

高速度ビデオでの運動解析プログラムの開発 - OpenMP を用いた相関計算並列化によるトレース高速化 -

横山直樹（東海大学・総合科学技術研究所）

Development of Versatile Motion Analysis Program(VMAP) - Parallel processing of correlation using OpenMP -

Naoki YOKOYAMA (Research Institute of Science and Technology, Tokai University)

キーワード：運動解析、相関計算、高速度ビデオグラフィ
Keywords: Motion analysis, Template Matching, High-speed videography

Motion analysis program for images acquired by high-speed videography was developed and tested so far. In the case of actual analysis scene, particle or object density can be very high. So some scheme of distributed computation will be required for practical use of this program. CORBA was adapted to perform cross-correlation calculation over networkly distributes computers. In this paper evaluation of template size effects in correlation calculation time and parallel processing in one node were discussed.

1. はじめに

前報までで高速度ビデオシステムによって記録された動画を対象とし、対象物体の運動を自動解析するプログラムを開発し、その有効性を実験的に検証してきた。特に時間分解能を高めた高速度ビデオシステムにおいては、空間分解能に制限があり、各フレームにおける対象物体の位置の計測精度が低くなりがちであるが、画像相関値を評価関数とし、その極大値が得られる位置を物体位置として把握するアプローチでは、位置情報をサブピクセルの単位で求めることが可能であること示した。しかし正規化相関を用いたアプローチでは、その計算量が膨大になるために、実用的にはなんらかの計算負荷軽減が必要である。

2. VMAP(Versatile Motion Analysis Program for ultra high-speed videography)

このプログラムの主な特徴は以下のとおりである、

- 島津製作所の 100 万駒/秒超高速ビデオカメラで記録されたファイルに関しては、その内部構造までを把握しており、ヘッダ部分から撮影速度等の記録時の主要なパラメータを読み取る。
- 上記システム以外で得られた動画であっても、Windows の標準的な AVI ファイルであれば解析可能である。この場合は撮影速度等をマニュアルで正しくセットする必要がある。
- 物体の位置把握は、相互相関計算により行われる。位置検出は、サブピクセルの精度で実行される。
- 変位後の物体位置をさがすパターンは、螺旋である。これをスパイラルサーチと呼ぶ。時間分解能が十分な場合は、効率的なサーチが可能である。
- テンプレートを回転させながら相関計算を行うことで、物体の

動きに回転成分が含まれている場合にも、その角度を検出することができる。

- 静止画を扱うことができる。時間的に連続する静止画は、それらを一旦 AVI ファイルに連結してから取り扱うことができる。
- 運動解析結果の軌跡を、元の動画に重ねた形で動画として出力できる。
- マニュアル解析が可能である。
- 各フレームの画像を、ヒストグラムの等値化アルゴリズムにより強調できる。
- ROI の数が多いか、またはその領域が大きいかは、相関計算にかなり時間がかかることが見込まれるが、前者の場合はネットワーク上で複数の計算機を用意することで、分散処理が可能である。(前報まで)
- 相関計算を工夫することで、上記のネットワーク分散処理に加えて単一ノードでの処理の高速化を図ることができる。(本論文)

Fig. 1 にこのプログラムの操作パネルを示す。

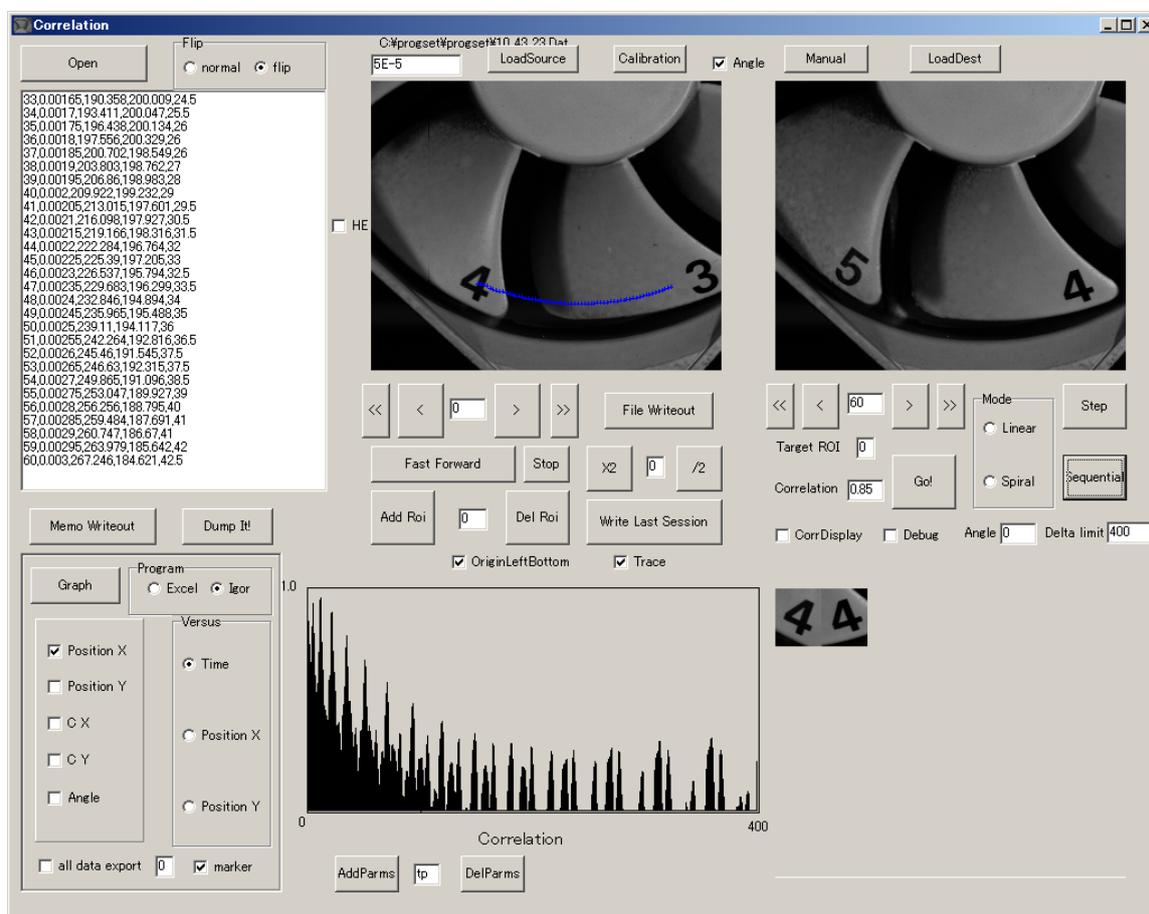


Fig. 1 Operational Panel of VMAP

物体の着目領域である ROI(Region of Interest)は、フレーム毎に相互相関値の極大点を探索することで追跡できる。この場合の相互相関値は、次の式で与えられる。

$$\frac{\sum(f_i - \bar{f}) \times (g_j - \bar{g})}{\sqrt{\sum(f_i - \bar{f})^2} \times \sqrt{\sum(g_i - \bar{g})^2}}$$

ここで f_i と g_j はそれぞれ ROI と探索対象画像上にとったその対応部分上の画素を示す。 \bar{f} と \bar{g} は対象領域にわたって計算された画素の輝度の平均値である。従って、この値は正規化相関値と呼ばれる。対象物体の運動部位については、基本的に剛体運動を仮定している。ただし純粋な並進運動だけでなく、物体内部および外部の基準点のまわりの回転運動にも対応できる。干渉像における干渉縞の広がりのような場合には、対象の運動が剛体運動ではないが、この場合は手動で解析するなどの方法で対応できる。

3. 負荷分散のストラテジ

対象部位が多数存在するような場合、負荷分散の方法は原則的に下記のものが考えられる。

1) フレームによる分割 (時間軸での分割)

これは実際には、前のフレームにおける物体位置が後のフレームのスタート位置として影響するので、順次位置を求めることが必要なので、かなり困難である。

2) 位置による分割 (空間軸での分割)

ROI によって分割する場合は実現容易であるが、ROI の個数が少なく、さらにそのサイズが大きい場合に負荷分散の効果があまり大きくなることが予想される。つまり ROI の数以上には分割できないということである。その場合でも、相関計算自体を並列化することで、マルチコアを利用した単一ノードでの高速化は可能である。

空間軸での分割を用いて、ネットワーク分散処理により負荷軽減ができることは既に前報までで示した。本論文では単一ノードでの相関計算の高速化をテンプレートの形状の工夫と並列計算により目指す。

4. 実験

4.1 ROI のサイズおよび形状による相関計算の高速化

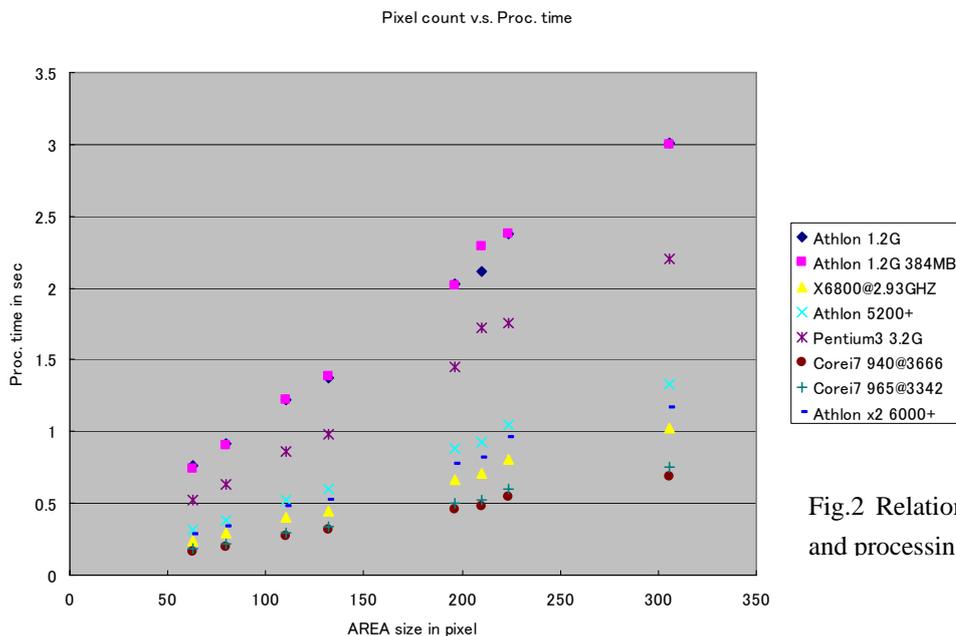


Fig.2 Relation between pixel count and processing time

対象画像と 8 個の ROI は前報までと同様である。Fig.2 に各 ROI の画素単位での大きさと処理時間の関係を、最近のものを含めた種々のマシンで計測した結果を示す。

画素の数と処理時間はほぼリニアな関係で、CPU の速度やメモリの読み出し速度によりその傾斜が異なるグラフとなっている。直線の Y 切片がゼロになっていないのは、ディスク I/O 等の時間のためと思われるが、それもまたマシンによって異なる。

次に ROI (テンプレート) と対象部位との過不足性とトレース結果の関係について述べる。Fig. 3 はテンプレートが余剰な場合、ちょうど外接する場合、内接する場合について、計算時間とトレース結果を示したものである。計算時間は左から 3.35 秒、1.56 秒、0.49 秒で、トレース結果は左二つの場合はほぼ同一で正しい結果が得られている。



3.35 1.56 0.49

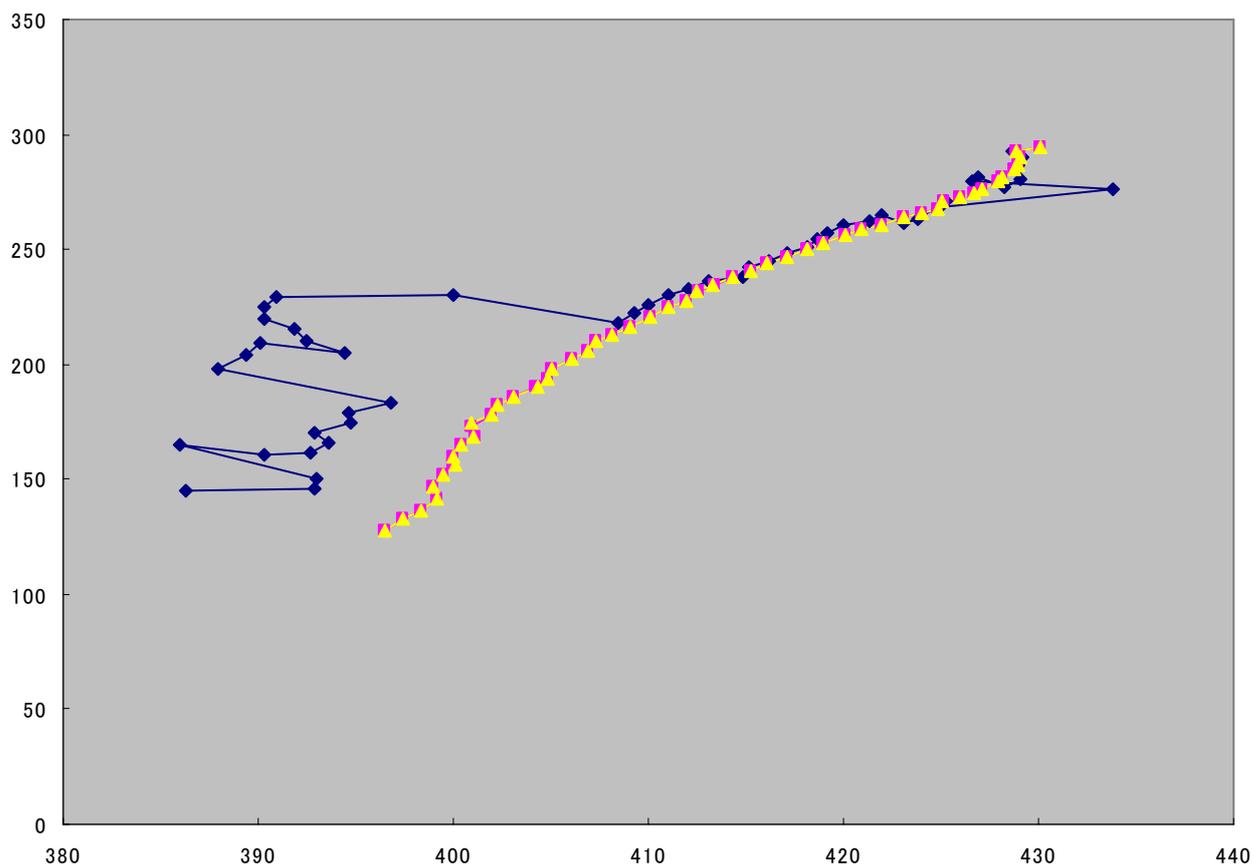


Fig.3 ROI size's effects to traceability

結局矩形のテンプレートを用いる限り、対象部位にちょうど外接するサイズが最適なテンプレートということがわかった。しかし内部の均一な部分の計算は省ける可能性がある。そこで主に対象部位の輪郭領域に物体情報があると考え、輪郭部のみを相関計算に使うアプローチを試

してみた。結果の一部を Fig. 4 に示す。本来の 195 画素のうちエッジ情報を持つ 39 画素に対してのみ計算することで、計算時間が 0.36 秒から 0.18 秒と半減し、かつ相関値そのものも大きく、より良好なトレース結果が得られた。

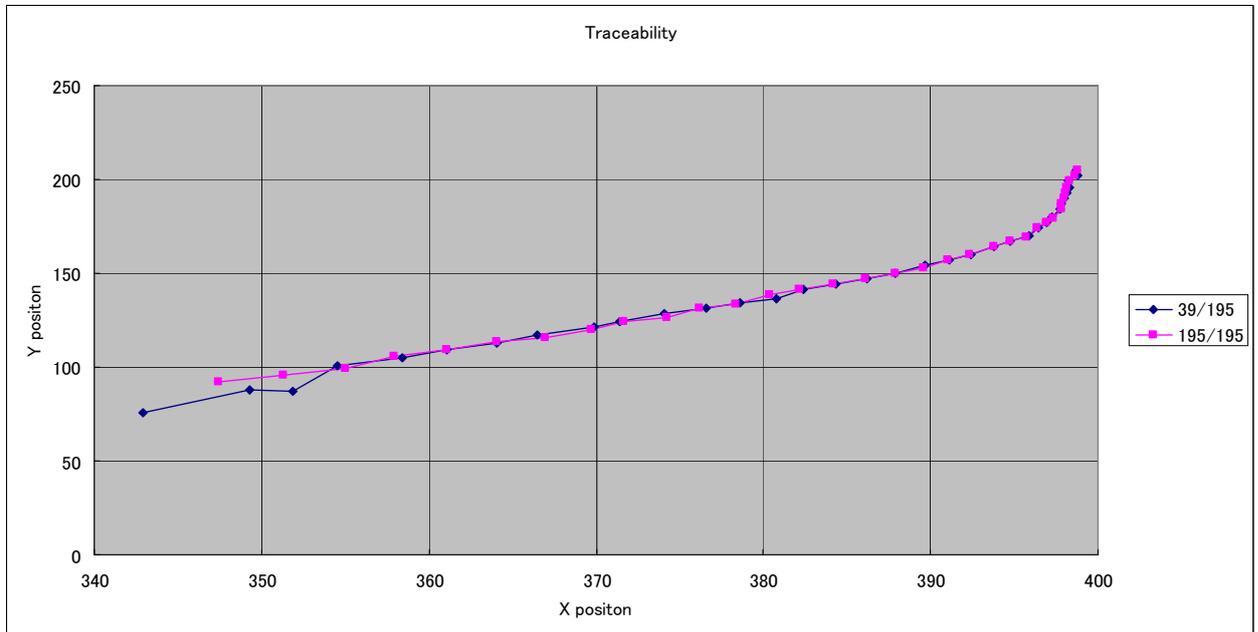


Fig. 4 Traceability results by weighted templates

4.2 OpenMP による並列処理での相関計算高速化

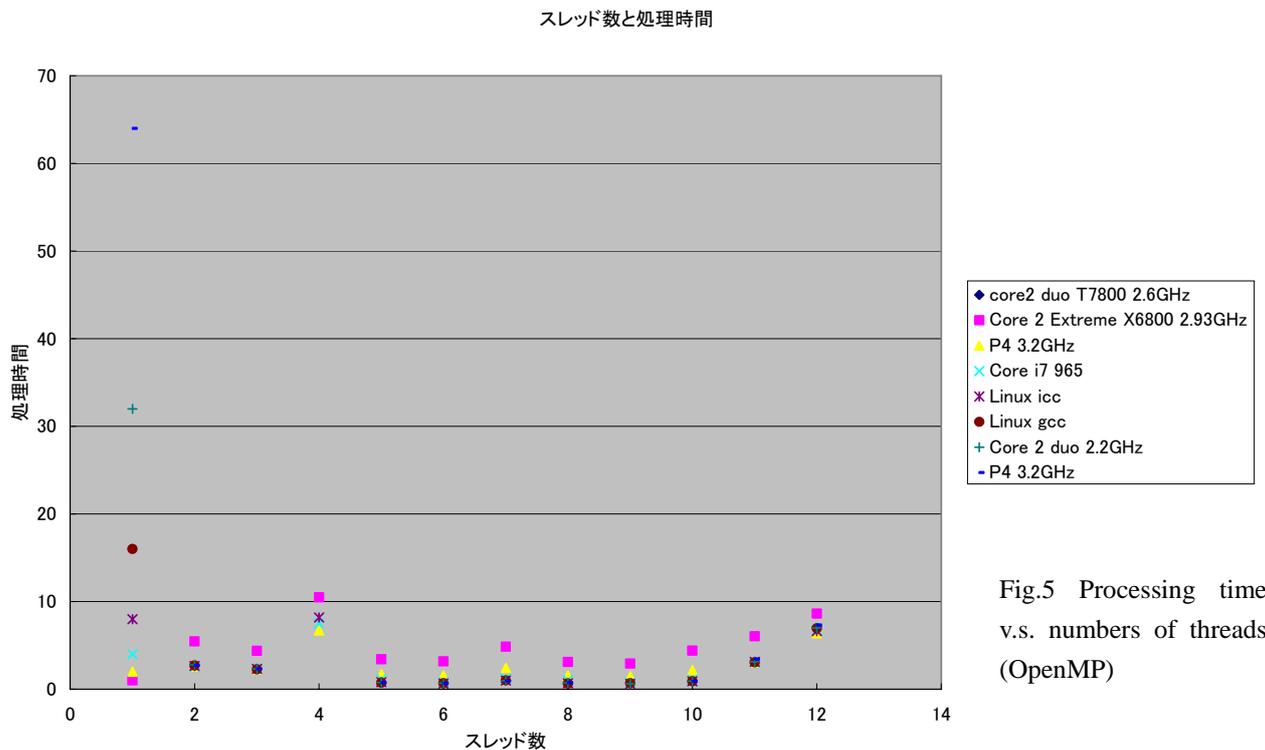


Fig.5 Processing time v.s. numbers of threads (OpenMP)

Fig.5 には 32 個の ROI を対象とし、複数の相関計算をスレッドに分配することで高速化を図った場合での、処理時間とスレッド数との関係をプロットしたものである。もちろん CPU やコンパイラの種類にも依存するが、その CPU のネイティブなコア数やスレッド数の場合に処理時間が最小化されることがわかった。たとえば最新の Intel の Core i7 の場合は、ハイパースレッドと呼ばれる機構により 1 コアあたり 2 個のスレッドが使えるので、スレッド数 8 で処理時間が最小となっている。

5. 結論

テンプレートのエッジ情報に着目し、エッジ成分が高い画素のみを相関計算に使用することで、計算の高速化だけでなくより良い追跡が可能であることがわかった。また OpenMP を用いることにより、複数の相関計算を並行して走らせることができ、それによってコア数程度の高速化が可能となることがわかった。ただしエッジ情報抽出の際のしきい値などを自動的に決定するまでにはいたっていない。従って、多数のバブルないし粒子を対象とし、重み付きテンプレートと相関計算を併用して運動解析を行う場合には、自動的にテンプレートを生成することが必要であり、今後の課題である。

参考文献

- [1] 横山直樹, 高速度撮影とフォトニクスに関する総合シンポジウム 2007 講演論文集, 2007, pp. 37-40 (2007)
- [2] Steve, V. [CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments](#), *IEEE Communications Magazine*, Vol. 14, No. 2, February, 1997.
- [3] OpenMP <http://openmp.org/>