I-41 2次元ワープ法を用いた動画像中の動きベクトルの抽出

Extraction of The Motion Vector in The Motion Picture Using The Two-Dimensional Warping

斎藤 敦子† 鈴木 基之‡ 伊藤 彰則‡ 牧野 正三‡
Atsuko Saito Motoyuki Suzuki Akinori Ito Shozo Makino

1. はじめに

動画像中の動きを求める場合，動画像そのものを特徴量とする方法と動画像から動きのみを特徴量として抽出する方法が考えられる．動画像そのものを特徴量とした場合は，動作位置が違うと同じ動きでも違う特徴量となる．しかし，たとえばジェスチャ認識においてジェスチャ部位の動きを求めるとき，同じジェスチャならば動画像中のジェスチャ部位が映っている位置に関係なく同じ動きと判断できる特徴量を用いることが望ましい．そこで本稿では，動画像中における移動物体の位置などの制約を受けない特徴量として動きのみを抽出した動きベクトルについて検討する．

動きベクトルの抽出手法として，天田ら[2]は単調連続2次元ワープ法[1](以下，2次元ワープ法)を用いた方法を提案している．動画像中の物体は少しずつ変形しながら動くことが考えられる．2次元ワープ法は変形に適応可能な手法であるため，動きベクトルの抽出に用いることは有効であると考えられる．しかし，2次元ワープ法は膨大な計算量が必要となる．動きベクトルの抽出を実時間で行うことを考えた場合，計算時間の短縮が必要となる．そこで，本稿では画素数の削減，また，1画素あたりの計算量の削減により高速化を図る．また，オプティカルフローで求めた動きベクトルと抽出精度の比較をし，本手法の有効性を示す．

2. 動きベクトル抽出法

単調連続2次元ワープ法[1]とは2画像間の最大一致を実現する画素間のマッピング法である．隣接する2画素がワープ後も上下左右関係と近傍関係を保つという単調連続条件を制約として用いることで2次元的自由度をもちながらトポロジーも保存したワープを求めることができる．動画像中の時間的に連続した2画像間に2次元ワープ法を用いることで，画素ごとの対応関係から動きベクトルが抽出できる．しかし動画像の場合，物体が移動することで移動物体と背景の境界でトポロジーが崩れる．このような場合も2次元ワープ法はトポロジーを保存しようとするため，うまく対応がとれない場合がある．そこで，前処理として背景消去を行うことで，各画素を移動物体と背景に分離し，トポロジーの保存をコントロールする．背景消去には画素の時間方向の差分情報を用い，時間方向の平均画像との差分がある閾値以下の画素を背景とする．そして画像の左下から対応点探索をし，既に探索済みの左と下の隣接画素が背景か移動物体かによって式(1)～(4)のように探索範囲を変更する[2]．背景と移動物体との境界(エッジ)で探索範囲を広げることで，本手法は移動量の多い画像にも対応できる．

† 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku Univ.
‡ 東北大学大学院工学研究科
Graduate School of Engineering, Tohoku Univ.

左の隣接画素 $(i-1,j)$ において，

・ $(i-1,j)$ が移動物体のとき

$$0 \le x(i,j)-x(i-1,j) \le 2, \ |y(i,j)-y(i-1,j)| \le 1 \quad (1)$$

・ $(i-1,j)$ が背景のとき

$$|x(i,j)-x(i-1,j)| \le W, \ |y(i,j)-y(i-1,j)| \le W \quad (2)$$

注目画素の下の隣接画素 $(i,j-1)$ において，

・ $(i,j-1)$ が移動物体のとき

$$|x(i,j)-x(i,j-1)| \le 1, \ 0 \le y(i,j)-y(i,j-1) \le 2 \quad (3)$$

・ $(i,j-1)$ が背景のとき

$$|x(i,j)-x(i,j-1)| \le W, \ |y(i,j)-y(i,j-1)| \le W \quad (4)$$

ただし，ある点 (i,j) の対応点を $(x(i,j),y(i,j))$ とし，また，式(1)～(4)は注目画素が移動物体の場合とする．[2]では注目画素が背景の場合も[1]と同じ探索範囲で探索を行っている．しかし，背景の点は動きベクトルが0であると考えられるため，本手法では注目画素が背景の場合，探索は行わない．これにより探索が必要となる画素数を大幅に削減し，計算時間を短縮することができる．また，本手法でも[1][2]と同様に，各画素の探索ごとにコストが最小となる候補点を R 個残すという枝刈を行っている．

また，注目画素が移動物体，左と下の隣接画素が共に背景である場合，式(2)(4)より探索範囲内の画素数は $2W^2+1$ となる．しかし，探索範囲内には注目画素とは対応するはずのない背景や別の物体の画素も多く含まれていると考えられる．そのような画素とのコスト計算は省いても問題はない．そこで，注目画素と色の近い画素が対応点すると考え，探索範囲内の画素において注目画素とのRGB空間上での距離(式(6))が最小の R_{rgb} 個を残すという枝刈を行い探索を行う画素を絞り込む。これにより，1画素あたりの探索にかかる計算量を大幅に減らすことができる．

コストの計算には画素の色情報や変形具合から求められる値にそれぞれ重みを加えて足し合わせた値を用いる[2]．

$$dist(i,j,x(i,j),y(i,j)) = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3 \quad (5)$$

P_1 は (i,j) のRGB値 (r_t,g_t,b_t) と $(x(i,j),y(i,j))$ のRGB値 $(r_{t+1},g_{t+1},b_{t+1})$ におけるRGB空間上での距離である．

$$P_1 = \sqrt{(r_{t+1}-r_t)^2+(g_{t+1}-g_t)^2+(b_{t+1}-b_t)^2} \quad (6)$$

ただし，$r,g,b = \frac{1}{9}\sum_{di=-1}^{1}\sum_{dj=-1}^{1} r,g,b(i+di,j+dj)$

P_2 はワープ後も周辺画素との距離があまり変化しないという仮定に基づく．

$$P_2 = \sqrt{(p_{i,j}^x-1)^2+(p_{i,j}^y)^2}+\sqrt{(q_{i,j}^x)^2+(q_{i,j}^y-1)^2} \quad (7)$$

ただし，$p_{i,j}^x = x(i,j)-x(i-1,j)$，$p_{i,j}^y = y(i,j)-y(i-1,j)$

$q_{i,j}^x = x(i,j)-x(i,j-1)$，$q_{i,j}^y = y(i,j)-y(i,j-1)$

P_3 は周辺画素で得られる動きベクトルが注目画素における動きベクトルとあまり変わらないという仮定に基づく．

$$P_3 = \left| D_{i,j} - (D_{i-1,j}+D_{i-1,j-1}+D_{i,j-1}+D_{i+1,j-1})/4 \right| \quad (8)$$

ただし，$D_{i,j} = (x(i,j),y(i,j))-(i,j)$

3. 実験結果

本手法の動きベクトル抽出精度を調べるために実験を行った．実験には，背景{なし(平坦)，あり(他の物体が映っている)}の 2 パターン，移動物体のテクスチャ{なし，あり 2 種類}の 3 パターンとした計 6 パターンの動画像(画像サイズ 192×128[pixel]，24bit カラー画像)を用いた．その例を図 1 に示す．定量的な評価をするために目視によって正解の動きベクトルを与え，以下の式を用いて抽出精度を評価した．

$$\varepsilon = \frac{1}{n}\sum_{(i,j)\in M}\sqrt{\{(x(i,j)-\bar{x}(i,j)\}^2+\{(y(i,j)-\bar{y}(i,j)\}^2} \quad (9)$$

$$\mathrm{E} = \frac{1}{XY}\sum_{i=0}^{X-1}\sum_{j=0}^{Y-1}\sqrt{\{(x(i,j)-\bar{x}(i,j)\}^2+\{(y(i,j)-\bar{y}(i,j)\}^2} \quad (10)$$

ただし，M は移動物体の点の集合，n は移動物体の点の数，$(\bar{x}(i,j),\bar{y}(i,j))$ は正解の対応点，X,Y は画像の幅および高さとする．式(9)は移動物体の画素の結果のみを正解の動きベクトルと比較した抽出誤差，式(10)は全画素に対して正解の動きベクトルと比較した抽出誤差である．比較対象として勾配法[3]およびブロックマッチング法[4]を用いた．なお，本手法のパラメータは，枝刈の幅 R=100，R_{rgb}=100，物体のエッジの探索幅 W=25，コストの重み α_1=1.0，α_2=3.0，α_3=2.0，勾配法は反復回数 100，α=10，ブロックマッチング法はブロックサイズ 8×8，探索範囲 W=25 とした．

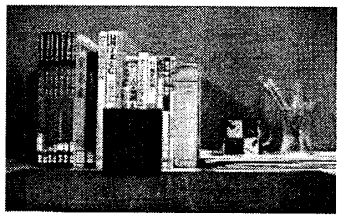

図 1　実験に使用した画像の例
（背景あり，中央の四角が移動物体(テクスチャなし)）

実験結果を図 2,3 に示す．monotone はテクスチャなしの単色物体，texture1,2 はテクスチャのある物体である．また，本実験での 1 フレームあたりの平均計算時間を表 1 に示す．

図 2 より，本手法はいずれの画像においても勾配法よりも誤差が小さくなった．ブロックマッチング法と比較した場合は，誤差はほぼ同程度もしくはそれ以下となった．単色物体の内部のように輝度値の変化がない平坦な部分に関して，勾配法は動きベクトルを求めることができない．またブロックマッチング法はそのような部分でノイズの影響を受けやすくなる．しかし，本手法は物体内部でトポロジーを保存しながら対応点を求めるので，物体のエッジで正しい対応点を求めることができれば単色物体の内部でもそれに追従する形で動きベクトルを求めることができる．しかし，背景消去によって背景と移動物体を正しく分離できず移動物体のエッジ付近の背景部分が誤って移動物体とされてしまった場合や，エッジにおいて誤った対応点のみが枝刈後に残ってしまった場合でもトポロジーを保存しながら対応点探索を行うため，エッジの誤対応の影響で物体内部も誤った動きベクトルが抽出されることがある．しかし，目視によって背景消去を行ったり，枝刈の幅を広げることによって正しい動きベクトルを求めることは可能である．

図 3 より，画像全体で正解の動きベクトルと比較を行った場合，本手法はいずれの方法よりも誤差が小さくなった．

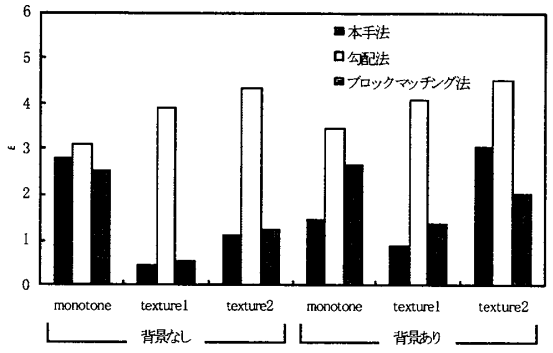

図 2　動きベクトル抽出誤差 ε

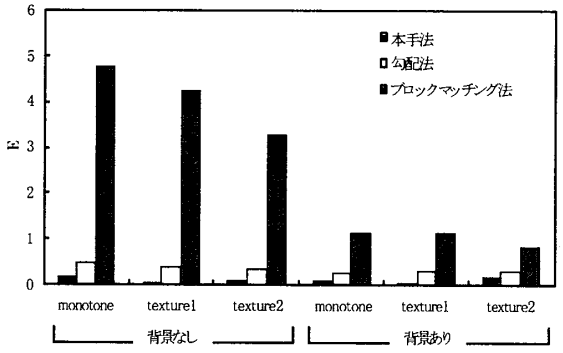

図 3　動きベクトル抽出誤差 E

本手法は背景消去によって背景と判断された部分においては動きベクトルが 0 であると仮定しているため，背景消去が正しく行われれば静止部分の動きベクトルは全て 0 となる．しかし，勾配法およびブロックマッチング法ではそのような静止部分においてもノイズの影響などで 0 ではない動きベクトルが出てしまう場合がある．特に，ブロックマッチング法では前述したように輝度が平坦な部分では照明などのノイズの影響を受けやすいため，背景部分が平坦な場合，特に誤差が大きくなっている．

表 1　平均計算時間（CPU ： Alpha 21264 750MHz）

	本手法	勾配法	ブロックマッチング
平均計算時間 [sec/frame]	1.85	0.78	7.33

4. まとめ

オプティカルフローと動きベクトルの抽出精度の比較を行うことで，本手法の有効性を示すことができた．しかし，背景消去などの問題から誤った動きベクトルが抽出される場合もあり，さらなる抽出精度の向上が今後の課題である．

参考文献

[1] 内田，迫江「動的計画法に基づく単調連続 2 次元ワープ法の検討」信学論，vol.J81-D-Ⅱ，No.6, pp-1251-1258, June 1998

[2] 天田，鈴木，後藤，牧野「動作位置の変化に頑健なジェスチャ認識」信学技報，PRMU99-106, Nov. 1999

[3] B.K.P.Horn and B.G.Schunck「Determining Optical Flow」Artif. Intell., vol.17, pp.185-203, 1981

[4] 安居院，長尾「画像の処理と認識」昭晃堂，1992