i',j',k)}+\frac{1}{n_n(i',j',k)}\right\}\{v(i,j,k)-v(i',j',k)\} \tag{34}$$

一般に，$n_n(i,j,k)$ であるが，$n_n(i,j,k)=2$ の場合が圧倒的に多いので近似的処理として $n_n(i,j,k)=2$ とすると式 (33), (34) は以下のようになる。

$$\begin{aligned}\frac{\partial e_1}{\partial u(i,j,k)}&=\sum_{(i',j')\in N_{8e}(i,j,k)}\{ui,j,k)-u(i',j',k)\}\\&=|N_{8e}(i,j,k)|u(i,j,k)-\sum_{(i',j')\in N_{8e}(i,j,k)}u(i',j,k)\\&=|N_{8e}(i,j,k)|\{u(i,j,k)-\bar{u}(i,j,k)\}\end{aligned} \tag{35}$$

$$\begin{aligned}\frac{\partial e_1}{\partial v(i,j,k)}&=\sum_{(i',j')\in N_{8e}(i,j,k)}\{v(i,j,k)-v(i',j',k)\}\\&=|N_{8e}(i,j,k)|v(i,j,k)-\sum_{(i',j')\in N_{8e}(i,j,k)}v(i',j',k)\\&=|N_{8e}(i,j,k)|\{(i,j,k)-\bar{v}(i,j,k)\}\end{aligned} \tag{36}$$

ただし，

$$\bar{u}(i,j,k)=\frac{1}{|N_{8e}(i,j,k)|}\sum_{(i',j')\in N_{8e}(i,j,k)}u(i',j',k) \tag{37}$$

$$\bar{v}(i,j,k)=\frac{1}{|N_{8e}(i,j,k)|}\sum_{(i',j')\in N_{8e}(i,j,k)}v(i',j',k) \tag{38}$$

式 (34), (35) より次式を得る。

$$\begin{aligned}\frac{\partial e_1}{\partial u(i,j,k)}&=\frac{1}{n_n(i,j,k)}\sum_{(i',j')\in N_{8e}(i,j,k)}\{u(i,j,k)-u(i',j',k)\}\\&\quad+\frac{1}{n_n(i',j',k)}\sum_{(i',j')\in N_{8e}(i,j,k)}\{u(i,j,k)-u(i',j',k)\}\\&=2\alpha(i,j,k)u(i,j,k)-2\tilde{u}(i,j,k)\end{aligned} \tag{39}$$

$$\begin{aligned}\frac{\partial e}{\partial v(i,j,k)}&=\frac{1}{n_n(i,j,k)}\sum_{(i',j')\in N_{8e}(i,j,k)}\{v(i,j,k)-v(i',j',k)\}+\frac{1}{n_n(i',j',k)}\sum_{i',j')\in N_{8e}(i,j,k)}\{v(i,j,k)-v(i',j',k)\}\\&=2\alpha(i,j,k)v(i,j,k)-2\tilde{v}(i,j,k)\end{aligned} \tag{40}$$

ただし，

$$\alpha(i,j,k)=\frac{1}{2\cdot n_n(i,j,k)}+\sum_{(i',j')\in N_{8e}(i,j,k)}\{u(i,j,k)-u(i',j',k)\} \tag{41}$$

$$\tilde{u}(i,j,k)=\frac{1}{2n_n(i,j,k)+\sum_{(i',j')\in N_{8e}(i,j,k)}u(i',j',k)}+\sum_{(i',j')\in N_{8e}(i,j,k)}\frac{u(i',j',k)}{2n_n(i',j',k)}\{u(i,j,k)-u(i',j',k)\} \tag{42}$$

$$\tilde{v}(i,j,k)=\frac{1}{2n_n(i,j,k)}+\sum_{(i',j')\in N_{8e}(i,j,k)}v(i',j',k)+\sum_{(i',j')\in N_{8e}(i,j,k)}\frac{v(i',j',k)}{2n_n(i',j',k)}\{v(i,j,k)-v(i',j',k)\} \tag{43}$$

ここで，$n_n(i,j,k)=2$ for $\forall(i,j)\in I_e(k)$ とすると式 (41), (42), (43) は以下のようになる。

$$\alpha(i,j,k)=1 \tag{44}$$

$$\begin{aligned}\tilde{u}(i,j,k)&=\frac{1}{2}\sum_{(i',j')\in N_{8e}(i,j,k)}u(i',j',k)\\&\equiv\bar{u}(i,j,k)\end{aligned} \tag{45}$$

$$\begin{aligned}\tilde{v}(i,j,k)&=\frac{1}{2}\sum_{(i',j')\in N_{8e}(i,j,k)}v(i',j',k)\\&\equiv\bar{v}(i,j,k)\end{aligned} \tag{46}$$

ここで，

$$f_t\equiv f(i,j,k+1)-f(i,j,k) \tag{47}$$

とおくと，式 (39), (40) は以下のように書き換えられる。

$$(\alpha+\lambda f_x^2)u(i,j,k)+\lambda f_x f_y v(i,j,k)=\alpha\bar{u}(i,j,k)-\lambda f_x f_t \tag{48}$$

$$\lambda f_y f_x u(i,j,k)+(\alpha+\lambda f_y^2)v(i,j,k)=\alpha\bar{v}(i,j,k)-\lambda f_y f_t \tag{49}$$

式 (48), (49) は $u(i,j,k)$, $v(i,j,k)$ に関する連立方程式と考えることができるので，それぞれについて解くと

$$u(i,j,k)=\bar{u}(i,j,k)-\lambda\frac{f_x\bar{u}(i,j,k)+f_y\bar{v}(i,j,k)+f_t}{\alpha+\lambda(f_x^2+f_y^2)}f_x \tag{50}$$

$$v(i,j,k)=\bar{v}(i,j,k)-\lambda\frac{f_x\bar{u}(i,j,k)+f_y\bar{v}(i,j,k)+f_t}{\alpha+\lambda(f_x^2+f_y^2)}f_y \tag{51}$$

となり，これを繰り返し解法（反復法）に変形すると

次の式を得る。

$$u^{l+1}(i,j,k)=\bar{u}^{l}(i,j,k) - \lambda\frac{f_x\bar{u}^{l}(i,j,k)+f_y\bar{v}^{l}(i,j,k)+f_t}{\frac{|N_{8e}(i,j,k)|}{2}+\lambda(f_x^2+f_y^2)}f_x \quad (52)$$

$$v^{l+1}(i,j,k)=\bar{v}^{l}(i,j,k) - \lambda\frac{f_x\bar{u}^{l}(i,j,k)+f_y\bar{v}^{l}(i,j,k)+f_t}{\frac{|N_{8e}(i,j,k)|}{2}+\lambda(f_x^2+f_y^2)}fy \quad (53)$$

上式を用いることで，エッジ線上のみでのオプティカルフロー検出が可能となる。

4. 実 験 方 法

今回提案した方法の有効性を検証するために，コンピュータにより生成した動画像を用いて，オプティカルフロー検出を行った。

用いた画像は次の3種類であり，それぞれ大きさが128×128画素で256階調のモノクロ画像である。各画像の移動前の原画像およびエッジ画像を図2に示す。各画像はそれぞれ次のような動きをする。

（画像A）画像上で上方に並進移動する物体。上に8画素，右に2画素の移動量である。

（画像B）画像上で右上に並進移動する物体。上に12画素，右に8画素の移動量である。

（画像C）回転移動物体。移動量は右回りに約3度である。

実験においては，前処理としての画像のボケの度合い（σ）が，求められるオプティカルフローの検出精度や計算における収束回数に与える効果について調べた。また，オプティカルフローの滑らかさの重み（λ）のオプティカルフロー検出への影響についても調べた。

5. 実 験 結 果

5.1 オプティカルフロー検出

ボカシ処理におけるボカシの度合いとして式(19)におけるσの値を1, 2,・・・・, 10（画素）の10種類について実験を行い，効果の比較を行った。画像Aに対するボカシ処理後の画像をσ=2, 6, 10の場合について図3に示す。

3種類の画像について，σ=2, 6, 10の条件で検出されたオプティカルフローを図4～6に示す。図4～6では，マーク■が移動前，マーク□が移動後のエッジ点を示し，実線が求められたオプティカルフローである。（オプティカルフローの始点は移動前のエッジ点である）

実験結果より，画像A～Cのすべての場合で，ボカシ処理がない従来の方法ではオプティカルフローが正

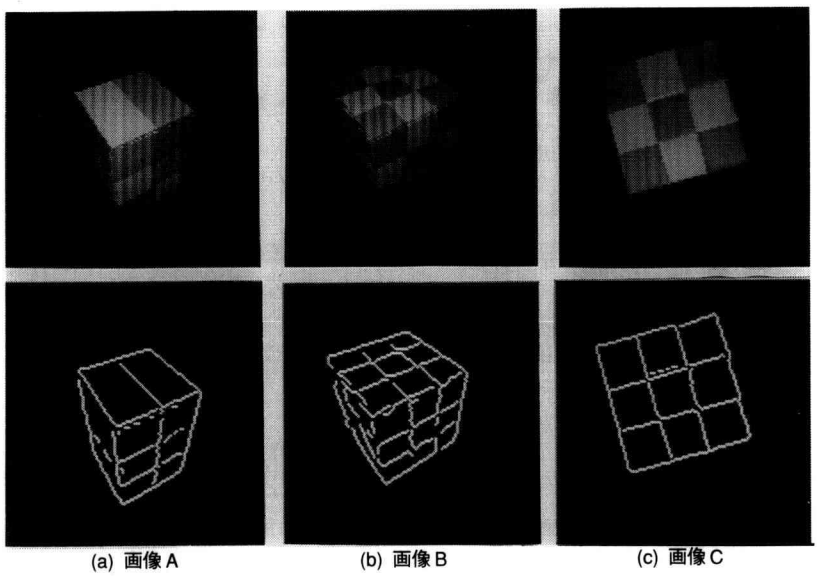


図2．実験に用いた原画像とエッジ画像

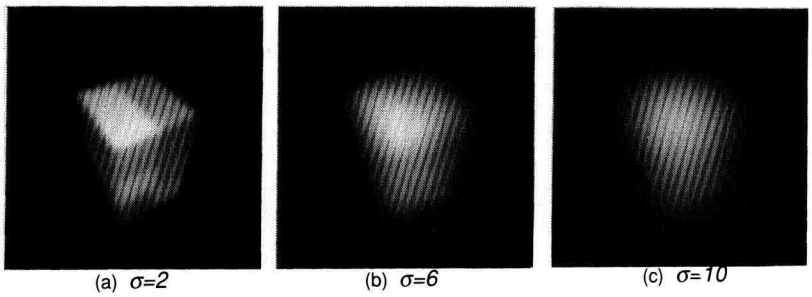


図3．ボカシ処理後の画像

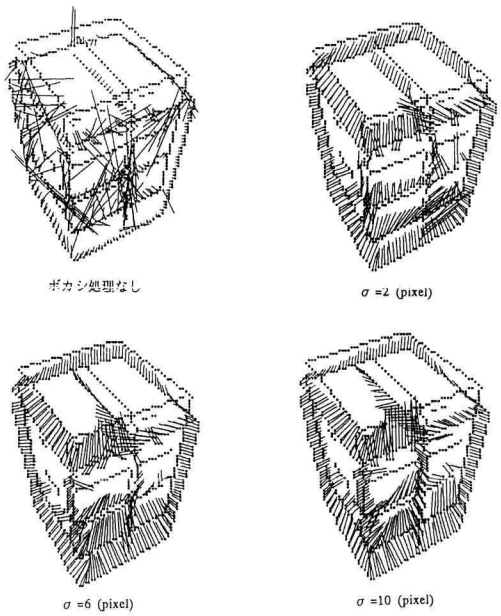


図4．求めたオプティカルフロー（画像A）

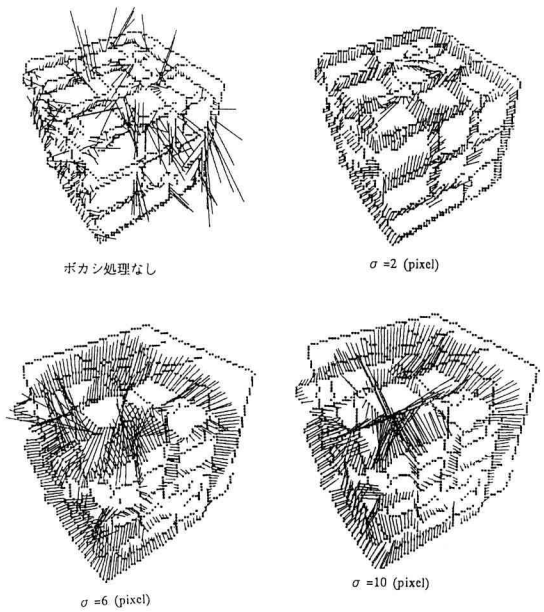


図5．求めたオプティカルフロー（画像B）

しく検出されていないが，ボカシ処理を施す提案方法では正しい検出が行われたことを示している。

画像Aに関しては，$\sigma=2$ のボカシ条件で多くの位置で正しいオプティカルフローの検出が得られている。しかし，$\sigma=10$ の条件では移動量によっては正しい検出ができず，全体としても検出精度が不十分であった。一方，画像Bでは $\sigma=2$ の条件では不十分で $\sigma=10$ 程度のボカシ量が必要であった。これより，最適なオプティカルフローを得るには対象画像の特性に合わせたボカシの量を設定する必要があることがわかる。

回転移動をしている画像Cに関しては $\sigma=2$ のボカシ条件である程度の検出ができていたが，ボカシの度合いが大きくなると検出ができなかった。また，画像Cよりも回転移動量が大きくなるとまったく検出ができなかった。回転移動のように画素毎に移動量や移動方向が大きく異なる場合はボカシ処理を行うだけではオプティカルフロー検出が十分できないことがわかる。

5.2　ボカシの度合いと収束性

本方法では繰り返し計算によりオプティカルフローを求めている。従って収束性の良さが重要となる。従来のボカシ処理を用いない方法では画素サイズ（今回の場合128）程度の繰り返しでほぼ収束すると言われ

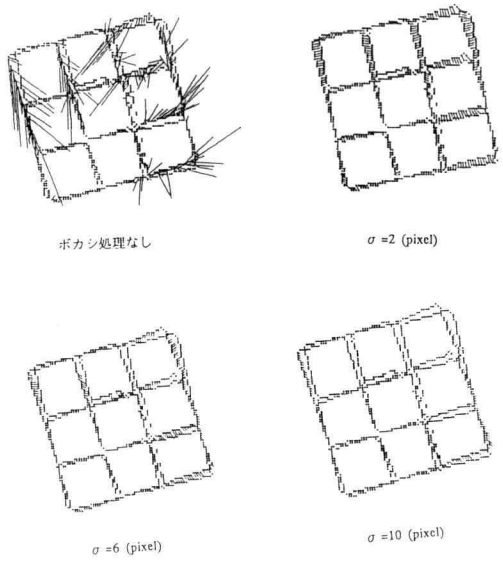


図6．求めたオプティカルフロー（画像C）

ている。今回提案した方法でボカシを施すことで収束性にどのような影響があるかを調べた。

収束判定には次の式を用いた。

$$\text{収束判定値} = \frac{\sum_{i,j}\{u'-u)^2+(y'-v)^2\}}{n^2} \qquad (54)$$

ただし，定数nは処理画像全画素数，u'およびv'は，成分u, vよりも繰り返し数を1回多く演算したオプティカルフロー成分の値である。式(54)による収束判定値が0に近づくほど繰り返し法の演算値が収束していることを示す。

実験では，収束判定値として我々のこれまでの検討で経験的に得られた10^{-5}を用い，この値になるまでの繰り返し回数を調べ評価を行った。結果を表1(a)～(c)に示す。

収束までの繰り返し数がボカシ処理のない場合にくらべ多いことがわかる。また，通常言われている画素の行数（＝列数）の回数（128回）よりも多い。しかし，ボカシ処理のない場合はオプティカルフローが正しく求まらないことや物体の移動距離が通常のオプティカルフロー計算の場合よりも大きいことを考慮すると，繰り返し数が多いのは必ずしもボカシ処理のみが原因であるとは特定できない。特にオプティカルフローの検出が困難であった画像Cの場合では，全体に繰り返し数が少なく，繰り返し数がオプティカルフロー検出

表1(a)．ボカシの度合いと収束回数（画像A）

ボカシの度合い σ（画素）	繰り返し数（回）
ボカシ処理なし	97
1	147
2	318
3	337
4	446
5	361
6	279
7	405
8	306
9	292
10	265

表1(b)．ボカシの度合いと収束回数（画像B）

ボカシの度合い σ（画素）	繰り返し数（回）
ボカシ処理なし	63
1	226
2	231
3	293
4	266
5	362
6	292
7	300
8	229
9	269
10	225

表1(c)．ボカシの度合いと収束回数（画像C）

ボカシの度合い σ（画素）	繰り返し数（回）
ボカシ処理なし	47
1	41
2	66
3	105
4	64
5	43
6	50
7	27
8	35
9	26
10	26

表2．滑らかさの重みλと収束回数（画像A）

ボカシの度合い σ（画素）	λ 0.01	0.1	1	10	100
ボカシ処理なし	56	75	97	93	91
1	65	113	147	151	151
2	79	297	318	318	318
3	83	267	337	343	343
4	68	310	446	474	476
5	77	317	361	371	371
6	71	241	279	279	279
7	74	328	405	406	404
8	72	256	306	312	314
9	74	318	292	282	281
10	81	243	265	264	264

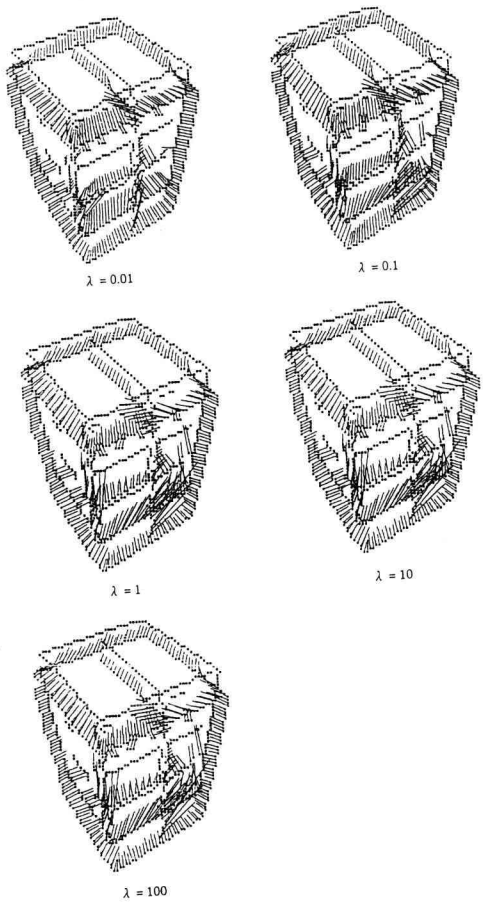


図7．求めたオプティカルフロー（λ による違い）

精度を反映していないことがわかる。今後ボカシ処理が収束にあたえる影響を調べるために収束の過程を詳細に検討する必要があろう。

5.3 滑らかさの重みの影響

オプティカルフロー場の滑らかさの重みλが検出処理においてどのような影響があるかを収束回数により評価した。画像Aに対して調べた結果を表2に示す。またボカシの度合い $\sigma=2$ の場合の求めたオプティカルフローを図7に示す。

収束回数では $\lambda=0.01$ の場合に顕著に少ないが，その他の条件では大きな違いは見られない。また求められたオプティカルフローではやや違いが見られるものの全体には大きな違いが見られない。従って，オプティカルフロー検出精度の点では滑らかさの重みλは広い範囲にわたり設定可能なことがわかる。繰り返し回数に関してはばらつきがみられるが今回の実験ではその原因はわからず，更に実験が必要と考える。

5.4 処理速度

処理の高速化の目的で今回提案したエッジ線上の点のみを用いる演算処理方法を実験より検討した。先に述べた結果のように正常な動作が確認できた。また処理速度に関しても今回用いた画像では画像全体を用いる場合に比べ約10倍の速度の向上が認められた。

6. ま　と　め

本研究では，オプティカルフロー検出において対象画像としては不向きとされてきたフレーム間の移動量が大きい高速移動物体のオプティカルフローを検出する目的で，対象画像にガウス平滑化によるボカシ処理を加え，その明るさ（階調値）の勾配を滑らかにすることによりオプティカルフロー検出を可能とする方法を提案した。

生成動画像を用いた実験結果より，対象画像にボカシ処理を施す本処理によりフレーム間の移動量が大きい並進運動物体のオプティカルフロー検出が可能となった。

ボカシの度合いは対象動画像により適当な値があることがわかった。一方，回転運動をする物体に関しては本処理においても十分な検出ができなかった。また，オプティカルフローの場の滑らかさの重みの値を変化させたところ，ボカシ処理を施している本処理では同パラメータは検出されるオプティカルフローに対する影響が非常に緩いことが認められた。

本研究においては処理速度の向上を目的とした画像のエッジ線のみを対象としたオプティカルフロー検出の提案についても実験を行った。その結果正しいオプティカルフロー検出が確認でき，また期待通りの処理速度の大幅な向上が認められた。

今後の課題としては，

(1) 画素位置によって移動量が異なる回転移動物体のオプティカルフロー検出
(2) 繰り返し計算における収束過程の詳細な検討
(3) ボカシの階層化による最適ボカシ量の検討
(4) オプティカルフロー成分の精度の追求

などが挙げられる。

なお，本研究は本学動画像処理室のシステムを用い

て行った。また，本研究の一部は本学大学院情報工学専攻修士課程平成6年度修了生である田中裕子さんの研究として行われた。

参 考 文 献

1) 橋本，百田，野村："4. オプティカルフローの検出"，「パソコンによる動画像処理」，森北出版，1993, p. 133～178.

付 録 1

本文における式 (29)，(30) の8近傍局所平均が式 (14)，(15) に適応できることを示す。

速度の平滑化として速度の8近傍の差の2乗和をとると次式として表せる。

$$e=\sum_{(i,j)\in N_\infty}[\lambda\langle u(i,j,k)f_x+v(i,j,k)f_y+\{f(i,j,k+1)-f(i,j,k)\}\rangle^2+\frac{1}{16}\sum_{(\Delta i,\Delta i)\in N_{r8}}\{u(i+\Delta i,j+\Delta i,k)-u(i,j,k)\}^2+\{v(i+\Delta i,j+\Delta j,k)-v(i,j,k)\}^2\rangle] \tag{A-1}$$

ただし，$I_\infty=\{(i,j)\mid i,j$ は整数$\}$

$N_{r8}=\{(\Delta i,\Delta j)\mid -1\leq 1\wedge -1\leq\Delta j\leq 1\wedge(\Delta i,\Delta j)\neq 0\}$

式 (A-1) を $e=\lambda c+s$ とおくと，以下のように表せる。

$$s=\frac{1}{16}\sum_{(i,j)\in N_\infty}\sum_{(\Delta i,\Delta j)\in N_{r8}}[\{u(i+\Delta i,j+\Delta j,k)-u(i,j,k)\}^2+\{v(i+\Delta i,j+\Delta j,k)-v(i,j,k)\}^2]$$
$$=\frac{1}{16}\sum_{(i,j)\in N_\infty}\sum_{(\Delta i,\Delta j)\in N_{r8}}[\{u(i',j',k)-u(i'-\Delta i,j'-\Delta j,k)\}^2+\{v(i',j',k)-v(i'-\Delta i,j'-\Delta j,k)\}^2] \tag{A-2}$$

上式を x および y に関して偏微分すると，以下のようになる。

$$\frac{\partial s}{\partial u(i,j,k)}=\frac{2}{16}\sum_{(\Delta i,\Delta j)\in N_{r8}}[\{u(i,j,k)-u(i+\Delta i,j+\Delta j,k)\}+\{u(i,j,k)-u(i-\Delta i,j-\Delta j,k)\}]2\cdot[u(i,j,k)-\{\sum_{(\Delta i,\Delta j)\in N_{r8}}u(i+\Delta i,j+\Delta j,k)+\sum_{(\Delta i,\Delta j)\in N_{r8}}u(i-\Delta i,j-\Delta j,k)\}/16]$$
$$=2\cdot\{u(i,j,k)-\bar{u}(i,j,k)\} \tag{A-3}$$

$$\frac{\partial s}{\partial u(i,j,k)}=\frac{2}{16}\sum_{(\Delta i,\Delta j)\in N_{r8}}[\{u(i,j,k)-v(i+\Delta i,j+\Delta j,k)\}+\{u(i,j,k)-v(i-\Delta i,j-\Delta j,k)\}]2\cdot[v(i,j,k)-\{\sum_{(\Delta i,\Delta j)\in N_{r8}}v(i+\Delta i,j+\Delta j,k)+\sum_{(\Delta i,\Delta j)\in N_{r8}}v(i-\Delta i,j-\Delta j,k)\}/16]$$
$$=2\cdot\{v(i,j,k)-\bar{v}(i,j,k)\} \tag{A-4}$$

ただし，$\bar{u}(i,j,k)=\sum_{(\Delta i,\Delta j)\in N_{r8}}u(i+\Delta i,j+\Delta j,k)/8$

$\bar{v}(i,j,k)=\sum_{(\Delta i,\Delta j)\in N_{r8}}v(i+\Delta i,j+\Delta j,k)/8$

このとき，本文の式 (10), (11) が極小条件としてそのまま成立する。従って，式 (14), (15) にも適用できる。