

画像のレジストレーションにおける同時推定法の高速化手法

張 馴 権, 清水 雅夫, 奥富 正敏

東京工業大学 大学院理工学研究科 機械制御システム専攻
東京都目黒区大岡山 2-12-1

{skchang, mas}@ok.ctrl.titech.ac.jp, mxo@ctrl.titech.ac.jp

概要 本論文では, 時系列画像を有効に利用した, 同時推定法の高速化アルゴリズムを提案する. 同時推定法は, 領域ベースマッチングを拡張した高精度な画像間モーションパラメータ推定法で, 繰り返し計算を利用しないことや, 注目領域形状に制限がないなどの利点がある. 本論文で提案する高速化アルゴリズムは, 従来の同時推定法より, 約 8.5 高速である.

一方, 画像間のモーションパラメータを推定するためには, 勾配法 (Lucas-Kanade 法) が広く利用されている. 本論文では, 計算量とモーションパラメータ推定精度を勾配法と比較した. 合成モーション画像と実画像を用いた実験の結果, 提案する高速化アルゴリズムは, 勾配法の高速計算法である Inverse Compositional 法と比べて, 平均計算時間においてやや高速で, かつ繰り返し計算を必要としないため計算時間が常に一定であるという利点を持つことが確認された.

An Efficient Algorithm for Multi-Parameter Simultaneous Estimation on Image Registration

SoonKeun CHANG, Masao SHIMIZU and Masatoshi OKUTOMI

Graduate School of Science and Engineering, Tokyo Institute of Technology,
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8550, Japan

Abstract This paper proposes a fast algorithm of simultaneous motion parameter estimation method, intended for sequential image motion estimation. The simultaneous motion parameter estimation method can estimate a set of highly precise motion parameters among images using area-based image matching. This method offers advantages of non-iterative computation and a nonrestricted shape of the region of interest. The fast algorithm proposed in this paper is about 8.5 times faster than the previous simultaneous method.

This paper also compares the computational cost and the accuracy of the estimated parameters of the proposed fast algorithm with the gradient descent method (the so-called Lucas-Kanade method), which is widely used to estimate motion parameters. The computational cost of the proposed fast algorithm is slightly lower than a fast version of the gradient descent method, and their accuracies are almost equal. Experiments using a synthesized motion sequence and a real image sequence are performed to confirm the comparisons.

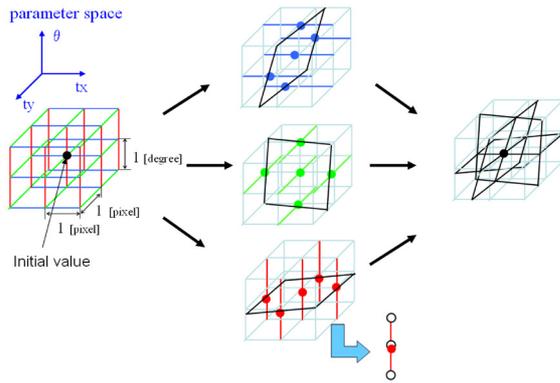


図 1. 同時推定法 (3 パラメータ).

1 まえがき

画像の超解像処理 [2, 4, 14], や道路面検出 [11], ターゲットトラッキング [15], センサーフュージョン [8] などの多くの画像処理では, 画像レジストレーションを利用して, そのためには高精度なモーション推定が必要である。

多くの場合, 注目領域間のモーション推定アルゴリズムは, 1) 注目領域内から抽出した特徴点 [10] の中から, RANSAC(Random Sample Consensus) など外れ値 (outlier) を除外した上で対応を求めて変形パラメータを推定する手法 [18, 3], 2) 注目領域の画素値をそのまま利用して, 変形パラメータを推定する手法に分けることができる. 1) の手法は, 画像のスケールや物体の形状などの影響で抽出される特徴点の位置が異なる場合があるなどの問題があり, 現在でも特徴点抽出についての研究が続いている. 2) の手法には, 繰り返し計算を利用した勾配法 [5, 9], 同時推定法 [13] などがある. 同時推定法には, 繰り返し計算を利用しない, 注目領域形状に制限がないなどの利点がある。

同時推定法は, N 次元パラメータ空間において離散的な位置で得られた画像間の類似度値を利用して, 離散分解能よりも遥かに高分解能に類似度最大位置を推定する手法である. 今までに, 同時推定法によって実際に高精度なモーションパラメータ推定が可能であることを示したが, 計算コストや計算速度に関する検討は行っていなかった. 本論文では, 時系列画像を有効に利用した, 同時推定法の高速度化アルゴリズムを提案する. 提案手法の有効性を示すため, 勾配法との計算量の比較と, 合成モーション画像と実画像を用いた実験を行った。

本論文は, 次のように構成する. 2 章では, 2 パラメータ同時推定法の拡張である 8 パラメータ同時推定法のアルゴリズムを述べる. 3 章では, 提案する高速化アルゴリズムを説明する. 4 章では, 勾配法

の計算コストについて説明する. 5 章では, 勾配法と 8 パラメータ同時推定法の比較実験を行い, 6 章で結論を述べる。

2 8 パラメータ同時推定法

8 パラメータ同時推定法は, 2 パラメータ同時推定法 [12] における平行移動量の推定を 8 パラメータの射影変形に拡張した手法である [13]. ここでは同時推定法の概略を勾配法と対応するように説明し, その計算量を見積もる。

2.1 同時推定法

領域ベースマッチングを利用した画像間の平行移動量推定では, 濃度差の 2 乗和 (SSD; Sum of Squared Difference) が最小になる画素単位の変位を探索した後, その周囲の SSD 値を利用したサブピクセル推定を行っていた. 同時推定法は領域ベースマッチングを拡張した手法で, 画像間の平行移動量に限定しない, より多くのモーションパラメータを同時にかつ高精度に推定することができる。

説明を簡単にするために, まず平行移動と回転の 3 パラメータ同時推定法を説明する. 3 パラメータ同時推定法では, 3 次元パラメータ空間における 3 平面の交点を求めることで, サブピクセルのパラメータ推定を行う. それぞれの平面を推定するために, 図 1 に示すような $5 (= 3 \times (3 - 1) + 1)$ 点のサブピクセル位置を利用する. このサブピクセル位置はそれぞれの直線上の 3 点の SSD 値を使ったパラボラフィッティングで推定する. 三つの平面を推定するため, 合計 $19 (= 2 \times 3^2 + 1)$ 個の SSD 値を利用する。

8 パラメータ同時推定法では, 8 次元パラメータ空間における 8 個の超平面を推定する. 1 個の超平面を推定するために $15 (= 2 \times (8 - 1) + 1)$ 点のサブピクセル位置を使い, 合計 $129 (= 2 \times 8^2 + 1)$ 個の SSD 値を使用する. 15 点のサブピクセル位置を推定するためには, $15 \times 3 = 45$ 点の SSD 値が必要なので, 8 個の超平面を推定するために $45 \times 8 = 360$ 点の SSD 値が必要となる. しかし, パラメータの重なりがあるため, 最終的に必要な SSD 値は 129 個となる。

画像間の平行移動に対する SSD は, 画像の標本化間隔に対応する間隔で求めることができる. しかし, 平行移動以外のモーションの場合には, モーションパラメータがあらかじめ標本化されているわけではない. このため, モーションパラメータに対応する補間画像を作れば, 任意の標本化間隔で SSD を求めることができる. 最適な標本化間隔 (Sampling Grid; SG) は, 既に [13] で与えられている。

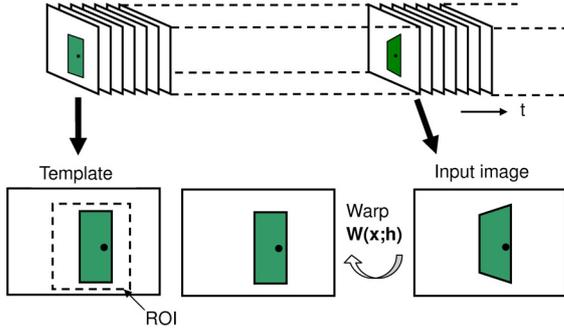


図 2. 時系列画像中のテンプレートと入力画像と変形の関係。

2.2 8パラメータ同時推定法のアルゴリズムと計算コスト

図 2 に示すように、テンプレートに対する入力画像の変形を、 $\mathbf{h} = [h_1, h_2, \dots, h_8]^T$ をパラメータとする平面射影変換で表す。

$$\mathbf{W}(\mathbf{x}; \mathbf{h}) = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

このとき、注目領域 (ROI) に対する SSD は次式で定義される。

$$E(\mathbf{W}(\mathbf{x}; \mathbf{h})) = \sum_{\mathbf{x} \in ROI} |I(\mathbf{W}(\mathbf{x}; \mathbf{h})) - T(\mathbf{x})|^2 \quad (2)$$

ここで、 I は入力画像、 T はテンプレートである。次に、同時推定法の各段階に対する計算量を示す。

(1) Warp 及び SSD: h_1 成分に対する超平面を推定するサブピクセル位置を求めるために、次の 3×15 個の SSD 値を利用する。

$$\begin{cases} E_{-\Delta_i}(\mathbf{W}(\mathbf{x}; \mathbf{m} + \mathbf{s}_i^1(-1))) \\ E_{0_i}(\mathbf{W}(\mathbf{x}; \mathbf{m} + \mathbf{s}_i^1(0))) \\ E_{+\Delta_i}(\mathbf{W}(\mathbf{x}; \mathbf{m} + \mathbf{s}_i^1(+1))) \end{cases} \quad (3)$$

ただし、 $i = \{1, \dots, 15\}$ であり、 \mathbf{m} は初期パラメータである。また、次の 15 個のベクトルの組 $\mathbf{s}_{1, \dots, 15}^1(+1)$ は h_1 成分に対する 15 個の SSD 値を求めるためのモーションパラメータの組を表している。

$$\begin{bmatrix} \mathbf{s}_1^1(\kappa) \\ \mathbf{s}_2^1(\kappa) \\ \mathbf{s}_3^1(\kappa) \\ \vdots \\ \mathbf{s}_{14}^1(\kappa) \\ \mathbf{s}_{15}^1(\kappa) \end{bmatrix} = \begin{bmatrix} \kappa \cdot \Delta_1 & 0 & \dots & 0 \\ \kappa \cdot \Delta_1 & -\Delta_2 & \dots & 0 \\ \kappa \cdot \Delta_1 & +\Delta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \kappa \cdot \Delta_1 & 0 & \dots & -\Delta_8 \\ \kappa \cdot \Delta_1 & 0 & \dots & +\Delta_8 \end{bmatrix} \quad (4)$$

表 1. 従来の 8 パラメータ同時推定法の計算コスト

Procedure	Computational complexity	$N=8$
Warp	$O((2N^2 + 1)NS)$	$O(1032S)$
SSD	$O((2N^2 + 1)S)$	$O(129S)$
Plane	$O(N^3)$	$O(512)$
Intersection	$O(N^2)$	$O(64)$
Total	$O(((2N^2 + 1)N + 1)S)$	$O(1161S + 512)$

ただし、 $\kappa = \pm 1, 0$ である。また、 $[\Delta_1, \dots, \Delta_8]^T$ は、各パラメータに対する最適 SG を表す。 $h_2 \sim h_8$ 成分に対しても同様に、合計 129 組のモーションパラメータの組を用意し、ワープと SSD の計算を行う。

(2) Plane: h_1 成分に対する超平面を推定するために、次式で 15 個のサブピクセル位置を推定する。

$$F_{1i}(h_1) = \frac{E_{-\Delta_i} - E_{+\Delta_i}}{2E_{-\Delta_i} - 4E_{0_i} + 2E_{+\Delta_i}} \quad (5)$$

式 (5) のコストは比較的小さいので、無視できる。式 (5) で得られる 15 点のサブピクセルを用い、最小二乗法で 1 個の超平面を近似する [13]。

$h_2 \sim h_8$ に対しても、同様に、各パラメータに対する 3×15 個のパラメータセットに対する SSD から 8 個の超平面を推定する。

$$a_{j1}p_1 + a_{j2}p_2 + \dots + a_{j8}p_8 + a_{j9} = 0 \quad (6)$$

ここで、 $j = \{1, \dots, 8\}$ である。また、 \mathbf{a}_j は h_j 成分に対する超平面の法線ベクトルであり、 p_j は 8 個のモーションパラメータ変数である。

(3) Intersection: 最後に、次式で 8 つのパラメータを同時に求める。

$$\begin{bmatrix} h_1 \\ \vdots \\ h_8 \end{bmatrix} = - \begin{bmatrix} a_{11} & \dots & a_{18} \\ \vdots & \ddots & \vdots \\ a_{81} & \dots & a_{88} \end{bmatrix}^{-1} \begin{bmatrix} a_{19} \\ \vdots \\ a_{89} \end{bmatrix} \quad (7)$$

表 1 に、8 パラメータ同時推定法の各段階の計算コストを示す。ここで、 N はパラメータの数、 S は注目領域内のピクセルの数である。

1 回ワープするためには、各ピクセルに対してモーションパラメータ数に対応する計算量が必要なので、**Warp:** では $129 = 2N^2 + 1$ 回のワープに対する $O((2N^2 + 1)NS)$ の計算量を必要とする。SSD を求めるためには、 $O(S)$ の計算量が必要なので、**SSD:** では $O((2N^2 + 1)S)$ の計算量が必要である。**Plane:** では $O(N^2)$ の計算量の逆行列計算をパラメータ数だけ行うので、 $O(N^3)$ の計算量が必要である。**Intersection:** では一回の逆行列演算が必要なので $O(N^2)$ の計算量が必要である。

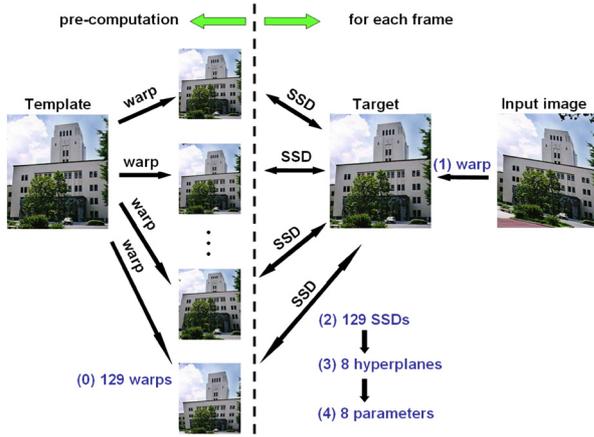


図 3. 高速化アルゴリズム.

3 提案手法

3.1 高速化アルゴリズム

従来の同時推定法では、各入力画像に対して 129 回のワーブを行うため、多くの計算量を必要とした。提案する高速化アルゴリズムでは、あらかじめテンプレートから 129 種類のワーブ画像を用意しておく。時系列画像を入力とすることを前提に、時系列画像の各フレームを初期値で 1 回だけワーブして、テンプレートから作った 129 種類のワーブ画像との SSD を計算する。このとき、各フレームを初期値で 1 回だけワーブした画像をターゲットと呼ぶことにする。

高速同時推定法のターゲットとテンプレート間の SSD を求める式は次のように表される。

$$\sum_{\mathbf{x} \in W} |T(\mathbf{W}^{-1}(\mathbf{x}; \mathbf{h})) - I(\mathbf{W}(\mathbf{x}; \mathbf{m}))|^2 \quad (8)$$

ただし、

$$\mathbf{W}^{-1}(\mathbf{x}; \mathbf{h}) = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (9)$$

である。また、 $\mathbf{W}(\mathbf{x}; \mathbf{m})$ は初期値であり、 $\mathbf{W}(\mathbf{x}; \mathbf{h})$ は推定するパラメータである。入力画像の注目領域をテンプレートへワーブする変換行列は、 $\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{h}); \mathbf{m})$ になる。以下で、アルゴリズムの手順を図 3 に合わせて説明する。

(0) Warp: テンプレートを 129 回ワーブした画像を用意する。例えば、 h_1 成分に対する 45 個のワーブパラメータは、 $\mathbf{W}(\mathbf{x}; \mathbf{0} + \mathbf{s}_i^1(-1))$, $\mathbf{W}(\mathbf{x}; \mathbf{0} + \mathbf{s}_i^1(0))$, $\mathbf{W}(\mathbf{x}; \mathbf{0} + \mathbf{s}_i^1(+1))$ である。ただし、 $\mathbf{0}$ 行列は同一画像にワーブする行列パラメータを表す。

(1) Target: 初期値で時刻 t フレームをワーブし、ターゲットを生成する。

表 2. 高速 8 パラメータ同時推定法の計算コスト

Procedure	Computational complexity	$N = 8$
Pre-comp.	$O((2N^2 + 1)NS)$	$O(1032S)$
Target	$O(NS)$	$O(8S)$
SSD	$O((2N^2 + 1)S)$	$O(129S)$
Plane	$O(N^3)$	$O(512)$
Intersection	$O(N^2)$	$O(64)$
Inverse	$O(N^2)$	$O(64)$
Total	$O((2N^2 + N + 1)S + N^3)$	$O(137S + 640)$

(2) SSD: ワーブした 129 個の画像とターゲット間の 129 個の SSD 値を求める。

(3) Plane: 求めた 129 個の SSD 値を使い、8 個の超平面を推定する。

(4) Intersection 及び Inverse: 8 個の超平面の交点を求め、逆行列を計算する。

表 2 に、8 パラメータ同時推定法の高速化アルゴリズムの計算コストを示す。表 1 と表 2 を比較すると、注目領域が十分大きいときには $\frac{137}{1161} \approx \frac{1}{8.5}$ 、すなわち約 8.5 倍高速になることがわかる。

提案手法である高速化アルゴリズムと従来の同時推定法との同一性に関しては、付録で説明する。

3.2 初期値の推定

同時推定法では、適切な初期値が得られていることを前提としている。また、次章で説明する勾配法も、全く同様に適切な初期値を必要としている。同時推定法での初期値は、画像間の平行移動量を求めるための領域ベースマッチングでは、画素単位の移動量の探索に相当する。

時系列画像に対するモーションパラメータ推定のためには、次のような初期値推定法が考えられる。

推定法 1. 前フレームに対して求めたモーションパラメータをそのまま使う。

推定法 2. 前フレームと現在フレーム間のモーションを平行移動だけに近似して、2 パラメータ同時推定法で求める。平行移動以外のモーションパラメータは前フレームに対して求めたモーションパラメータをそのまま使う。

推定法 3. 時系列画像のイメージピラミッドを構成し、縮小画像に対して推定法 2. を適応し、8 パラメータ同時推定を行う。その推定結果を元サイズの現在フレームに対する初期値とする。

時系列画像に応じて初期値推定法を選択する必要がある。推定法 1. は、計算コストを必要としないがフレーム間のモーションが大きい場合には適切な初期値が得られない可能性がある。後述する 5.2 節の実験では、推定法 3. を利用して、初期値推定法の有効性と計算コストも検討した。

表 3. Forward Additive アルゴリズムの計算コスト

	Computational complexity	$N = 8$
Per-comp.	-	-
Per iteration	$O(N^3 + N^2S + 5NS + S)$	$O(105S + 512)$

表 4. Inverse Compositional アルゴリズムの計算コスト

	Computational complexity	$N = 8$
Per-comp.	$O(2N^2S + 2NS)$	$O(144S)$
Per iteration	$O(N^3 + 2NS + S)$	$O(17S + 512)$

3.3 例外処理

同時推定法における初期値は、SG 単位でモーションパラメータを正確に表している必要がある。また、超平面を推定するためのサブピクセル位置を求めるときに、3 点の SSD 値が $E_0 < E_{-\Delta_s}$, $E_0 < E_{+\Delta_s}$ になっていることが必要である。

サブピクセル位置を求めるときに、この条件を調べ、条件が満たされていないときには次のような例外処理を行う。例外処理は、式 (4) の κ のオフセットを加えることで実現する。多くの合成モーション画像と実画像を使った実験を通して、 $-3 \leq \kappa \leq +3$ で十分であることを確認した。例外処理では、前処理よりも広範囲のモーションパラメータでワープしたテンプレート画像を用意する必要があるが、一度用意すれば再利用できる。

4 勾配法との比較

勾配法は、画像間の平行移動量を推定する手法として提案された [9] が、その後アフィン変形パラメータの推定に拡張された。その後、射影変形に拡張され [7, 16]、最近の高速化研究に至る [1]。

勾配法の目的は、テンプレートと入力画像との濃度差の 2 乗和である $E(\mathbf{W}(\mathbf{x}; \mathbf{h}))$ を最小化する変形パラメータを求めることである。 $E(\mathbf{W}(\mathbf{x}; \mathbf{h}))$ を最小化するために、 $\Delta \mathbf{h}$ だけ更新し、 $|\Delta \mathbf{h}|$ があるしきい値以下になるまで繰り返す (Forward Additive アルゴリズム [9])。

最近、テンプレートとターゲットの役割を交換し、Forward Additive アルゴリズムでは繰り返しごとに計算したヘシアン行列を前処理で 1 回だけ計算する高速化アルゴリズムが提案された (Inverse Compositional アルゴリズム [1])。

表 3 と表 4 に、それぞれ Forward Additive と Inverse Compositional アルゴリズムの計算コストを示す。¹

¹[1] では、コストの計算に多少の省略があった。表 3 と 4 はより詳細な計算コストである。



(a) 先頭フレーム.

(b) 第 100 フレーム.

図 4. 合成モーション画像.

本論文では、変形中心を注目領域の中心に設定しているが、勾配法 [1] では変形中心を注目領域の左上に設定している。この理由はページ制限により省略する。

勾配法は、推定したパラメータに対する画像間の誤差の勾配から、誤差が最小になるパラメータを推定してパラメータを更新する。これに対して同時推定法は、画像間の誤差をある間隔でサンプリングし、複数のサンプリング値を使って誤差が最小になるパラメータを推定する。このように、2 つの手法は同じ目的に対して異なるアプローチを採用している。

5 実験結果

本章では、8 パラメータ同時推定法と勾配法の推定誤差や計算時間を比較する。

5.1 合成モーション画像

この実験の目的は、前述の計算量の見積と実際の計算時間の対応を確認することである。このため、初期値推定が不要なフレーム間移動量が十分に少ない合成モーション時系列画像を使用する。なお、計算には Pentium4(2.8GHz) を使用し、計算の高速化のために Intel ライブラリ [6] を利用している。

実験に使用した合成モーション画像は、図 4(a) に示す画像から作成した合計 100 フレームの時系列画像である。画像サイズは 480×640 [画素] で、フレーム間モーションは

$\mathbf{h} = 0.05SG [+1, -1, +1, -1, +1, +1, +1, -1]^T$ に設定した。

従来の同時推定法では前処理の必要がないが、高速化同時推定法では $O(1032S)$ の計算量が必要である。これに対応して勾配法は、Forward Additive では必要なく、Inverse Compositional では $O(144S)$ の計算量が必要である。このため、前処理に関して高速化同時推定法は Inverse Compositional より約 7 倍の計算量を必要とする。しかし実際の計算時間は比較的短い (60×60 [画素]) の注目領域に対して約

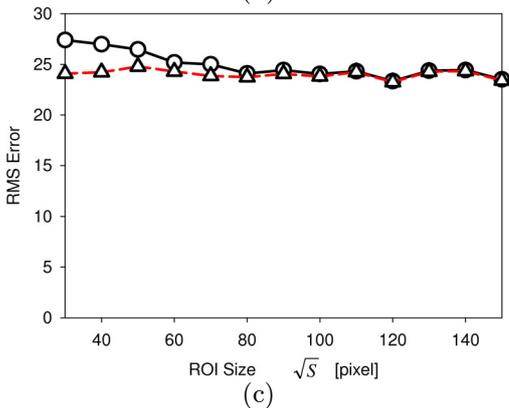
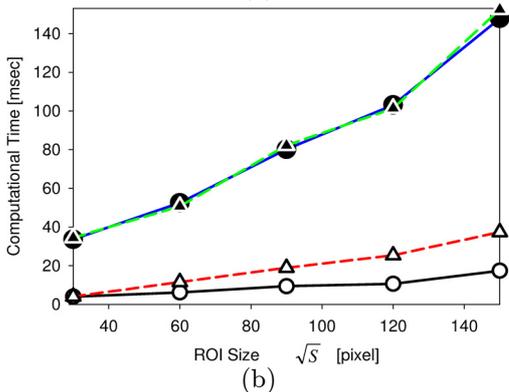
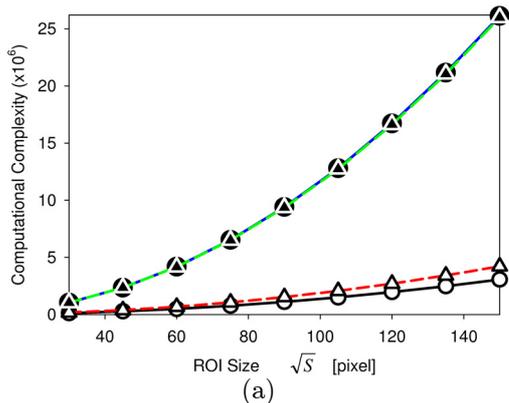


図 5. 合成モーション画像の実験結果. (a) 注目領域のサイズによる手法別計算コスト. 注目領域のサイズは注目領域の面積の平方根である. △と▲は Inverse Compositional と Forward Additive アルゴリズムを表す. ○と●は提案手法と従来の 8 パラメータ同時推定法を表す. (b) 実画像に対する注目領域のサイズによる手法別計算時間. (c) 注目領域のサイズによる手法別 RMS 誤差.

0.0468[秒]).

図 5(a) に表 1~4 で見積もった計算量の比較結果を示す. 各フレームに対する計算量は, 注目領域サイズによって異なり, 勾配法では繰り返し回数によっても異なる. 我々の実験では, 勾配法は平均 11~12 回繰り返しで収束するので, ここでは, どちらも 11 回の繰り返し回数に固定している. 60×60 [画素] の注



(a) 先頭フレーム. (b) 第 200 フレーム.

図 6. 実画像.

目領域に対して, 高速化同時推定法と同じ計算量になる繰り返し回数は, Forward Additive では 1.3 回, Inverse Compositional では 8.0 回である. 従って, 提案手法は高速勾配法である Inverse Compositional アルゴリズムより少し高速で, 注目領域が大きくなると計算時間の差は広がる.

図 5(b) に, 高速化同時推定法と 2 種類の勾配法の計算時間測定結果を示す. この結果は, 合成モーション画像中に 10 カ所の注目領域の位置を設定し, それぞれに対するモーション推定に要する時間を 10 回計測して平均した. 図 5(a)(b) を比較すると, 実際の計算時間が計算量の見積と良く一致していることがわかる.

推定したモーションパラメータの精度を比較するため, 高速化同時推定法と Inverse Compositional によって推定したモーションパラメータで各フレームを変形し, これとテンプレートとの RMS 誤差を調べた. 図 5(c) に, この結果を示す. 注目領域サイズが小さいときには若干の差があるが, 全体としてはほとんど差がない.

5.2 実画像

本節では, 手持ちカメラを使って撮影した時系列画像を使って, 初期値推定法の有効性と, モーションパラメータ推定に要する総合的な計算時間を調べた結果を示す.

図 6 に, 実験に使用した時系列画像の先頭フレーム(テンプレート)と最終フレーム(200 フレーム)を示す. 画像サイズは 640×480 [画素] である.

高速化同時推定法に対する初期値は初期値推定法 3 を利用した. 勾配法では, 手法として推定法 3 に準じる方法を利用した. すなわち, イメージピラミッドを利用し, 現在時刻の縮小画像におけるパラメータ推定では, 前フレームの元画像に対して推定したパラメータを縮小画像に適用したものをそのまま利用し, 縮小画像に対して推定したパラメータを元画像に適用して初期値とした.

勾配法の収束条件は, $\sum_i |\Delta h_i| \leq 10^{-5}$ (勾配法の推定パラメータの絶対値の総和) と最大繰り返し回数 20 回で設定した. 勾配法の場合, 最大繰り返し回数までに収束しないで発振する場合があるので, 発振

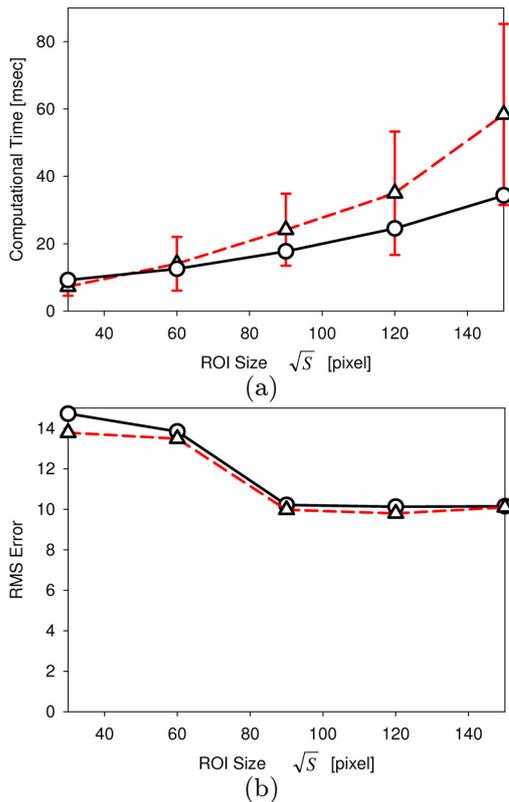


図 7. 実画像の実験結果. (a) 注目領域のサイズによる勾配法と 8 パラメータ同時推定法の計算時間. 記号の意味は図 5 と同じである. Δ (Inverse Compositional アルゴリズム) のエラーバーは手法の計算時間の最大値と最小値を表す. (b) 注目領域のサイズによる勾配法と 8 パラメータ同時推定法の RMS 誤差.

を検出したら前フレームのパラメータと現在フレームのパラメータの平均を推定パラメータとして繰り返し計算を打ち切るようにした.

図 7(a) に, 3 種類の推定法の計算時間を示す. 計算時間には初期値推定も含まれている. 時系列画像の 200 フレームに対する平均時間を示す. 勾配法では繰り返し回数にばらつきがあるため, 計算時間の最大値と最小値をエラーバーで示している.

この実験では, 勾配法の平均繰り返し回数は 11~12 回であった. 高速化同時推定の計算時間と勾配法の平均繰り返し回数に相当する計算時間を比較すると, 高速化同時推定は Inverse Compositional よりやや高速で, Forward Additive より遥かに高速な手法である.

図 7(b) に, 高速化同時推定法と Inverse Compositional によって推定したモーションパラメータで各フレームを変形し, これとテンプレートとの RMS 誤差を表す. この結果から, 二つの手法によるモーションパラメータ推定精度は, ほぼ等しいことがわかる.

6 むすび

時系列画像を有効に利用した, 同時推定法の高速化アルゴリズムを提案した. 提案した高速化同時推定法の計算量とモーションパラメータ推定精度を, 一般的によく利用されている勾配法 (Lucas-Kanade 法) と比較した. また, 合成モーション画像と実画像を用いた実験を行い, 比較結果を確認した.

勾配法は, 推定したパラメータに対する画像間の誤差の勾配から, 誤差が最小になるパラメータを推定してパラメータを更新する. これに対して同時推定法は, 画像間の誤差をある間隔でサンプリングし, サンプリング値を使って誤差が最小になるパラメータを推定する. このように, 2 つの手法は同じ目的に対して異なるアプローチを採用している.

合成モーション画像と実画像を用いた実験の結果, 提案する高速化アルゴリズムは, 勾配法の高速計算法である Inverse Compositional 法と比べて, 平均計算時間においてやや高速で, かつ繰り返し計算を必要としないため計算時間が常に一定であるという利点を持つことが確認された.

参考文献

- [1] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *International Journal of Computer Vision*, vol. 56, no. 3, March, 2004, pp. 221-255.
- [2] M. Elad and A. Feuer, "Restoration of a Single Superresolution Image from Several Blurred, Noisy, and Undersampled Measured Images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, December 1997, pp. 1646-1658.
- [3] M. Brown and D. G. Lowe, "Invariant Features from Interest Point Groups," *British Machine Vision Conference (BMVC 2002)*, Cardiff, Wales, September 2002.
- [4] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint MAP Registration and High Resolution Image Estimation Using a Sequence of Undersampled Images," *IEEE Transactions on Image Processing*, vol. 6, no.12, December 1997, pp. 1621-1633.
- [5] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, 1981, pp. 185-204.
- [6] "Intel(R) Integrated Performance Primitives 4.0," 2004.

- [7] M. Irani, B. Rousso and S. Peleg, "Computing Occluding and Transparent Motions," International Journal of Computer Vision (IJCV), vol. 12, no. 1, February 1994, pp. 5-16.
- [8] M. Irani and P. Anandan, "Robust Multi-Sensor Image Alignment," IEEE International Conference on Computer Vision (ICCV), India, January 1998.
- [9] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), April, 1981, pp. 674-679.
- [10] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," Proceedings of the 7th European Conference on Computer Vision (ECCV 2002), Copenhagen, Volume I, May 2002, pp. 128-142.
- [11] M. Okutomi, K. Nakano, J. Maruyama, and T. Hara, "Robust Estimation of Planar Regions for Visual Navigation using Sequential Stereo Images," In Proc. ICRA, 2002, pp. 3321-3327.
- [12] M. Shimizu and M. Okutomi, "Two-Dimension Simultaneous Sub-Pixel Estimation on Area-Based Image Matching," Asian Conference on Computer Vision, Jan. 28, 2004, Jeju Island, Korea.
- [13] M. Shimizu, T. Yano and M. Okutomi, "Precise Simultaneous Estimation of Image Deformation Parameters," Second IEEE Workshop on Image and Video Registration (IVR 2004), Washington DC, July 2nd, 2004.
- [14] M. Shimizu, T. Yano and M. Okutomi, "Super-Resolution under Image Deformation," 17th International Conference on Pattern Recognition (ICPR) 2004, Cambridge, UK, 23-26 August 2004.
- [15] M. R. Shortis, T. A. Clarke, and T. Short, "A Comparison of Some Techniques for the Subpixel Location of Discrete Target Images," Proceedings of the SPIE: Videometrics III, Boston, MA, USA, vol. 2350, 1994, pp. 239-250.
- [16] H.-Y. Shum and R. Szeliski, "Panoramic image mosaics," Microsoft Research, 1997.
- [17] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim, "Integrated Position Estimation Using Aerial Image Sequences," IEEE

Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, January 2002, pp. 1-18.

- [18] P. H. S. Torr and A. Zisserman, "Robust parameterization and computation of the trifocal tensor," Image and Vision Computing, vol. 15, 1997, pp. 591-605.

A 従来の同時推定法との同一性

同時推定法の高速度化アルゴリズムにおけるテンプレートワークと、従来の同時推定法における入力画像のワークが同一であることを説明する。

初期値 $\mathbf{W}(\mathbf{x}; \mathbf{m})$ と推定しようとするモーションパラメータ $\mathbf{W}(\mathbf{x}; \mathbf{h})$ により、式 (2) は次のようになる。

$$\sum_{\mathbf{x} \in ROI} |I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{h}); \mathbf{m})) - T(\mathbf{x})|^2 \quad (10)$$

式 (10) は離散信号であるが、連続信号を仮定すると、

$$\int_{\mathbf{x} \in ROI} |I(\mathbf{W}(\mathbf{W}(\mathbf{x}; \mathbf{h}); \mathbf{m})) - T(\mathbf{x})|^2 d\mathbf{x} \quad (11)$$

になる。

ここで、変数 $\mathbf{u} = \mathbf{W}(\mathbf{x}; \mathbf{h})$ 及び $\mathbf{x} = \mathbf{W}^{-1}(\mathbf{u}; \mathbf{h})$ を定義する。これは、パラメータ \mathbf{h} で変換した位置 \mathbf{x} を表す。式 (11) は、次のように表すことができる。

$$\int_{\mathbf{W}^{-1}(\mathbf{u}; \mathbf{h}) \in ROI} |I(\mathbf{W}(\mathbf{u}; \mathbf{m})) - T(\mathbf{W}^{-1}(\mathbf{u}; \mathbf{h}))|^2 \times \left| \frac{\partial \mathbf{W}^{-1}(\mathbf{u}; \mathbf{h})}{\partial \mathbf{u}} \right| d\mathbf{u} \quad (12)$$

適切な初期値が与えられたとすると、推定しようとするモーションパラメータ \mathbf{h} は同一変形 (形状が変化しないこと) に近い。このため $\left| \frac{\partial \mathbf{W}^{-1}(\mathbf{u}; \mathbf{h})}{\partial \mathbf{u}} \right| \approx 1$ と近似できる。また、積分領域も同様に、 $\mathbf{W}^{-1}(\mathbf{u}; \mathbf{h}) \approx \mathbf{u}$ と近似できる。従って、式 (12) は、次のように表すことができる。

$$\int_{\mathbf{u} \in ROI} |I(\mathbf{W}(\mathbf{u}; \mathbf{m})) - T(\mathbf{W}^{-1}(\mathbf{u}; \mathbf{h}))|^2 d\mathbf{u} \quad (13)$$

式 (13) を離散信号に戻すと

$$\sum_{\mathbf{x} \in ROI} |T(\mathbf{W}^{-1}(\mathbf{x}; \mathbf{h})) - I(\mathbf{W}(\mathbf{x}; \mathbf{m}))|^2 \quad (14)$$

になるので、従来の SSD の式 (2) と高速化アルゴリズムの SSD の式 (8) が同じ式になる。