

ニュートン補間法によるベクトル予測を用いた 動きベクトル検出処理の高速化手法

佐藤章浩[†] 尾本良子[†] 佐田宏史[†]
前川仁孝[†] 伊與田光宏[†]

本稿では、2 フレーム間の対応を求める動きベクトル検出処理において、ニュートン後退補間法を用いることにより SSDA 法の演算量削減効果を向上させる動きベクトル予測手法を提案する。従来の動きベクトル予測手法では、周辺ブロックで求められた動きベクトルから現ブロックの動きベクトルを予測する。そのため、急激な動きが現れた場合などに予測精度が低下し、SSDA 法による演算量削減効果が十分に得られないことがある。そこで提案手法では、高精度なベクトル予測を行うために過去フレームおよび現フレームで求められた動きベクトル値を補間点としたニュートン後退補間法により、次フレームの動きベクトルを予測する。このように、過去の動きの傾向を考慮したベクトル予測に基づく探索により早期に誤差の小さいブロックが探索され、SSDA 法における誤差和計算の打ち切り効率を向上させることができる。最後に、動きベクトル検出処理に要した処理時間について周辺ブロックから現ブロックの動きベクトルを予測する手法と比較した結果、テストシーケンスである“Flower Garden”において、提案手法により約 73.8% の処理時間の削減を確認することができた。

Fast Motion Estimation Algorithm Using Vector Prediction by Newton Interpolation

AKIHIRO SATO,[†] YOSHIKO OMOTO,[†] HIROSHI SATA,[†]
YOSHITAKA MAEKAWA[†] and MITSUHIRO IYODA[†]

This paper proposes an efficient vector prediction technique by using a newton interpolation with sequential similarity detection algorithm (SSDA) in motion estimation. A conventional technique predicts motion vectors from around blocks, thus it decreases prediction accuracy in a rapid acceleration. Therefore, the proposed technique predicts motion vectors of a next frame by a newton interpolation using motion vectors of before frames and current frame. Consequently, the proposed technique is able to improve efficiency of a calculation termination in SSDA. Finally, the proposed technique is compared to a conventional technique. As a results, the proposed technique is about 73.8 percents faster than a conventional technique in processing time of motion estimation with “Flower Garden” sequence.

1. はじめに

動きベクトル検出処理は、動画像圧縮、パターン認識などの様々な応用分野において重要な処理である。動きベクトル検出処理アルゴリズムの一つであるブロックマッチング法は、探索方法に全ブロックの動きベクトルを求める全探索法を用いる場合、精度の良い動きベクトルを検出することができる。しかし、フレーム全体に対して探索を繰り返すことにより膨大な演算量を要するという問題点がある。このため、従来よりブロッ

クマッチング法に対する様々な高速化手法が用いられている。

ブロックマッチング法の高速化手法は、大きく分けて2つに分類することができる。一つは、探索するブロック数を間引き、演算量を削減することにより高速化する手法¹⁾²⁾³⁾⁴⁾⁵⁾⁶⁾⁷⁾である。これらの手法は、探索するブロック数を減らすことにより演算量を削減するため、動きベクトルの検出精度は全探索法より低下する場合がある。もう一つは、全探索法の精度を保ちながらマッチング計算の演算量を削減することにより高速化する手法である。これらの手法は、演算量の削減効果は前者よりも低下するが、全探索法の精度を保証することが可能である。後者の代表的な手法である逐次残差検定法 (sequential similarity detection algorithm:以下

[†] 千葉工業大学 情報科学部 情報工学科
Department of Computer Science, Faculty of Information and Computer Science, Chiba Institute of Technology

SSDA 法⁸⁾⁹⁾は、閾値を設定することでマッチング処理時の誤差和計算を打ち切る手法であり、全探索法の精度を保証したまま演算量の削減が可能である。しかし SSDA 法では、探索範囲内の探索順番などによって良い閾値に更新するタイミングが変動するため、マッチング処理の演算量が変化しやすく、処理対象となるフレームによっては誤差和演算の打ち切り効率が低下することがある。そこで、探索領域内を渦巻き状に探索することで SSDA 法の誤差和計算の打ち切り効率を向上させる手法¹⁰⁾や、テンプレートに含まれる部分的なパターンの原画像における出現確率に基づいて照合順序を決定した後 SSDA 法を適用することにより誤差和演算の打ち切り効率を向上させる手法¹¹⁾、最小誤差を用いた単純な一次関数を閾値とすることで閾値の算出コストを抑える手法¹²⁾など、従来より様々な手法が提案されてきた。これらの手法に加え、誤差和演算の打ち切り効率を向上させるために、既に計算済みである周辺ブロックの動きベクトル値から現ブロックの動きベクトルを予測し、予測地点から探索を行うことで SSDA 法における誤差和計算の打ち切りを効率良く行う手法¹³⁾が提案されている。しかし周辺ベクトルを用いたベクトル予測では、急激な動きの変化を示すフレームや同領域に複数の動物体が存在するフレームなどで、周辺ブロックとの相関性が低下することがある。このような場合、動きベクトルの予測精度が低下し、SSDA 法による演算量削減効果が十分に得られない可能性がある。

そこで本稿では、従来のベクトル予測において予測精度が低下するような場合でも、高精度な動きベクトルの予測が可能となるベクトル予測手法を提案する。本手法は、過去数フレームおよび現フレームで求められた動きベクトルを補間点としたニュートン後退補間法により、次フレームの動きベクトルを予測する。これにより、SSDA 法における誤差和計算の打ち切り効率が向上し、動きベクトル検出処理を高速化できる。

2. ブロックマッチング法による動きベクトル検出処理

動きベクトル検出処理には、多様なアルゴリズムが存在する。中でも、一般に用いられているアルゴリズムの一つにブロックマッチング法がある。ブロックマッチング法では、図 1 のように動画像中の 1 つ前にあたるフレーム (前フレーム) 中のブロックをテンプレートとして、現時刻のフレーム (現フレーム) を探索することで両フレーム間の対応を求める。対応の尺度には、前フレーム f におけるテンプレートと現フレ

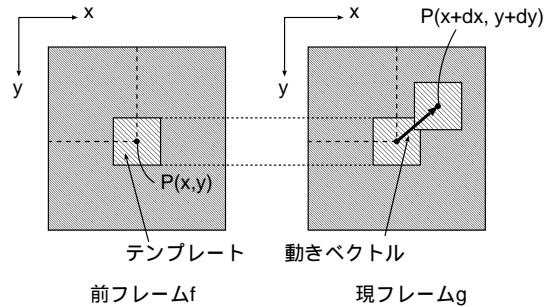


図 1 ブロックマッチング法による動きベクトル検出処理
Fig. 1 Motion Estimation Algorithm by Block Matching Method.

ーム g における注目ブロックとの評価関数を用いる。評価関数値が最小または最大となるブロックを対応点とし、求められた対応点への方向と移動量が動きベクトルとなる。評価関数値を求めるには、式 (1)、式 (2) で求められる差分絶対値和 (D_{SAD}) や二乗誤差和 (D_{SSD}) などが一般に用いられる。ブロックマッチング法における探索方法の中で最も基本的な方法である全探索法では、輝度値が急激な変化を示す部分でも求まる動きベクトルの誤差が少なくなるため、雑音に強いという利点がある。しかし、フレーム全体に対して対応点探索を繰り返すことで計算量が膨大になるため、処理の高速化が求められている。

$$D_{SAD} = \sum_i \sum_j |f(i, j) - g(i, j)| \quad (1)$$

$$D_{SSD} = \sum_i \sum_j \{f(i, j) - g(i, j)\}^2 \quad (2)$$

3. SSDA 法によるブロックマッチング法の高速化

ブロックマッチング法の高速化手法の一つに、全探索法によって求まる動きベクトルの精度を保証したまま高速化が可能な SSDA 法がある。SSDA 法では、マッチング処理の際に累積誤差和がある閾値を超えた場合、その時点で現テンプレート位置におけるブロックの類似度は低いと判断し、以降の誤差和計算を打ち切りテンプレートを移動する。以降のテンプレート位置においても同様に打ち切り判定を行うことで、動きベクトル検出処理全体の演算量を削減することができる⁸⁾。打ち切り判定の際の閾値には、探索済みブロックの誤差和の最小値を用いることで、対応点として求まるブロックは誤差が必ず最小値となる。このため、全探索法によって求まる動きベクトルの精度を保証することができる⁹⁾。

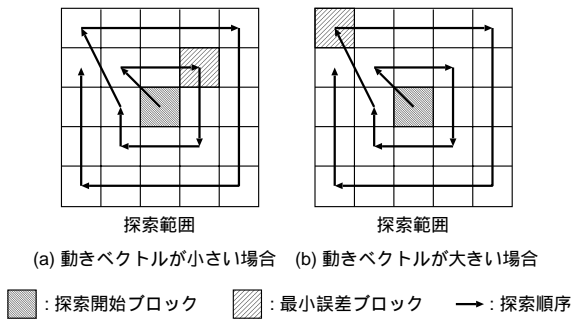


図 2 渦巻き探索における探索順序
Fig. 2 Search Order of Spiral Search.

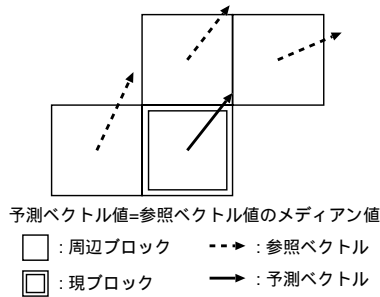


図 3 周辺ブロックの動きベクトルを用いたベクトル予測
Fig. 3 Vector Prediction Using Motion Vectors of Around Blocks.

SSDA 法は、探索する順序と最小誤差ブロックの位置によって閾値が最小値に更新されるタイミングが変動するため、マッチング処理の演算回数が増加しやすい。このため、効率良く計算を打ち切るためには、最小誤差ブロックへの探索回数を削減することが重要である。つまり、探索の初期の段階で閾値を最小値に更新することができれば、以降の探索における誤差和計算の打ち切り効率が大幅に向上する。そこで誤差和計算を効率良く打ち切るために、動画では最小誤差ブロックがテンプレートの中心位置付近に存在する確率が高いことを利用する。最小誤差ブロックを探索の初期段階で探索するようにするために、図 2 のように探索領域内の中心から外側へ向かって渦巻き状に探索する渦巻き探索を用いる。これにより、探索の早期に最小閾値に更新することができる¹⁰⁾。しかし、動きベクトルが大きい場合には、探索開始点から最小誤差ブロックまでの探索回数が増加し、最小閾値への更新が遅くなる。このため、動きベクトルが小さい場合に比べて誤差和計算の打ち切り効率が低下する。

渦巻き探索において、動きが大きい場合に誤差和計算の打ち切り効率が低下する問題を改善するために、

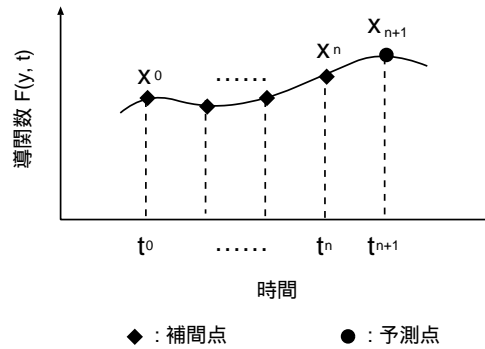


図 4 ニュートン後退補間法による多項式補間
Fig. 4 Newton Interpolation Polynomial.

フレーム内の空間的相関性を利用する手法が用いられる¹³⁾。本手法では、既に計算済みである周辺ブロックにおける動きベクトルから現ブロックの動きベクトル値を予測し、予測ベクトルの終点を開始地点として探索する。これにより、早期に最小誤差ブロックを探索することが可能となり、SSDA 法における誤差和計算の打ち切り効率が向上する。予測ベクトル値は、図 3 に示すように現ブロックの左、上、右上の 3 ブロックのメディアン値を用いる。

4. ニュートン後退補間法によるベクトル予測手法

従来の周辺ベクトルを用いたベクトル予測では、急激な動きの変化を示すようなフレームや、1 つのフレーム内に複数の動物体が混在するような場合、現ブロック中に存在する動物体とは異なる動物体の動きベクトルを予測ベクトルとする可能性がある。このため、予測に用いられる動きベクトルと現ブロックで求められる動きベクトルとの相関性が低くなり、動きベクトルの予測精度が低下する。これにより、SSDA 法における誤差和計算の打ち切り効率が低下する。

そこで、ベクトル予測の際にニュートン後退補間法を用いることで同領域における過去の動きの傾向から動きベクトルを予測し、周辺ベクトルを用いたベクトル予測において予測精度が低下するような場合でも高精度な予測ベクトルを求めることで、SSDA 法における誤差和計算の打ち切り効率を向上させる。

4.1 ニュートン後退補間法による外挿

既に求められたいくつかの値から未知の値を予測する方法として、多項式補間方法の一つであるニュートン補間法が用いられることがある。本稿では、過去に求められた値から未来の値を外挿する際になどに使われるニュートン後退補間法を用いる。ニュートン後退補間

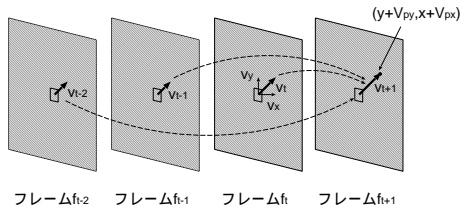


図 5 ニュートン後退補間法によるベクトル予測
Fig. 5 Vector Prediction by Newton Interpolation Polynomial.

法は、図 4 に示すように関数の変化率を用いて式 (3) に示す補間多項式によって点 x_{n+1} における関数の値 $f(x_{n+1})$ の近似値を求める。ここで、 x_0, x_1, \dots, x_n は補間多項式における補間点、 $f(x_0), f(x_1), \dots, f(x_n)$ は関数 $f(x)$ における関数値、 $f[x_n, \dots, x_0]$ は、 $n+1$ 個の点に関する平均変化率である n 階差商である。

$$f(x_{n+1}) = f(x_n) + (x_{n+1} - x_n)F[x_n, x_{n-1}] + \dots + (x_{n+1} - x_1)(x_{n+1} - x_0)F[x_n, x_{n-1}, \dots, x_1, x_0] \quad (3)$$

式 (3) から得られる x の n 次多項式は、 x_0, x_1, \dots, x_n を補間点とする補間多項式であり、 $f(x_{n+1})$ の近似値となる。ニュートン後退補間法は、補間点を追加してもそれまでの補間式が保存されるため、他の補間法に比べ計算量が少ないという利点がある。

4.2 ベクトル予測へのニュートン後退補間法の適用
本手法は図 5 に示すように、過去フレームおよび現フレームで求められた動きベクトルの値を補間点とするニュートン後退補間法により、次フレーム f_{t+1} における動きベクトル V_{t+1} を予測する。 m 個のフレームから V_{t+1} を予測する場合の $m-1$ 次の補間多項式は式 (4) のようになる。

$$V_{t+1} = V_t + (V_{t+1} - V_t)F[V_t, V_{t-1}] + \dots + (V_{t+1} - V_2)(V_{t+1} - V_1)F[V_t, V_{t-1}, \dots, V_{t-n+2}, V_{t-n+1}] \quad (4)$$

式 (4) を用いて、水平軸の予測ベクトル $V_{x_{t+1}}$ 、垂直軸の予測ベクトル $V_{y_{t+1}}$ を求める。これらから得られる V_{t+1} の終点となる $(x + V_{x_{t+1}}, y + V_{y_{t+1}})$ が、渦巻き探索を開始するテンプレート位置となる。このような過去の動きの傾向を考慮したベクトル予測により、従来の周辺ベクトルを用いたベクトル予測において予測精度が低下する場合でも、高精度な動きベクトルを予測することができる。これにより、従来のベクトル予測に基づく探索よりも早い段階で最小誤差ブロックを探索することにより、早い段階で最小閾値へと更新することができ、SSDA 法における誤差和計算の打ち切り効率が向上する。



図 6 “Football” シーケンス
Fig. 6 “Football” Sequence.



図 7 “Flower Garden” シーケンス
Fig. 7 “Flower Garden” Sequence.

ここで、ニュートン後退補間法によるベクトル予測における補間多項式で用いる次数に関して、動きベクトルの予測精度と予測にかかる計算コストの 2 つの面から、適切な次数を設定することが重要である。精度の面では、一般には次数を上げることで精度の高い近似を行うことが可能である。しかし、マッチング処理においてフレーム中のノイズ成分などにより対応点が誤検出された場合などには、その点を補間点とすると補間多項式の解に誤差が生じる。また計算コストの面では、次数の増加により予測のための計算コストが増加し、ニュートン後退補間法によるベクトル予測による演算量削減効果が打ち消されることが考えられる。このことから、ニュートン後退補間法によるベクトル予測における補間多項式の次数を上げることで必ずしも良い結果を得られるとは限らないことが予想される。

5. 評価

ニュートン後退補間法によるベクトル予測の効果を示すために、周辺ベクトルを用いたベクトル予測およびニュートン後退補間法によるベクトル予測における、動きベクトル検出処理に要した演算回数および処理時間、求められた動きベクトルの値の差異について評価する。

評価環境は、CPU は Intel Pentium4 3.4GHz、メモリは DDR SDRAM 2GBytes、OS は Linux 2.6.10 を搭載するワークステーションを用いた。コンパイラ

表 1 “Football” シーケンスにおける総演算回数

Table 1 Total Number of Calculations with “Football” Sequence.

手法	総演算回数 [回]
Around	20287742727
Pro1	14728884677
Pro2	10315155824
Pro3	8530909502
Pro4	7416924645

表 2 “Flower Garden” シーケンスにおける総演算回数

Table 2 Total Number of Calculations with “Football” Sequence.

手法	総演算回数 [回]
Around	16317070288
Pro1	12641622297
Pro2	6619921006
Pro3	5009359672
Pro4	4181146328

は gcc ver.3.3.5 を用い、コンパイルオプションは O4 を使用した。また、入力データには、図 6、図 7 に示すようなテストシーケンスである “Football”、“Flower Garden” を用いた。各シーケンスのフレームサイズは 352×240 画素であり、フレームレートは 30fps である。またブロックマッチング法におけるパラメータは、テンプレートサイズが ±8 画素、探索範囲が ±24 画素である。

5.1 演算量に対する評価

ニュートン後退補間法を用いたベクトル予測による演算量削減の効果を評価するために、表 1、表 2 に動きベクトル検出処理に要した総演算回数を示す。表中の Around は周辺ベクトルを用いたベクトル予測、Pro1、Pro2、Pro3、Pro4 はそれぞれニュートン後退補間法によるベクトル予測における補間多項式の次数が 1、2、3、4 の場合を表している。表 1、表 2 より、ニュートン後退補間法によるベクトル予測によって周辺ベクトルを用いたベクトル予測よりも、演算回数が削減されていることがわかる。“Football” において最大 63.4%、“Flower Garden” において最大 74.4%の演算量削減効果が得られたことが確認できた。これは、過去の動きの傾向がフレームの外側へ向かうような場合に、ニュートン後退補間法を用いたベクトル予測によって予測ベクトルがフレーム外を示したことが考えられる。これにより、対応点が存在しないにも関わらず、マッチング処理を行うという無駄な処理を削減できる。また、急激な輝度値の変化などにより動きベクトルを誤検出した場合、後のフレームで誤検出ベクトルが補間点となり、補間多項式の解に誤差が生じた時も、予測ベクトルがフレーム外を示すことがある。このような場合には、予

表 3 “Football” シーケンスにおけるテンプレート位置あたりの平均演算回数

Table 3 Number of Calculations Average for Each Template Position with “Football” Sequence.

手法	平均演算回数 [回]
Around	139203
Pro1	132406
Pro2	130313
Pro3	127873
Pro4	127591

表 4 “Flower Garden” シーケンスにおけるテンプレート位置あたりの平均演算回数

Table 4 Number of Calculations Average for Each Template Position with “Flower Garden” Sequence.

手法	平均演算回数 [回]
Around	92900
Pro1	91110
Pro2	94246
Pro3	94604
Pro4	93917

測ベクトルがフレーム外を示したブロックは動きベクトルが求まらなくなるため、今後このような場合への対処が必要となる。

また、予測ベクトルがフレーム外を示すことによる演算量削減の効果を除く、SSDA 法による演算の打ち切り効率の向上による効果を示すために、表 3、表 4 に各シーケンス毎のマッチング処理における、各テンプレート位置あたりの平均演算回数を示す。表 3、表 4 より、“Football” ではニュートン後退補間法によるベクトル予測によって周辺ベクトルを用いたベクトル予測よりも演算回数が削減されていることがわかる。また、補間多項式の次数が上がるに伴い演算回数の削減量が増加する。演算回数削減の割合は、最大で次数 4 の時に約 8.3%という結果が得られた。これは、ニュートン後退補間法によるベクトル予測によって高精度な動きベクトルを予測でき、SSDA 法における誤差和計算が効率良く打ち切られたためである。また、“Flower Garden” ではニュートン後退補間法によるベクトル予測における補間多項式の次数が 1 の時に演算回数が約 2.1%削減されているものの、次数が 2 以降の時は周辺ブロックからのベクトル予測に対して演算回数が増加している。次数が 1 の時の演算回数の削減量が “Football” の場合と比べて少ないのは、“Flower Garden” は動きが小さく線形的であり、周辺ベクトルを用いたベクトル予測でも十分な予測精度が得られたため、演算量削減効果の差が小さくなったと考えられる。また、“Flower Garden” はフレーム全体が平行移動するシーケンスであり、フレーム中のノイズ成分によるマッチング時の

表 5 “Football” シーケンスにおける動きベクトル検出処理時間
Table 5 Motion Estimation Processing Time with “Football” Sequence.

手法	処理時間 [s]
Around	400.31
Pro1	293.15
Pro2	205.53
Pro3	170.24
Pro4	158.12

表 6 “Flower Garden” シーケンスにおける動きベクトル検出処理時間

Table 6 Motion Estimation Processing Time with “Flower Garden” Sequence.

手法	時間 [s]
Around	326.03
Pro1	255.14
Pro2	133.72
Pro3	101.43
Pro4	85.48

誤検出ベクトルが生じやすい。このような誤検出ベクトルを後のフレームで補間点とした場合、補間多項式の解に誤差が生じ、誤差和計算の打ち切り効率が低下する。一方、“Football” は動きが部分的に現れるシーケンスであるため、マッチング結果がノイズ成分の影響を受けづらくなることから、正確に物体の動きを補間点とすることができ、高精度な動きベクトルを予測できたと考えられる。

5.2 処理時間に対する評価

ニュートン後退補間法によるベクトル予測による演算量削減に対する動きベクトル検出処理実行時間の削減効果を示すために、表 5、表 6 に各シーケンスにおける動きベクトル検出処理に要した実行時間を示す。表 5、表 6 より、演算回数と同様に処理時間においてもニュートン後退補間法によるベクトル予測によって処理時間が削減されていることがわかる。“Football” において最大 60.5%，“Flower Garden” において最大 73.8%の処理時間削減効果が得られたことが確認できた。

また、SSDA 法による演算の打ち切り効率の向上による効果を示すために、表 7 に “Football”，表 8 に “Flower Garden” における予測ベクトルがフレーム外を示さなかったテンプレート位置、つまりマッチング計算が行われたテンプレート位置の総数およびマッチング計算のみの処理時間、1 つのテンプレート位置における平均処理時間を示す。表 7 より、ニュートン後退補間法によるベクトル予測における処理時間が周辺ベクトルを用いたベクトル予測の処理時間に対して最大で約 7.7%削減されていることが確認できた。一

表 7 “Football” シーケンスにおける総テンプレート数およびマッチング計算のみに要した処理時間

Table 7 Total Number of Templates and Matching Time with “Football” Sequence.

手法	テンプレート数	処理時間 [s]	処理時間/テンプレート [s]
Around	145824	399.34	0.00274
Pro1	111240	292.24	0.00263
Pro2	79157	204.70	0.00259
Pro3	66714	169.44	0.00254
Pro4	61851	156.61	0.00253

表 8 “Flower Garden” シーケンスにおける総テンプレート数およびマッチング計算のみに要した処理時間

Table 8 Total Number of Templates and Matching Time with “Flower Garden” Sequence.

手法	テンプレート数	処理時間 [s]	処理時間/テンプレート [s]
Around	175224	324.83	0.00185
Pro1	138751	254.06	0.00183
Pro2	70241	132.82	0.00189
Pro3	52951	100.56	0.00190
Pro4	44150	83.84	0.00190

表 9 ベクトル予測処理時間と処理全体に対する割合

Table 9 Vector Prediction Processing Time and Percentage of Prediction Processing Time in Total Number of Calculations.

手法	処理時間 [s]	割合 [%]
Around	0.0393	0.00982
Pro1	0.0638	0.02176
Pro2	0.0773	0.03761
Pro3	0.0908	0.05334
Pro4	0.1017	0.06432

方、表 8 においては、最大で約 1.1%削減されていることが確認できた。このことから、演算回数の削減量とほぼ同程度の処理時間が削減されたといえる。

5.3 ベクトル予測に要するコストに対する評価

ニュートン後退補間法によるベクトル予測における動きベクトルを予測する際のコストを評価するために、表 9 に従来の周辺ベクトルを用いたベクトル予測、ニュートン補間法によるベクトル予測における動きベクトルの予測処理に要した処理時間および表 5 に示した全体の処理時間に占めるその割合を示す。表 9 より、全体的にニュートン後退補間法によるベクトル予測に要した処理時間が、周辺ベクトルを用いたベクトル予測に要した処理時間を上回っていることがわかる。このことから、ニュートン後退補間法によるベクトル予測のコストは、補完多項式の次数の大きさによらず周辺ベクトルを用いたベクトル予測よりも大きくなるという結果が得られた。しかし、これらのコストは前節で

示した動きベクトル検出処理全体の処理時間に対して非常に微小なものであり、ニュートン後退補間法によるベクトル予測によるコストの影響は少ないものであるといえる。

5.4 動きベクトルの大きさに対する評価

ニュートン後退補間法を用いたベクトル予測では、ニュートン後退補間法によって求められた予測ベクトルの終点から探索を開始する。ここで、探索範囲は予測ベクトルの終点を探索開始点とした ± 24 画素の範囲に設定している。よって、従来のベクトル予測に基づく探索とは探索領域が異なることがあり、動きベクトル検出処理によって求まる動きベクトルの大きさは周辺ベクトルを用いたベクトル予測と異なる値となることがある。そこで、周辺ベクトルを用いたベクトル予測とニュートン後退補間法によるベクトル予測で求められた動きベクトルの大きさの差異を示すために、図 8、図 9 に各シーケンスにおけるフレーム単位で求められた動きベクトルの大きさを示す。図中のベクトル値は、予測ベクトルがフレーム外を示した場合やマッチング処理によって求められた動きベクトルの大きさがゼロであるブロックを除く、全てのブロックにおいて求められた動きベクトルの大きさの 1 フレームあたりの平均の値である。図 8、図 9 より、周辺ベクトルを用いたベクトル予測とニュートン後退補間法によるベクトル予測により求められた動きベクトルの大きさに差が生じていることがわかる。これは、周辺ベクトルを用いたベクトル予測とニュートン後退補間法によるベクトル予測では、予測ベクトルがフレーム外を示すブロック数が異なるためである。また、次数が 4 になると動きベクトルの大きさの差が大きくなっていることがわかる。これは、過剰に次数を上げることでニュートン後退補間法による近似値の誤差が大きくなったためだと考えられる。“Football” の場合と比べて “Flower Garden” における周辺ベクトルを用いたベクトル予測とニュートン後退補間法を用いたベクトル予測の動きベクトルの大きさの差が大きくなったのは、“Flower Garden” ではニュートン後退補間法によるベクトル予測による予測ベクトルの精度がノイズ成分による動きベクトルの誤検出に影響されやすくなり、予測誤差が比較的大きく現れたためであると考えられる。

6. おわりに

本稿では、動きベクトル検出処理におけるベクトル予測の際にニュートン後退補間法を用いることにより、従来の周辺ベクトルを用いたベクトル予測では予測精度が低下するような場合でも、高精度なベクトル予測

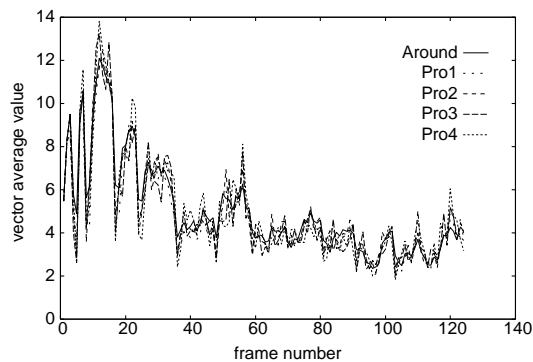


図 8 “football” における動きベクトルの値の差
Fig. 8 Difference of Motion Vector Value with “football”.

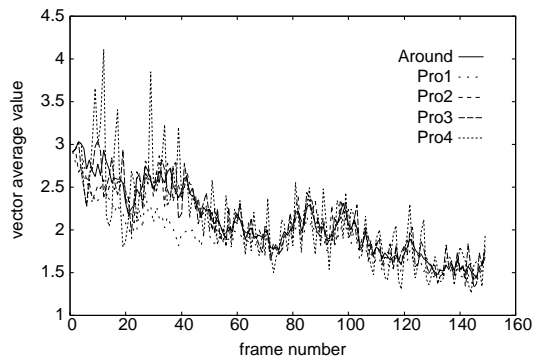


図 9 “flower garden” における動きベクトルの値の差
Fig. 9 Difference of Motion Vector Value with “flower garden”.

により SSDA 法の演算量削減効果を向上させる手法を提案した。

ニュートン後退補間法を用いたベクトル予測を行うことにより、従来の周辺ベクトルを用いたベクトル予測に対して、最大で 73.8% の処理時間削減効果が得られたことが確認できた。ニュートン後退補間法を用いたベクトル予測における効果の傾向は、演算量の削減効果の面では全体的に動きの大きなシーケンスでは補間多項式の次数を上げることで演算量の削減効果が向上するが、動きの小さなシーケンスではフレーム中のノイズ成分による動きベクトルの誤検出の影響により、従来のベクトル予測よりも演算量が増加することがある。また、ニュートン後退補間法を用いたベクトル予測における補間多項式の次数を上げることでベクトル予測に要するコストも増加する。また求まる動きベクトルの精度の面では、過剰に次数を上げることでベクトル値に誤差が生じることがある。よって、ニュートン後退補間法を用いたベクトル予測では処理対象となるシーケンスによって補間多項式の次数を変化させる必要があると考えられる。

今後の課題は、補間多項式の解がフレーム中のノイズ成分の影響に左右されないような安定した補間方法の適用、予測ベクトルがフレーム外を示した場合の対処などが考えられる。

参 考 文 献

- 1) R.Y.Wong and E.L.Hall: Sequential hierarchical scene matching, *IEEE Transactions on Computers*, Vol.C-27, No.4, pp.359-366 (1978).
- 2) S.L.Tanimoto: Template matching in pyramids, *CGIP*, Vol. 16, pp. 356-369 (1981).
- 3) 尾上守夫, 曾根光男: ピラミッド構造の階層間の幾何学的対応に基づいた画像の高速重ね合わせ, *信学論 (D)*, Vol. J70-D, No. 1, pp. 139-147 (1987).
- 4) 富永英義, 小松尚久, 宮下壮史, 花村剛: 階層画素情報を用いた動画像における動き量検出方式, *信学論 (D-II)*, Vol. J72-D-II, No. 3, pp. 395-403 (1989).
- 5) T.Koga, K.Iinuma, A.Hirano and Y.Iijima: Motion compensated interframe coding for video conferencing, in *Proc. Nat. Telecommun. Conf.*, pp. G5.3.1-G5.3.5 (1981).
- 6) R.Li, B.Aeng and M.L.Liou: A new three-step search algorithm for block motion estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, pp. 438-442 (1994).
- 7) J.N.Kim and T.S.Choi: A fast three-step search algorithm with minimum checking points using unimodal error surface assumption, *IEEE Transactions on Consumer Electronics*, Vol.44, No. 3, pp. 638-648 (1998).
- 8) D.I.Barnea and H.F.Silverman: A class of algorithms for fast digital image registration, *IEEE Transactions on Computers*, Vol. 21, No. 2, pp. 179-186 (1972).
- 9) 尾上守夫, 前田紀彦, 斎藤優: 残差逐次検定法による画像の重ね合わせ, *情報処理*, Vol. 17, No. 7, pp. 634-640 (1976).
- 10) TelenorResearch: TMN(H.263)Encoder/Decoder, Version2.0 (1997).
- 11) 嶋好博, 柏岡誠治, 江尻正員: 部分画像の出現確率を用いた高速化パターンマッチング方式, *信学論 (D)*, Vol. 68, No. 2, pp. 161-168 (1985).
- 12) 青木啓史, 仙田裕三: 予測打ち切り法 SSDA に基づく高速ブロックマッチング, *画像符号化シンポジウム資料*, Vol. 16th, pp. 105-106 (2001).
- 13) 森吉達治, 黒田一朗: 動き適応探索順設定による動きベクトル検索の高速化, *信学技報*, Vol. 99, No. 25, pp. 33-40 (1999).