

時空間的近傍を用いた速度情報の抽出

やちど
谷内田正彦 (大阪大学基礎工学部)

まえがき

動画像から速度情報を抽出することは、物体の速度や軌跡を決定するのに重要であるばかりでなく、画像を分割 (Segmentation) するのに一つの重要な“手がかり”となる。例えば、明るさが等しいため、静止画像のみでは分割し難い2つの物体を速度の違いを利用して分割したり、またその反対に、実際には1つの物体であるが、2つに見える物体を、速度の類似性を利用して結合したりするのに速度情報は重要な働きをする。(1)

この速度情報を得る素直な方法として、画像中で他の部分とは異なる特徴的な点を発見して、照合法 (matching method) により画像間で特徴点の対応をとるという方法が考えられる。(2) この方法は、最近の弛緩法を適用する手法の発展などにより、比較的信頼性良く特徴点の対応をとることができるようになってきた。(3,4) しかし、この照合法を用いる方法では数少ない点の対応しかとれず、当然のことながら、画像分割に利用できる程の“密”な点の速度情報を得ることはできない。

密な点の速度情報を得る手法として、動画像においては、空間的な明るさの変化と時間的な明るさの変化は互いに独立ではなく、次の様な関係を満たすという事実を利用するという方法が提案されている。すなわち

$$V_x * G_x + V_y * G_y = Dt \quad (1)$$

ここで、 (V_x, V_y) は点 (x, y) における速度ベクトルの2つの成分、 (G_x, G_y) は空間的な明るさの勾配の2つの成分、 Dt は連続フレーム間の明るさの変化分である。

式(1)では、2つの未知数に対し、一つの方程式しか与えられていないので、速度ベクトルの両成分を決定するには、さらに他の拘束条件が必要となる。Cafforio ら⁽⁵⁾あるいはFennema ら⁽⁶⁾は動物体に対応する領域内では速度がすべて等しいと仮定して、ハフ (Hough) 変換に似た手法を用いて画像内の動物体の速度を決定した。この大局的な方法は物体の速度を比較的精度良く求められるが、上で述べた仮定のために、画面に対して平行に移動する物体にしか適用できないという欠点を持つ。すなわち、その“見かけ”の速度ベクトルは物体の各点において一定ではなくなるため上の仮定に反するからである。

一方、Horn ら⁽⁷⁾は、最近、“速度ベクトルは滑らかに変化する”という仮定のもとに、各点の速度ベクトルを求める方法を提案した。この仮定は、物体が剛体 (rigid body) であり、滑らかな運動をしていると、物体が重なって見えない限り、その見かけの速度も滑らかに変化するので、かなり一般的な仮定である。しかし、Horn らの方法は、処理時間がかかると同時に、上に述べた物体の剛体性という仮定以外に、①物体の表面は平面的である、②物体の表面の明るさは微分可能である(すなわち表面上で明るさは滑らかに変化しており、明るさの一樣な部分や縁点のように明るさが急激に変化することを許さない)、③物体は重なって見えないなどの多くの仮定をしている。これらの仮定を満たす物体は実際には少ないため、理想的な合成画像に対して実験を行なっているが、実際の画像に対してどの程度有効であるのかは明

らかでない。

本論文では、もう少し現実的な仮定のもとに、実際の動画像から画像の各点の速度ベクトルを求める方法について述べる。前にも述べたように、画像中で他の部分とは異なる特徴的な点が画面間で最も信頼性良く対応がとれる点である。この特徴点の対応の方法は、最近、急速に進歩しているのでこれらは既知であると仮定し、この特徴点から他の点の速度ベクトルを“速度ベクトルの変化の滑らかさ”という仮定を用いて求める。⁽⁸⁾最初の2画面から各点の速度ベクトルが決定されると、以降の画面では、前画面までの処理結果を利用してmodel-drivenな方法で能率良く処理を行う。このmodel-drivenな処理方法は単に処理能率を向上させるだけでなく、処理結果の信頼性も向上させる。⁽⁹⁾こうして、動画像全体に渡って各点での速度ベクトルが得られる。しかし、こうして得られた速度ベクトルは、雑音の影響を受けて誤った値を持つ部分もある。このため、“速度ベクトルの変化の滑らかさ”という拘束条件をくり返しの(iterative)に時空間的な近傍を用いて伝播させることにより、誤りを軽減させる。

2. 特徴点からの拘束条件の伝播

図1からも解るように、速度ベクトルを局所的に求めることは一般的には

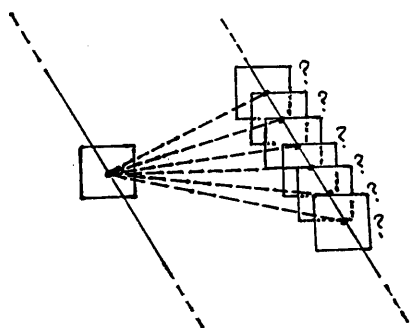


図1. 局所的に対応をとることは難しい

難しい。すなわち、図で正方形で囲まれた局所領域は次画面で多くの部分に対応するので、この領域の速度を決定することはわれわれ人間でも困難である。しかし、図2に示すように特徴的な点があると、これらの特徴点の間で対応をとることはそれ程困難ではない。また、一旦、特徴点の対応が解ると、物体が剛体であると仮定すればそれか他の点の対応を拘束するため、他の点の対応もそれ程困難ではなくなる。このような考え方にに基づき、ここでは、次に述べる隣接点間の速度ベクトルの持つ拘束条件を利用して、特徴点の速度ベクトルから他の点の速度ベクトルを推定する。

物体が剛体であると仮定すると、物体が任意の3次元的な運動をしても隣接点間の速度ベクトルの変化は“滑らか”であると考えることが出来る。ここで、この速度ベクトルの変化の滑らかさを次のように表現する。⁽⁷⁾

$$E2 = \text{SORT}[(\bar{v}_x - v_x)^2 + (\bar{v}_y - v_y)^2] \Rightarrow \text{Minimum} \quad (2)$$

ここで、 (v_x, v_y) は点 (x, y) の速度ベクトルを示し、また $(\bar{v}_x, \bar{v}_y)$ は点 (x, y) の近傍の点の速度ベクトルの平均を示す。一方、画像には雑音が含まれているので、式(1)を次のように書きかえる。

$$E1 = v_x * G_x + v_y * G_y + D_t \Rightarrow \text{Minimum} \quad (3)$$

次に、 $E1, E2$ の自乗の和

$$E^2 = \alpha^2 * E1^2 + E2^2 \quad (\alpha: \text{定数}) \quad (4)$$

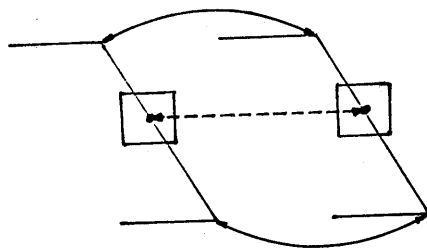


図2. 特徴点の対応が他の点の対応を拘束する

を最小にする。(4) 式を偏微分することにより、次の式が得られる。

$$\begin{aligned} \overline{v_x} &= \overline{v_x} - G_x * (G_x * \overline{v_x} + G_y * \overline{v_y} + Dt) / (\alpha^2 + G_x^2 + G_y^2) \\ \overline{v_y} &= \overline{v_y} - G_y * (G_x * \overline{v_x} + G_y * \overline{v_y} + Dt) / (\alpha^2 + G_x^2 + G_y^2) \end{aligned} \quad \text{----- (5)}$$

(5) 式において、 $\overline{v_x}$, $\overline{v_y}$ を除いては、すべて既知であるので、 $\overline{v_x}$, $\overline{v_y}$ が決まれば、点(x, y)の速度ベクトル $\overline{v_x}$, $\overline{v_y}$ を決定することができる。

しかし、 $\overline{v_x}$, $\overline{v_y}$ は点(x, y)の周りの点の速度ベクトルであるので、一般的には既知でない。ここで、特徴点の速度ベクトルが既知であるとするとき、それを初期値として他の点のベクトルを(5)式を用いて次のような手順で決定できる。

(1) 各特徴点の隣接点の速度ベクトルを決定する。すなわち、ある特徴点の速度ベクトルを $\overline{v_x^F}$, $\overline{v_y^F}$ とした時、それらを式(5)における $\overline{v_x}$, $\overline{v_y}$ とし、同式から隣接する点の速度ベクトルを決定する。

(2) 上で速度ベクトルの求まった点の隣接点の速度ベクトルを決定する。この時、式(5)の $\overline{v_x}$, $\overline{v_y}$ としては、5×5の近傍においてすでに速度ベクトルの決定している点の平均を用いる。

(3) 画像上のすべての点の速度ベクトルが求まるまで(2)の処理をくり返す。

このように、ここでは信頼性良く速度ベクトルの求まる所を出発点として、拘束条件を用いることにより対応の付け難い部分の速度ベクトルを求めている。このような拘束条件の伝播方法をとると、信頼性などによる差別を付けず自由に拘束条件を伝播させる方法と比較すると、誤った情報が伝播し難いという利点がある。特に拘束条件の伝播という手法は多数派(majority)が少数派に強く影響を及ぼすという傾向があるが、信頼性よく対応付けの行える特徴的な点は他の点に比べて圧倒

的に少数であるため、このような伝播方法は重要である。この伝播方法を、自由な伝播方法と区別するために、強制的な拘束条件の伝播と呼ぶ。

3. 実際の処理手順

実際の処理手順を図3に示す。式(1)が成立するのは明るさについて微分可能な領域であり、縁点のように明るさが急激に変化している部分や明るさの変化しない一様な領域については成立しない。明るさの一様な領域内の各点の対応をとることは、いずれにしても局所的には困難である。しかし、縁点は重要な点であるので、この部分で式(1)が成立するように、入力画像列 $I(x, y, t)$, $t=1, \dots, N$ をまず平滑化する。平滑化のオペレーターとしては、いくつか考えられるが、ここでは5×5点を単純に平均化するオペレーターを使用している。

平滑化画像 $A(x, y, t)$, $t=1, \dots, N$ が得られると、次に、各点の明るさの勾配(G_x, G_y)と連続画面間の明るさの変化分(Dt)を求める。明るさの勾配を求める方法としては、Sobelのオペレーターを用いている。明るさの時間的な変化分を求める方法としてはいく

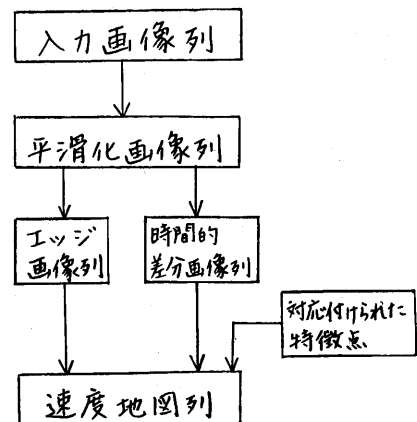


図3. 実際の処理手順

つか考えられるが、ここでは次画面と現画面との明るさの差を用いている。すなわち点 (x, y) における前画面と現画面の明るさを $A(x, y, t-1)$, $A(x, y, t)$ とした時、明るさの変化分 $Dt(x, y, t)$ は

$Dt(x, y, t) = A(x, y, t) - A(x, y, t-1)$ (6) で求める。こうして、勾配の画像と時間的差分画像が得られると、2.で述べた手法により画像の各点の速度ベクトルを得ることができる。しかし前にも述べた様に、明るさの一律な領域内では、 Dx , Dy , Dt はともにゼロになるため、ここでは速度ベクトルを求めることは困難である。このため、2.で述べた手法により速度を決定する時、 $|Dx + Dy|$ がある値以下の時は、その点の速度ベクトルは求めることができない、あるいは求められたとしても信頼性が低いと考えて、除外する。

以上のようにして、特徴点から式(5)を用いて他の点の速度ベクトルが決定されていくが、ここで問題となる点はいつ、この伝播を停止するかである。物体が剛体で、しかも2つの物体が重なって見えることはないと仮定すると、速度ベクトルはすべての点で滑らかに変化すると考えることができるのであまり問題とはならない。しかし、2つの物体が重なり合うと、重なりあった点で速度ベクトルの変化は不連続となる。速度ベクトルは特徴点から等速度で伝播していくため、図4の①に示すように、2つの物体が重なり合っている時は、異なる速度ベクトルを持つ物体の影響が伝播して来るため、この部分での速度ベクトルの推定は誤差が大きくなる。

この問題を解くため、ここでは、式(4)で求められている誤差 E^2 を各点について記憶しておく。そうすると、異なった速度を持つ物体から伝播してきた点では誤差 E^2 が大きな値を持つ。一

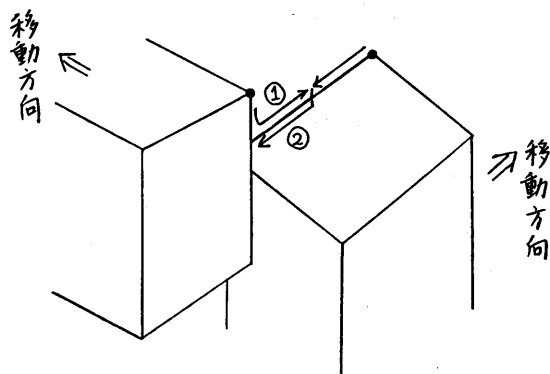


図4. 伝播の停止

方、本来の物体から伝播してきた速度ベクトルは、速度ベクトルの滑らかな変化という仮定を満たすので、当然 E^2 は小さくなる。このため、すでに速度ベクトルが求められている点であっても、図4の②に示すように他から伝播して求めた速度ベクトルの誤差 E^2 が小さければ、それを優先することにする。こうして、もう伝播できなくなった時、(すなわちすでに求められている速度ベクトルと新たに伝播して求められた速度ベクトルがあまり変わらない時)、伝播を停止する。

4. 連続画面におけるプランを用いた速度ベクトルの推定

3.で述べた方法により、入力画像列 $I(x, y, t)$, $t=1, \dots, N$ から速度ベクトルの地図列 $V(x, y, t)$, $t=2, \dots, N$ を求めることができる。例えば、2つの画像 $I(x, y, t-1)$ と $I(x, y, t)$ から速度ベクトルの地図 $V(x, y, t)$ を得ることができる。しかし、毎回、2つの入力画像から速度ベクトルの地図を計算するのは非能率的であり、例えば最初の2画面 $I(x, y, 1)$ と $I(x, y, 2)$ から速度ベクトル地図 $V(x, y, 2)$ が求められれば、その結果を有効に利用して次画面の速度ベクトル $V(x, y, 3)$ を求めた方が能率が良い。(10, 11)

4.1 フランを用いた速度ベクトルの推定

ここで、連続画面間で速度ベクトルが滑らかに変化すると仮定すると、前画面の速度ベクトル地図 $V(x, y, t-1)$ から現画面での速度ベクトル地図 $V(x, y, t)$ を式(5)を用いて推定できると考えられる。すなわち、式(5)において、 $\bar{v}_x, \bar{v}_y$ を前画面における点 (x, y) の周りの点の速度ベクトルの平均、 (G_x, G_y) を現画面における明るさの勾配、 Dt を現画面と前画面との明るさの変化分とすると、これらはすべて既知であるので、現画面における点 (x, y) の速度ベクトルは式(5)から直ちに求めることができる。

しかし、ここで問題となる点は、物体が移動しているため、前画面の速度地図から求めた $\bar{v}_x, \bar{v}_y$ が現画面でのそれと必ずしも適応しないという点である。特に明るさが一様な所では速度ベクトルは未知としているので、図5に示すように、 $\bar{v}_x, \bar{v}_y$ を求めようとしても未知となる。この問題を解決するために、前画面で求めた速度地図から次画面における状態を予測した地図を作成する。すなわち、 $V(x, y, t-1)$ から各点の持っている速度ベクトル分だけ移動した予測地図 $V'(x, y, t)$ を作成し、これを $\bar{v}_x, \bar{v}_y$ を計算するのに用いる。

4.2 重みをつけた伝播

4.1で述べた方法は物体が重なって見えない場合には適効であるが、物体

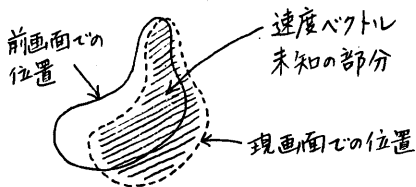


図5. 予測の必要性

が重なって見える場合に問題点がある。すなわち、物体が重なっている部分では、明るさの勾配 (G_x, G_y) も画面間の明るさの変化 (Dt) も信頼性良くは求まらないので、当然速度ベクトル (V_x, V_y) も信頼性良くは求まらない。ここで、式(5)を良く検討してみると、前画面での隣接点の速度ベクトルの平均 $(\bar{v}_x, \bar{v}_y)$ を基に現画面における点 (x, y) の速度ベクトル (V_x, V_y) を推定している。したがって、画像を単純に平均化すると縁点 (edge) の部分がボケるように、2つの物体の速度ベクトルの平均的な速度ベクトルが徐々に伝播して行く。特に、図6①に示すように2つの物体が重なっている部分での信頼性の低い速度ベクトルが他の部分に伝播して行くのは望ましくなく、逆に②に示すように信頼性の高い部分から伝播して行き、信頼性の低い部分を信頼性の高い部分の値から訂正するという事が望ましい。これを実現する方法として、①画像を平滑化する時に用いられるような非線型な平均化と②重みをつけた平均化が考えられる。両方法ともほぼ期待通り)の結果が得られるが、ここでは計算の比較的簡単な②の方法について述べる。

重み付け平均を行なう場合、重み付けの評価として何を定むかが問題となるが、ここでは速度ベクトルの分散をその評価として用いる。すなわち、2

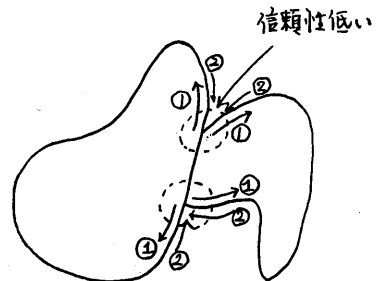


図6. 重み付けされた伝播の必要性

つの物体が重なっている部分では速度ベクトルは分散が大きいが、他の部分では速度ベクトルの変化は滑らかであるので分散が小さいと考えられるからである。

実際の処理手順としては、前画面での速度ベクトルが得られると、まず各点について 5×5 点の近傍で速度ベクトルの分散を求める。次に4.1で述べた方法により現画面における速度ベクトルの予測地図を求めるが、この時各点には速度ベクトルと同時にその信頼性として分散の値も記述しておく。この予測地図を基に現画面での速度ベクトルを式(5)を用いて求めるが、この時 $\overline{v_x}$, $\overline{v_y}$ を次のように定義する。

$$\overline{v_x} = \frac{\sum_{j=-2}^2 \sum_{i=-2}^2 W(x+i, y+j) * v_x(x+i, y+j)}{\sum_{j=-2}^2 \sum_{i=-2}^2 W(x+i, y+j)}$$

$\overline{v_y}$ = 同様

ここで、 $W(x, y)$ は点 (x, y) における分散の逆数であり、 $v_x(x, y)$, $v_y(x, y)$ は予測地図における点 (x, y) の速度ベクトルである。

このように重み付け平均を用いることにより、信頼性の良い部分の速度ベクトルが信頼性の低い部分へと伝播して行き、2つの物体が重なっている部分で速度ベクトルは鋭く (sharp に) 変化する。

5. 時空間近傍を用いた推定誤差の補正

入力画像列 $I(x, y, t)$, $t=1, \dots, N$ から速度地図列 $V(x, y, t)$, $t=2, \dots, N$ を得ることができた。しかし、これらの速度ベクトルは局所的な計算により求めたので、もちろん雑音により生じる誤差がある。これらの推定誤差を軽減するために、おでに求めた速度ベクトルの地図から時空間的近傍の情報を用いて、各点の速度ベクトルをくり返し法を用いて求めなおす。

ここで物体は剛体で画面間で滑らかに動くと仮定すると、式(5)は空間的かつ時間的な近傍に対しても成立する。したがって、 t 番目の画面の点 (x, y) の速度ベクトルを補正するのに、式(5)を再び利用できる。ただし、この場合は、 $(\overline{v_x}, \overline{v_y})$ を計算するのに、 $(t-1)$ 番目、 t 番目、 $(t+1)$ 番目の画面の 5×5 点の近傍の速度ベクトルの平均を利用する (したがって、合計で $5 \times 5 \times 3 = 75$ 点)。平均

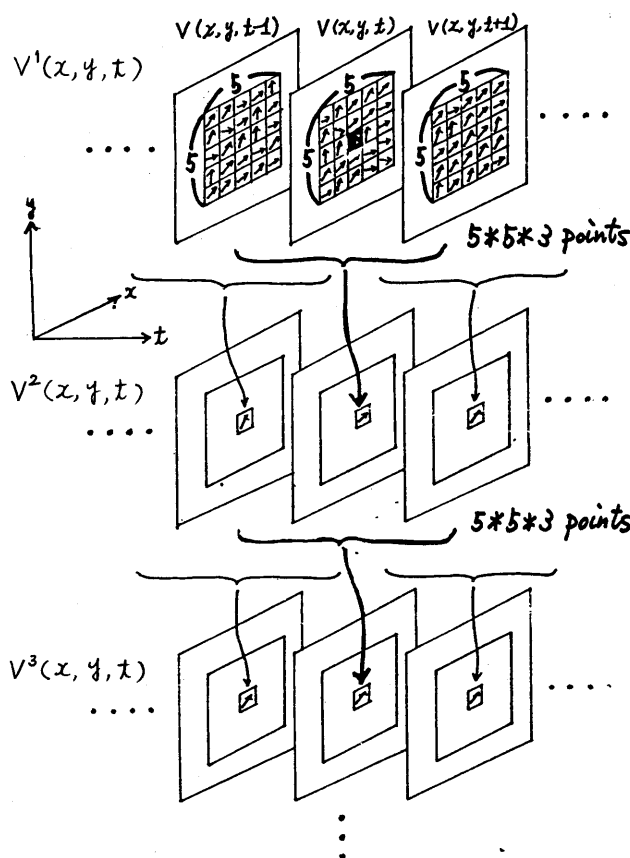
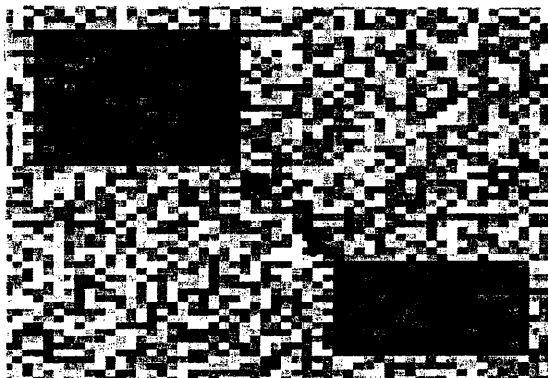


図7. 時空間的近傍を用いた速度ベクトルの補正

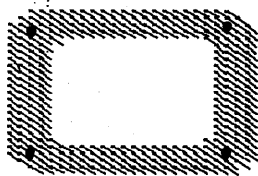
の方法としては、当然4.2で述べた重み付け平均を用いる。以上の方法を速度地図列 $V^1(x, y, t), t=2 \cdots N$ の全画面の各点に対して適用することにより、新しい速度地図列 $V^2(x, y, t), t=2, \cdots, N$ を得る(図7参照)。この新しい速度地図列を入力として、同じ操作をくり返すことにより再び新しい速度地図列 $V^3(x, y, t), t=2, \cdots, N$ を得る。この操作を行っても速度地図列に変化が起こらなくなるまで、この操作をくり返す。

6. 実験結果と考察

以上述べた手法を評価するためにいくつかの実験を行なった。まず測定誤差の程度を知るために、図8(a)のような合成画像を図に示す方向に画面ごとに1絵素ずつ動かした。画像の大き



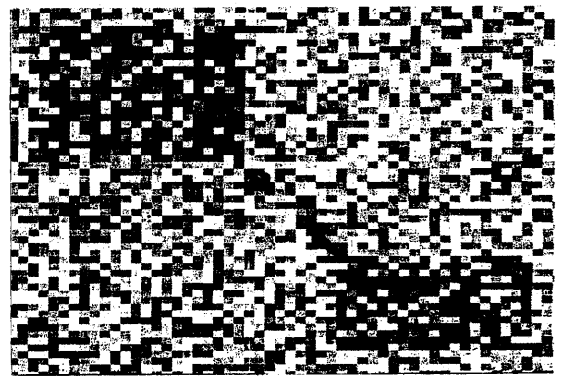
(a)



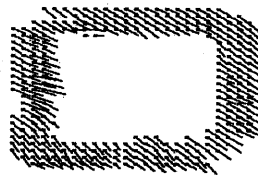
(b)

図8. 伝播法による速度ベクトルの決定
(雑音 20%)

さは 64×64 点で各点の明るさは8ビットに量子化している。また、画像には、20%のランダムな雑音を加えている。この画像列(全体で12画面)の内、最初の2画面から伝播法を用いて速度ベクトルを求めた結果を同図(b)に示す。図中、黒丸は速度ベクトルを求めるのに用いた既知の特徴点の位置を示す。また、図9、図10には、それぞれ雑音を60%、90%を加えた場合の結果を示す。これらの結果から20%程度の雑音ではかなり精度よく、60%程度の雑音では誤った推定をする部分が生じ、90%程度の雑音になると不正確になるといことが解る。2つの物体が重なって見える時の結果を調べるために図11のような画像に対して同様の実験を行なった結果を同図

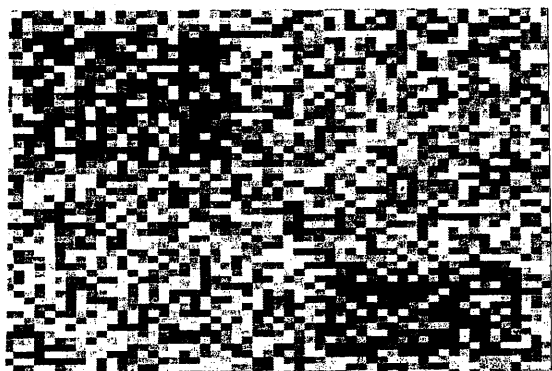


(a)



(b)

図9. 伝播法による速度ベクトルの決定
(雑音 60%)



(a)

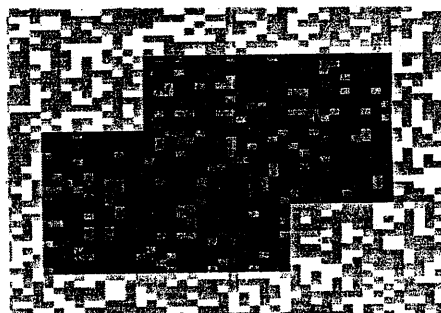


(b)

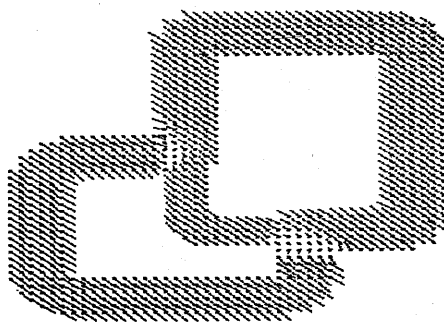
図10. 伝播法による速度ベクトルの決定
(雑音 90%)

(b) に示す。これから解るように、2つの物体が重なっている部分でのみ速度ベクトルの推定誤差は生じており、誤ったベクトルが広い部分には伝播していないことが解る。

最初の2画面から速度地図が決定されると、以降の画面では前画面での処理結果から予測地図を作成し、それを用いて現画面の速度地図を作成する。図12は図9からこうして得られた第3画面と第10画面の結果を示す。これから解るように、速度ベクトルは比較的精度良く求まる。速度地図が画像列に対して求まると時空間的近傍の情報をを用いて速度ベクトルの補正を行なう。



(a)

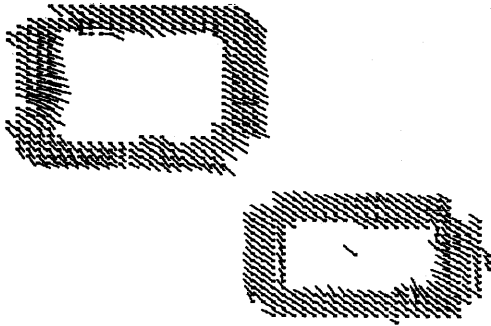


(b)

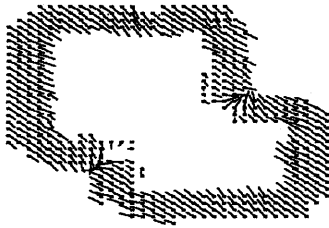
図11. 伝播法による速度ベクトルの決定
2つの物体が重なっている場合
(雑音 20%)

図13は図12に対して1回のくり返し後の結果を示す。

実物体に対する結果を図14に示す。伝播法による最初の速度ベクトルの決定では(b), ①に示す静止物体と重なっている所と②に示すタイヤの部分での誤差が大きい。①の部分は明るさが一様で、しかもエッジの方向と車の進行方向が同一のため対応をとるのが非常に困難(局所的に見ると静止しているように見える)であり、静止物体からの影響を受けている。②の部分はタイヤが黒く一様なため対応付けの困難な所である。しかし、くり返しによる速度ベクトルの補正を行うと(d), これらの部分の速度ベクトルも正しく求まってくる。

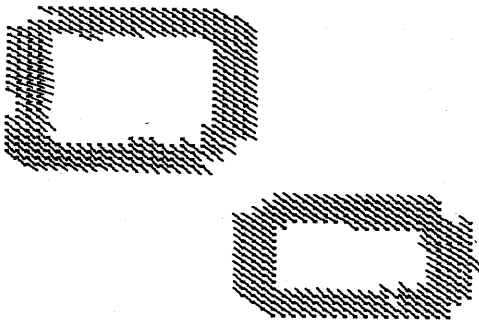


(a) 第3画面

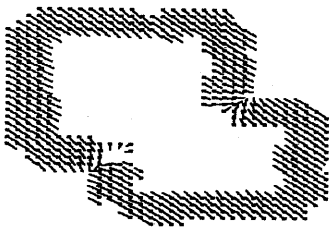


(b) 第10画面

図12. プランを用いた速度ベクトルの決定
(雑音 60%)



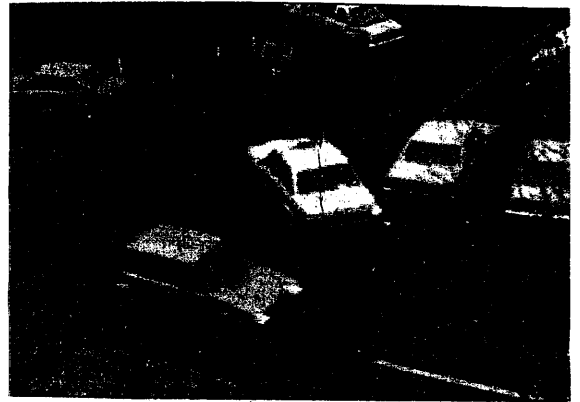
(a) 第3画面



(b) 第10画面

図13. 時空間的近傍を用いた速度ベクトルの補正 (雑音 60%)

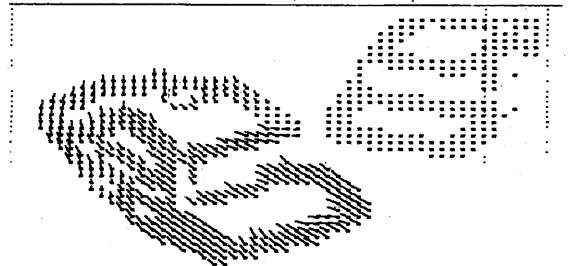
1回のくり返し後の結果



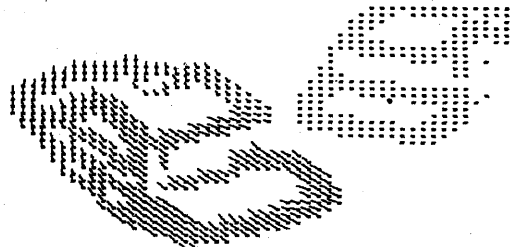
(a) コーナーを曲がる車のシーン (第4画面)



(b) 伝播法による速度ベクトルの決定
(最初の2画面から決定された第2画面目)



(c) プランを用いた速度ベクトルの決定 (第6画面)



(d) くり返し法による速度ベクトルの補正 (第6画面)
6回のくり返し後の結果

図14. 実物体に対する実験結果

7. むすび

動画像から各点の見かけの移動ベクトルを求める手法について述べた。その手法として、①最初の2画面からまず画像上の各点の速度ベクトルを推定する、②一旦、速度ベクトルが得られると前画面での処理結果から次画面の状態を予測して、次画面を能率良く処理して行く、③全体の速度ベクトルが決定されると、その情報を利用して、速度ベクトルをくり返し法を用いて補正するという3つのレベルに分割した。ここでは、時空間的近傍を用いてくり返し法により速度ベクトルの補正を行った。しかし速度ベクトルにあまり変化がない場合には、画像に雑音があっても②の処理で後の画面ではかなり精度良く速度ベクトルが決定できる。したがって、②の処理と同様のことを、最後の画面から時間軸方向を逆に行なうことにより、画像列全体に渡って速度ベクトルを精度良く求めることができる。このため、上で述べた場合に対しては、計算時間のかかる③の処理は不要であると思われる。

明るさの一樣な部分では人間でも局所的に対応をとることは困難である。このため、本論文では、ある程度明るさの変化している部分(縁点など)についてのみ速度ベクトルを求めた。このような縁点の部分は画像の特徴を示す重要な部分であり、この部分の速度ベクトルが得られれば十分であると思われる。しかし、明るさの一樣な部分の速度ベクトルも必要な場合には、縁点の速度ベクトルから内挿によりそれらの点の速度ベクトルを推定することができる。

ここで求めた速度ベクトルは、あくまでも画像上での見かけの速度ベクトルであって、元の3次元世界における速度ベクトルではない。しかし、Ullmann⁽¹²⁾を初めとする最近の“Structure

from motion”の理論^(13,14)を用いれば、本論文で述べた方法により速度地図から密な点の対応がとれるので、元の3次元世界での詳細な形の発見と各点の3次元動的な動きの決定を行うことも可能である。

ここでは、①物体の剛体性と②画面のサンプル速度に対して物体の動きは比較的小さいという2つの仮定をふいた。前者の仮定は大半の物体に対して成立するが、後者の仮定は撮映条件により成立しない場合もある。この仮定は式(1)から束縛されているため、これを取除くためには基本方程式である式(1)を使わない手法が必要となり、将来に残された課題である。また、ここでは画像から直接得られる縁点という画像の特徴に基づいて対応をとるというアプローチをとったが、Tennenbaumらの言う“シーンに固有の特徴”に基づいて対応をとった方がより確実な対応がとれる可能性があり、これも将来に残された課題である。

謝辞

本研究は筆者がHamburg大学に滞在中に行なった。同大学に招いて頂き、また本研究に対して多くの示唆と助言を頂いたNagel教授に深謝致します。また色々と討論して頂いたNeumann, Radig両氏、並びに同大学の画像処理装置等の使い方を親切に教えて頂いたNeumann, Dreschler, Westphal氏ら、また本研究を援助して頂いた西独Humboldt財団に感謝致します。また本研究をまとめるに当たり、本学辻教授から色々の示唆と助言を頂いたことを深謝致します。

参考文献

- (1) W. B. Thompson, "Combining motion and contract for segmentation", IEEE-PAMI-2, no.6, pp. 543-549, 1980.

- (2) H. P. Moravec, "Towards automatic visual obstacle avoidance", IJCAI-79, pp.598-600, 1979.
- (3) S. T. Barnard and W. B. Thompson, "Disparity analysis of images", IEEE-PAMI-2, pp.334-340, 1980.
- (4) L. Dreschler and H.-H. Nagel, "Volumetric model and 3D-trajectory of a moving car driven from monocular TV-frame sequences of a street scene", IJCAI-81, pp.692-699, 1981.
- (5) C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images", IEEE-IT-22, pp.573-579, 1976.
- (6) C. L. Fennema and W. P. Thompson, "Velocity determination in scenes containing several moving objects", Computer Graphics and Image Processing, 9, pp.301-315, 1979.
- (7) B. K. P. Horn and B. G. Schunck, "Determining optical flow", AI-Memo-572, Artificial Intelligence Laboratory, MIT, Cambridge/MA, 1980.
- (8) M. Yachida, "Determining Velocity map by 3-D iterative estimation", IJCAI-81, pp.716-718, 1981.
- (9) 谷内田, "動画像の理解", 第10回画像工学コンファレンス, pp.127-132, 1979.
- (10) M. Yachida, M. Ikeda, S. Tsuji, "Plan-guided analysis of cineangiograms for measurement of dynamic behavior of heart wall", IEEE Trans. PAMI-2, pp.516-521, 1980.
- (11) M. Yachida, M. Asada, S. Tsuji, "Automatic analysis of moving images", IEEE Trans. PAMI-3, pp.12-19, 1981.
- (12) S. Ullmann, "The interpretation of visual motion", MIT Press, Cambridge/MA, 1979.
- (13) M. Asada, M. Yachida, S. Tsuji, "Three dimensional motion interpretation for the sequence of line drawings", Proc. ICPR-5, pp.1266-1274, 1980.
- (14) H.-H. Nagel, B. Neumann, "On 3D reconstruction from two perspective views", IJCAI-81, pp.661-663, 1981.