

## 運動物体抽出を用いたぼけ動画像処理

秋山 圭<sup>†,††</sup> 羅 志偉<sup>††</sup> 大西 正輝<sup>††</sup> 細江 繁幸<sup>†,††</sup>

<sup>†</sup> 名古屋大学大学院工学研究科 〒464-8603 愛知県名古屋市千種区不老町  
<sup>††</sup> 独立行政法人理化学研究所バイオウ澗瓮膳 奪 灰鷗肇蹇璽觚 罐札鷗拭  
〒463-0003 愛知県名古屋市守山区下志段味穴ヶ洞 2271-130

E-mail: <sup>†</sup> {k\_akiyama, hosoe}@nuem.nagoya-u.ac.jp <sup>††</sup> {luo, onishi}@bmc.riken.jp

あらまし 従来、ぼけ画像復元手法として静止画像に対するウェーブレット変換に基づく最適化計算法が提案されている。しかし、これらの手法は繰り返し計算を含むため計算コストが高く、ぼけた動画像を短時間で復元することは極めて困難であった。本稿では動画像の変化規則をモデル化して、ある時刻より前の復元推定値から予測した運動物体位置を次の時刻の計算に用いることで、時系列的に変化する動画像の効率的な復元方法を提案する。最後に、運動物体抽出の計算機シミュレーションを行い、従来の手法と比較して計算時間が大幅に短縮され、しかも良好な推定結果が得られることを確認した。

キーワード 動画像復元, 運動物体抽出, ウェーブレット変換

## Restoration of Blurred Moving Image for Abstracting A Moving Object

Kei AKIYAMA<sup>†,††</sup>, Zhi-wei LUO<sup>††</sup>, Masaki ONISHI<sup>††</sup>, and Shigeyuki HOSOE<sup>†,††</sup>

<sup>†</sup> Graduate School of Engineering, Nagoya University Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan

<sup>††</sup> RIKEN BMC 2271-130, Anagahora, Shimoshidami, Moriyama-ku Nagoya, , 463-0003 Japan

E-mail: <sup>†</sup> {k\_akiyama, hosoe}@nuem.nagoya-u.ac.jp <sup>††</sup> {luo, onishi}@bmc.riken.jp

**Abstract** Iterative optimal calculation methods have been proposed for blurred static image restoration based on the wavelet transformation. However, it is quite difficult to apply these methods to process moving images due to the high computation cost. In this paper, we propose an effective restoration method for blurred moving images by modeling the motion of a moving object and predicting the future object position. We verified our method by computer simulations to show that our method can dramatically reduce the computation time.

**Key words** moving image restoration, extracting moving object, wavelet transformation

### 1. ま え が き

外界環境をカメラ等の観測装置で計測する際、観測画像には種々の要因で劣化が生じる。これらは観測装置の持つ素子特性のばらつきや大気などの媒質の揺らぎ、あるいはカメラ自身の運動などによるものである。このような劣化した観測画像から真のデータを推定する画像復元問題の研究はコンピュータビジョンの分野で盛んに行われている [4]、[6] が、その中でもウェーブレット変換を応用した多重解像度分解 [5] による手法 [1] で良好な結果が得られている。この手法の特長はウェーブレット多重解像度分解を用いることにより画像の持つ局所的な特徴を保存した復元を実現している点である。

しかしながら、文献 [1] の提案アルゴリズムは非線形最適化を含んでおり、反復計算などの高い計算コストを必要とする。このため観測画像が時系列で与えられた場合にこの手法をその

まま適用すると、処理に必要な計算コストは膨大なものになると考えられる。一般的な実環境から得られた動画像は隣り合うフレーム間で大部分が類似しており、各フレームを独立に処理するのは冗長である。そこで本稿では動画像の変化規則をモデル化し、ある時刻より前の復元推定値から予測した運動物体位置を次の時刻の計算に用いることで時系列的に変化する動画像の効率的な復元方法を提案する。最後に計算機シミュレーションを行い、提案手法の有用性を検証する。

### 2. ウェーブレット分解に基づいた静止画像復元方法

本節では動画像復元を考える上でベースとなるウェーブレット変換領域を利用した非線形画像復元手法 [1] について説明する。

## 2.1 劣化過程の定式化

$N \times N$  画素の静止画像の劣化過程は次のように定式化される。

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{u} \quad (1)$$

但し  $\mathbf{g}$  は観測画像,  $\mathbf{f}$  は真の画像,  $\mathbf{H}$  は線形の歪みやぼけを表現する行列,  $\mathbf{u}$  は加法ノイズである。ここで,  $\mathbf{g}, \mathbf{f}, \mathbf{u}$  は  $N^2$  次元ベクトル,  $\mathbf{H}$  は  $N^2$  次元正方行列であり, 画像はすべて辞書の配列順にベクトル化される。画像復元とは観測された画像  $\mathbf{g}$  から真の画像  $\mathbf{f}$  を推定することが目的である。

式 (1) は 2 次元ウェーブレット多重解像度分解 [5] を用いることで次のようにウェーブレット変換される。

$$\hat{\mathbf{g}} = \hat{\mathbf{H}}\hat{\mathbf{f}} + \hat{\mathbf{u}} \quad (2)$$

なお,  $\hat{\mathbf{g}}, \hat{\mathbf{H}}, \hat{\mathbf{f}}, \hat{\mathbf{u}}$  はウェーブレット変換行列  $\mathcal{W}$  を用いて  $\hat{\mathbf{g}} = \mathcal{W}\mathbf{g}, \hat{\mathbf{f}} = \mathcal{W}\mathbf{f}, \hat{\mathbf{u}} = \mathcal{W}\mathbf{u}, \hat{\mathbf{H}} = \mathcal{W}\mathbf{H}\mathcal{W}^T$  と表される。なお  $\mathcal{W}$  は直交性 ( $\mathcal{W}^T\mathcal{W} = I$ ) を持つ。

## 2.2 非線形最適化問題と繰り返し計算法

静止画像復元のための評価関数は次式で与えられる [1]。

$$J(\hat{\mathbf{f}}, \lambda) = \left\| \hat{\mathbf{g}} - \hat{\mathbf{H}}\hat{\mathbf{f}} \right\|_2^2 + \lambda_{(L,0)} \left\| \hat{\mathbf{f}}_{(L,0)} \right\|_p^p + \sum_{l=1}^L \sum_{j=1}^3 \lambda_{(l,j)} \left\| \hat{\mathbf{f}}_{(l,j)} \right\|_p^p \quad (3)$$

式 (3) は自然画像のウェーブレット領域における事前モデル [5] を利用した最大事後確率推定のための評価関数である。式 (3) の 2 項目以下の  $\hat{\mathbf{f}}_{(L,0)}, \hat{\mathbf{f}}_{(l,j)}$  は各解像度におけるスケーリング係数とウェーブレット係数である。また  $\lambda_{(l,j)}$  は事前モデルによって決まるパラメータである。この手法は解像度ごとに異なるパラメータ  $\lambda_{(l,j)}$  を割り当てることでエッジを保持しながら劣化要因を取り除くことができる。

式 (3) の評価関数  $J(\hat{\mathbf{f}}, \lambda)$  が最小値を持つとき, 固定点  $\hat{\mathbf{f}}^s$  での評価関数の勾配は 0 となる。ただし  $L_p$  ノルム項は  $p \leq 1$  のとき微分可能ではなくなるので, 次のように滑らかに近似する。たとえば  $\hat{\mathbf{f}}_{(L,0)}$  は

$$\left\| \hat{\mathbf{f}}_{(L,0)} \right\|_p^p \approx \sum_i \left( \left( \left| \hat{\mathbf{f}}_{(L,0)}(i) \right|^2 + \beta \right)^{p/2} - \beta^{p/2} \right) \quad (4)$$

となる。ただし  $\beta \geq 0$  は安定化定数である。他の  $\hat{\mathbf{f}}_{(l,j)}$  についても同様に近似する。

式 (4) を式 (3) に代入し勾配を取ると次の方程式が得られる。

$$\left( \hat{\mathbf{H}}^T \hat{\mathbf{H}} + \frac{p}{2} \mathbf{D}^* \right) \hat{\mathbf{f}}^* = \hat{\mathbf{H}}^T \hat{\mathbf{g}} \quad (5)$$

$$\mathbf{D}^* = \text{diag} \left[ \frac{\lambda(i)}{\left( \left| \hat{\mathbf{f}}^*(i) \right|^2 + \beta \right)^{1-p/2}} \right]_{i=1}^{N^2} \quad (6)$$

ここで  $\hat{\mathbf{f}}^*$  は  $J(\hat{\mathbf{f}}, \lambda)$  が最小となる  $\hat{\mathbf{f}}$  である。式 (5),(6) は  $\hat{\mathbf{f}}^*$  に関して非線形であるため直接解くことは困難であるが, Vogel と Oman [9] により次のような反復計算による解法が提案されている。

$$\left( \hat{\mathbf{H}}^T \hat{\mathbf{H}} + \frac{p}{2} \mathbf{D}^{(k)} \right) \hat{\mathbf{f}}^{(k+1)} = \hat{\mathbf{H}}^T \hat{\mathbf{g}} \quad (7)$$

$$\mathbf{D}^{(k)} = \text{diag} \left[ \frac{\lambda(i)}{\left( \left| \hat{\mathbf{f}}^{(k)}(i) \right|^2 + \beta \right)^{1-p/2}} \right]_{i=1}^{N^2} \quad (8)$$

式 (7),(8) を  $(\|\hat{\mathbf{f}}^{(k+1)} - \hat{\mathbf{f}}^{(k)}\| / \|\hat{\mathbf{f}}^{(k)}\|) < \gamma$  となるまで繰り返し計算することにより非線形方程式 (5),(6) の解  $\hat{\mathbf{f}}^*$  が得られる。但し  $\gamma$  は小さい正の定数である。なお, この非線形最適化法は  $p \leq 1$  でありかつ  $\hat{\mathbf{H}}$  がフルランクであれば大域的最適解に収束することが保証されている [2]。

文献 [1] では  $\lambda$  の最適な推定方法も提案されているが, 本稿では簡単のため以下で固定値として扱う。

## 3. 動画画像復元方法

観測画像が  $\mathbf{g}^{(1)} \sim \mathbf{g}^{(K)}$  のように時系列で与えられた場合の復元問題を考える。式 (7), (8) は一般に次数の大きな行列計算であり, 反復計算とともに高い計算コストが必要である。このため前節の手法をそのまま時系列画像データに適用しようとすると膨大な計算時間が必要となる。しかしながら, 十分短いサンプリング間隔で得られた時系列画像の場合, 隣り合うフレームの画像は一般的に類似している。そこで次のフレームの画像を予測し, 式 (7), (8) に対応した最適化計算を行うことで計算量を削減した動画画像復元処理方法を提案する。以下でこの詳細について説明する。

### 3.1 動画画像ダイナミクスの定式化

動画画像データを静止した背景と運動する前景とで構成されると仮定し動画画像のダイナミクスを定式化する。まず, 静止した背景の上に運動する物体の濃淡値を上書きする作用素を定義する。

$\mathbf{A}$  : 背景画像を対角に並べた行列 ( $N^2 \times N^2$  次元)

$\mathbf{h}^{(k)}$ : 背景が物体の背後に隠れている部分が 0,

それ以外の部分が 1 となるベクトル ( $N^2$  次元)

$\mathbf{m}^{(k)}$ : 運動物体のみを取り出したベクトル ( $N^2$  次元)

このとき, 背景の手前を物体が運動する画像  $\mathbf{f}^{(k)}$  は, 背景  $\mathbf{A}$  から運動物体領域  $\mathbf{h}^{(k)}$  を切り取り, 前景  $\mathbf{m}^{(k)}$  を加えることで次式のように表せる。

$$\mathbf{f}^{(k)} = \mathbf{A}\mathbf{h}^{(k)} + \mathbf{m}^{(k)} \quad (9)$$

微小時間における物体の運動を平行移動とみなし, 時刻  $k$  における運動物体領域  $\mathbf{h}^{(k)}$  と前景  $\mathbf{m}^{(k)}$  がそれぞれ遷移行列  $\mathbf{T}^{(k)}$  によって  $\mathbf{h}^{(k+1)}, \mathbf{m}^{(k+1)}$  に遷移するとすると,

$$\mathbf{h}^{(k+1)} = \mathbf{T}^{(k)}\mathbf{h}^{(k)} \quad (10)$$

となる。例えば運動物体領域  $\mathbf{h}^{(k)}$  が下方向に 1[pixel/frame], 右方向に 2[pixel/frame] 移動する場合, 画像の遷移は次のように表すことができる。

$$\begin{aligned} \mathbf{h}^{(k+1)} & [(n-1)N + m] \\ & = \mathbf{h}^{(k)} \{[(n-1) - 1]N + (m-2)\} \end{aligned} \quad (11)$$

$(1 \leq n \leq N, 1 \leq m \leq N)$

このときの遷移行列を  $\mathbf{T}^{(k)} := \mathbf{T}_{(1,2)}$  とすると次のように表すことができる。

$$\mathbf{h}^{(k+1)} = \mathbf{T}_{(1,2)} \mathbf{h}^{(k)}$$

$$\mathbf{T}_{(1,2)} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{C}_2 \\ \mathbf{C}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} & \ddots & \vdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \end{bmatrix}, \quad (12)$$

但し,

$$\mathbf{C}_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & 0 & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 \end{bmatrix} \quad (13)$$

なお、式 (12) 中の各ブロック要素は  $N$  次元正方行列である。垂直方向に  $i$ [pixel/frame]、水平方向に  $j$ [pixel/frame] 移動する場合、式 (12) の行列  $\mathbf{C}_j$  は対角ブロックから  $i$  ブロック下に配置され、式 (13) の  $\mathbf{C}_j$  は対角要素から  $j$  個下の要素が 1、それ以外が 0 となる。

また、前景  $\mathbf{m}^{(k)}$  は  $\mathbf{f}^{(k)}$  から運動物体領域  $\mathbf{h}^{(k)}$  が 0 の部分を取り出すことで次式のように得られる。

$$\mathbf{m}^{(k)} [(n-1)N+m] = \begin{cases} \mathbf{f}^{(k)} [(n-1)N+m] & \text{if } \mathbf{h}^{(k)} [(n-1)N+m] = 0 \\ 0 & \text{if } \mathbf{h}^{(k)} [(n-1)N+m] = 1 \end{cases} \quad (14)$$

### 3.2 動画復元における提案アルゴリズム

$\mathbf{T}^{(k)}$ ,  $\mathbf{A}$ ,  $\mathbf{m}^{(k)}$ ,  $\mathbf{h}^{(k)}$  はいずれも未知であり、復元推定値  $\mathbf{f}_{est}^{(k)}$  からこれらを推定する必要がある。

まず次のように仮定する。

- 真の画像  $\mathbf{f}$  は式 (9) で与えられる。
- 劣化画像  $\mathbf{g}$  は次式で与える。

$$\mathbf{g}^{(k)} = \mathbf{H}\mathbf{f}^{(k)} + \mathbf{u}^{(k)} \quad (15)$$

• 運動物体領域  $\mathbf{h}^{(k)}$  の重心  $(c_i^{(k)}, c_j^{(k)})$  は等加速度運動する。

$$\begin{aligned} c_i^{(k+1)} &= 2c_i^{(k)} - c_i^{(k-1)} + a_i \\ c_j^{(k+1)} &= 2c_j^{(k)} - c_j^{(k-1)} + a_j \end{aligned} \quad (16)$$

• 運動物体は単一で大きさは不変とし、常に同じ姿勢で運動する。

#### 3.2.1 運動領域抽出

動画からの運動領域抽出手法は数多く提案されている [7], [8] が、ここでは一般的な方法で運動物体領域  $\mathbf{h}^{(k)}$  を抽出する。まず差分画像  $\mathbf{f}_s^{(k)} := \mathbf{f}_{est}^{(k)} - \mathbf{f}_{est}^{(k-1)}$  を作成する。 $\mathbf{f}_s^{(k)}$  を閾値処理した画像  $\mathbf{f}_{sb}^{(k)}$  を計算し、 $\mathbf{f}_{sb}^{(k)}$  が 0 でない領域内で  $\mathbf{f}_{est}^{(k)}$  のエッジ検出を行い、この結果を運動物体のエッジとする。膨張、穴埋め、収縮処理を行った結果得られた領域を運動物体領域  $\mathbf{h}^{(k)}$  とする。

#### 3.2.2 運動領域予測

過去の情報から次のフレームの運動物体領域の予測値  $\mathbf{h}_{est}^{(k+1)}$  を求める。上記の仮定の下では運動物体領域全体の位置を予測する必要はなく、領域内の特定の点を予測すればよい。そこで運動物体領域  $\mathbf{h}^{(k)}$  の重心座標  $(c_i^{(k)}, c_j^{(k)})$  を計算する。

$$\begin{aligned} c_i^{(k)} &= \frac{\sum_{n=1}^N \sum_{m=1}^N n \cdot \mathbf{h}^{(k)} [(n-1)N+m]}{\sum_{n=1}^N \sum_{m=1}^N \mathbf{h}^{(k)} [(n-1)N+m]} \\ c_j^{(k)} &= \frac{\sum_{n=1}^N \sum_{m=1}^N m \cdot \mathbf{h}^{(k)} [(n-1)N+m]}{\sum_{n=1}^N \sum_{m=1}^N \mathbf{h}^{(k)} [(n-1)N+m]} \end{aligned} \quad (17)$$

カルマンフィルタを用いて  $k+1$  フレーム目の重心座標を推定する。各フレームで算出された重心座標  $(c_i^{(k)}, c_j^{(k)})$  が雑音を含むと仮定し、状態方程式を式 (16) より次のように定義する。

$$\begin{cases} \xi_i[k+1] = \mathbf{F}_p \xi_i[k] + \mathbf{G}_p \mathbf{w}_i[k] \\ \mathbf{y}_i[k] = \mathbf{H}_p \xi_i[k] + \mathbf{v}_i[k] \end{cases} \quad (18)$$

ただし

$$\xi_i[k] := [c_i^{(k)}, c_i^{(k-1)}, a_i^{(k)}]^T \quad (19)$$

$$\mathbf{F}_p = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{G}_p = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad (20)$$

$$\mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

である。上式は垂直方向の運動に関する方程式であるが、水平方向に関しても同様である。

推定値  $\hat{\xi}_i[k+1|k]$ ,  $\hat{\xi}_j[k+1|k]$  から重心位置の予測値  $(c_{ei}^{(k+1)}, c_{ej}^{(k+1)})$  が得られる。予測値から遷移行列  $\mathbf{T}^{(k)}$  を

$$\mathbf{T}^{(k)} := \mathbf{T}_{(\Delta c_i^{(k)}, \Delta c_j^{(k)})} \quad (21)$$

$$(\Delta c_i^{(k)}, \Delta c_j^{(k)}) = (c_{ei}^{(k+1)} - c_i^{(k)}, c_{ej}^{(k+1)} - c_j^{(k)}) \quad (22)$$

と求め、式 (10) により

$$\mathbf{h}_{est}^{(k+1)} := \mathbf{T}^{(k)} \mathbf{h}^{(k)} \quad (23)$$

と計算することで運動物体領域の予測値  $\mathbf{h}_{est}^{(k+1)}$  を得る。

### 3.2.3 動画復元アルゴリズム

上記を踏まえ、動画復元アルゴリズムを以下に示す。

<  $k$  フレーム目におけるアルゴリズム >

(1)  $k-1$  フレーム目で得られた初期値  $\mathbf{f}_i^{(k)}$  からウェーブレット変換形  $\hat{\mathbf{f}}_i^{(k)} = \mathcal{W}\mathbf{f}_i^{(k)}$  を計算する。 $k=1$  のときは  $\hat{\mathbf{f}}_i^{(1)}$  を適当に与える。

(2)  $\hat{\mathbf{f}}_i^{(k)}$  を用いて復元推定値  $\hat{\mathbf{f}}_{est}^{(k)}$  を次式により求める。

$$\left(\hat{\mathbf{H}}^T \hat{\mathbf{H}} + \frac{\beta}{2} \mathbf{D}^{(k)}\right) \hat{\mathbf{f}}_{est}^{(k)} = \hat{\mathbf{H}}^T \hat{\mathbf{g}}^{(k)} \quad (24)$$

$$\mathbf{D}^{(k)} = \text{diag} \left[ \frac{\lambda(i)}{(|\hat{\mathbf{f}}_i^{(k)}(i)|^2 + \beta)^{1-p/2}} \right]_{i=1}^{N^2} \quad (25)$$

$\mathbf{f}_{est}^{(k)} = \mathcal{W}^T \hat{\mathbf{f}}_{est}^{(k)}$  により逆ウェーブレット変換形  $\mathbf{f}_{est}^{(k)}$  を得る。

(3)  $\hat{\mathbf{f}}_{est}^{(k)}$  から3.2.1の方法で運動物体領域  $\mathbf{h}^{(k)}$  を抽出する。

(4) 式(14)を用いて、 $\mathbf{h}^{(k)}$  により  $\mathbf{f}_{est}^{(k)}$  から前景  $\mathbf{m}^{(k)}$  を取り出す。また背景  $\mathbf{A}^{(k)}$  は次式により求める。

$$\mathbf{A}^{(k)} [(n-1)N + m] = \begin{cases} \mathbf{f}_{est}^{(k)} [(n-1)N + m] & \text{if } \mathbf{h}^{(k)} [(n-1)N + m] = 0 \\ 0 & \text{if } \mathbf{h}^{(k)} [(n-1)N + m] = 1 \end{cases} \quad (26)$$

(5)  $\mathbf{h}^{(k)}$  から重心  $(c_i^{(k)}, c_j^{(k)})$  を検出し、3.2.2の方法で運動物体領域の予測値  $\mathbf{h}_{est}^{(k+1)}$  を求める。

(6) 前景と背景の予測値  $\mathbf{m}_{est}^{(k+1)}, \mathbf{A}_{est}^{(k+1)}$  を求める。 $\mathbf{m}_{est}^{(k+1)}$  は前のステップで  $\mathbf{h}_{est}^{(k+1)}$  を求めた時の遷移行列を用いて計算する。 $\mathbf{A}_{est}^{(k+1)}$  の各対角要素は  $\mathbf{A}^{(l)}$  ( $1 \leq l \leq k$ ) の各要素の平均を取ることで求める。

(7)  $k+1$  フレーム目の初期値  $\mathbf{f}_i^{(k+1)}$  を次式により得る。

$$\mathbf{f}_i^{(k+1)} = \mathbf{A}_{est}^{(k+1)} \mathbf{h}_{est}^{(k+1)} + \mathbf{m}_{est}^{(k+1)} \quad (27)$$

$\mathbf{f}_i^{(k+1)}$  をウェーブレット変換し、 $\hat{\mathbf{f}}_i^{(k+1)}$  を得る。

(8)  $1 \leq k \leq K$  に関して手順(2) - (7)を繰り返す。

上記のアルゴリズムのように方程式(24), (25)の計算に用いる初期値  $\mathbf{f}_i^{(k)}$  を真値  $\mathbf{f}^{(k)}$  に近づくように予測して与えることで、予測を行わない場合と比較して復元推定値  $\mathbf{f}_{est}^{(k)}$  の推定精度が向上すると考えられる。

## 4. シミュレーション

64×64画素の濃淡画像を用いてシミュレーションを行った。背景として標準画像 Boat を用い、前景として一定濃淡値で半径5[pixel]の円形の物体が  $a_i = 0.1, a_j = 0$  で放物運動する場合を考えた。なお運動物体の重心座標の初期値は(16, 15)とした。背景の上に前景を重ねたものを真の画像  $\mathbf{f}^{(k)}$  とし、 $\mathbf{f}^{(k)}$  に  $\sigma^2 = 1.5$  のガウス関数を畳み込み、さらにSN比30[db]の白色ガウスノイズ  $\mathbf{u}^{(k)}$  を加えたものを劣化画像  $\mathbf{g}^{(k)}$  とした。なお  $\mathbf{u}^{(k)}$  は全てのフレームで共通とした。3.2.3のアルゴリズムに従って動画復元を行い、提案手法と各フレームごとに文献[1]の方法で最適化計算を行ったものを比較した。計算

表1 シミュレーションに用いたパラメータ

ガウスぼけ処理の分散 ( $\sigma^2$ )	1.5
加法ノイズ (SN比)	30 [dB]
ウェーブレット分解レベル ( $L$ )	3
$(\lambda_1, \lambda_2)$	(0.1, 0.05)
$p$	1.0
$\alpha$	1.0
$\beta$	1.0

にはCPUがPentium4(3.4[GHz])、メモリが1[GByte]の計算機を使用し、開発言語はMatlabを用いた。

ウェーブレット分解レベル  $L$  は3とし、ウェーブレット変換にはDaubechiesの4タップウェーブレット[3]を用いた。式(24), (25)は共役勾配法を用いて計算した。またパラメータ  $\lambda_{(l,j)}$  は次のようにおいた。

$$\begin{aligned} \lambda_{(3,0)} &= \lambda_1 \\ \lambda_{(2,j)} &= \lambda_2, & 1 \leq j \leq 3 \\ \lambda_{(1,j)} &= \lambda_2 2^{-\alpha(l-3)}, & 1 \leq j \leq 3 \end{aligned} \quad (28)$$

なお  $k=1$  における初期値  $\mathbf{f}_i^{(1)}$ , 推定値  $\mathbf{f}_{est}^{(0)}$  はそれぞれ  $\mathbf{f}_i^{(1)} = \mathbf{g}^{(1)}$ ,  $\mathbf{f}_{est}^{(0)} = \mathbf{0}$  とし、運動推定に用いる初期状態の平均値  $\bar{\xi}_i[0], \bar{\xi}_j[0]$  と誤差共分散行列  $\bar{\Sigma}_{\xi_i[0]}, \bar{\Sigma}_{\xi_j[0]}$ , ノイズ  $\mathbf{w}_i[k], \mathbf{w}_j[k], \mathbf{v}_i[k], \mathbf{v}_j[k]$  の共分散行列  $\Sigma_{\mathbf{w}_i[k]}, \Sigma_{\mathbf{w}_j[k]}, \Sigma_{\mathbf{v}_i[k]}, \Sigma_{\mathbf{v}_j[k]}$  はそれぞれ次のように与えた。

$$\bar{\xi}_i[0] = \bar{\xi}_j[0] = [0, 0, 0]^T \quad (29)$$

$$\begin{aligned} \bar{\Sigma}_{\xi_i[0]} &= \bar{\Sigma}_{\xi_j[0]} = \Sigma_{\mathbf{w}_i[k]} = \Sigma_{\mathbf{w}_j[k]} \\ &= \Sigma_{\mathbf{v}_i[k]} = \Sigma_{\mathbf{v}_j[k]} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (30)$$

なおシミュレーションに用いたパラメータを表1に示す。

### 4.1 結果と評価

図1に3フレーム目(a), 21フレーム目(b), 37フレーム目(c)の真の画像  $\mathbf{f}$  と劣化画像  $\mathbf{g}$  を、図2に図1の各フレームにおける復元推定結果  $\mathbf{f}_{est}$  を示す。図2の各列は左が提案手法、右が[1]の方法で各フレームごとに最適化計算を行った場合である。提案手法の推定値はいずれのフレームにおいても劣化画像と比較して輪郭部がより鮮明になっており、各フレームごとに最適化計算を行った場合と比較して見た目では大差のない結果が得られている。

次に、3, 21, 37フレーム目における前景と推定値  $\mathbf{m}_{est}$  を図3に示す。3フレーム目の前景  $\mathbf{m}_{est}^{(3)}$  が正しく抽出されていないが、これは復元計算によって画像全体の推定値が変化しており、運動物体領域が正しく抽出されなかったためと考えられる。それに対し21, 37フレーム目では前景を正しく抽出できていることがわかる。このことから、3.2.1の運動領域抽出により初期のフレーム以外では運動物体領域  $\mathbf{h}^{(k)}$  を正しく取り出すことができたといえる。一方図4には21, 37フレームにおける背景の推定結果  $\mathbf{A}_{est}$  を示す。どちらのフレームとも前景を重ねる前の画像 Boat に近い画像が得られている。これは運動

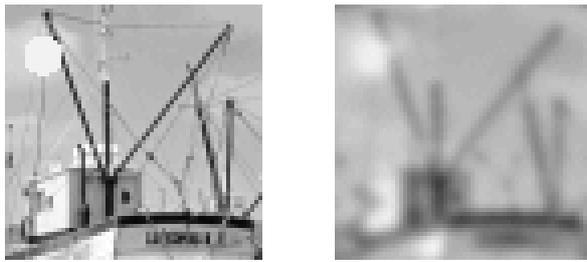
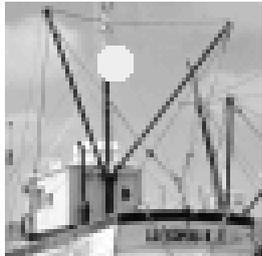
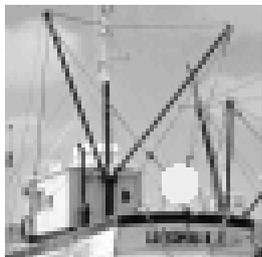
(a) 真の画像  $f^{(3)}$ (a') 劣化画像  $g^{(3)}$ (b) 真の画像  $f^{(21)}$ (b') 劣化画像  $g^{(21)}$ (c) 真の画像  $f^{(37)}$ (c') 劣化画像  $g^{(37)}$ 

図 1 真の画像  $f$  と劣化画像  $g$   
(3, 21, 37 フレーム目)

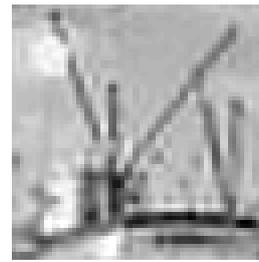
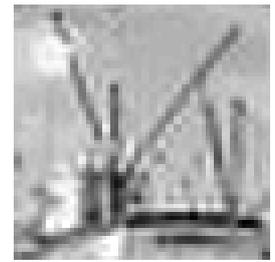
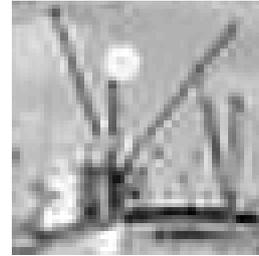
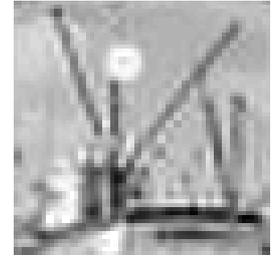
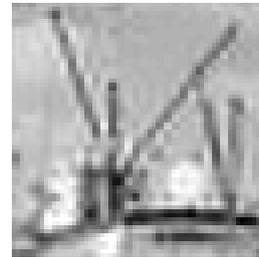
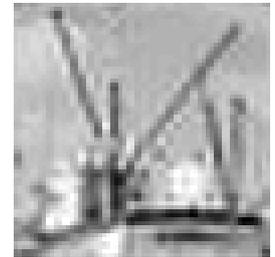
(a)  $f_{est}^{(3)}$  (提案手法)(a')  $f_{est}^{(3)}$  (静止画)(b)  $f_{est}^{(21)}$  (提案手法)(b')  $f_{est}^{(21)}$  (静止画)(c)  $f_{est}^{(37)}$  (提案手法)(c')  $f_{est}^{(37)}$  (静止画)

図 2 提案手法と静止画復元手法の推定結果

領域の推定により運動物体領域が正しく取り除かれたためである．図 3, 4 から, 3.2.1 の方法により物体の運動物体領域が正しく抽出され, 前景と背景の復元推定値をほぼ正確に得られることが示された．

さらに, 図 5 には 3.2.2 の方法による運動物体領域の重心位置の予測結果  $(c_{est}^{(k)}, c_{ej}^{(k)})$  を示す．(a) が垂直方向, (b) が水平方向であり, 各図実線が予測結果, 破線が真の運動物体領域の重心位置である．真の重心は垂直方向に関しては等加速度運動, 水平方向に関しては等速運動を行っている．予測結果は (a), (b) とも初期のフレームで誤差が見られるものの, その後のフレームでは真の値に近い予測値が得られている．したがって図 5 より, 等加速度運動を行う物体に対して 3.2.2 の方法により運動物体領域の予測が正しく行えていることが示された．なお, 初期のフレームにおける誤差の原因は, 復元処理により画像全体の推定値がフレーム間で変化し, 運動物体領域を正しく抽出できていないためと考えられる．

図 6 には 3~37 フレーム目における復元推定値と真値との RMSE を示す．RMSE とは平均 2 乗誤差の平方 (Root Mean Square Error) であり, 次のように定義する．

$$\text{RMSE} := \sqrt{\frac{\|f - f_{est}^{(k)}\|^2}{N^2}} \quad (31)$$

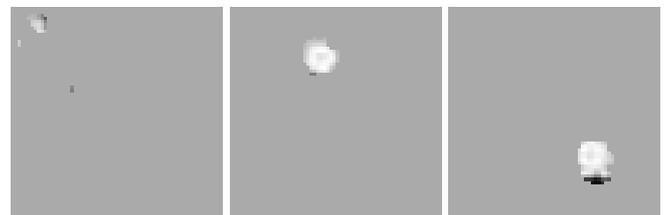
(a)  $m_{est}^{(3)}$ (b)  $m_{est}^{(21)}$ (c)  $m_{est}^{(37)}$ 

図 3 前景画像の推定結果

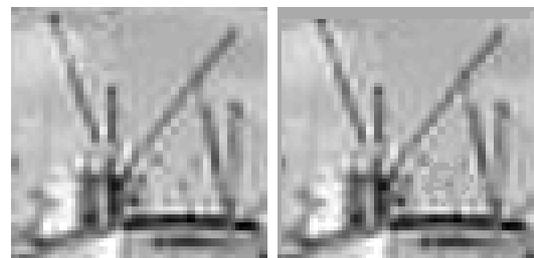
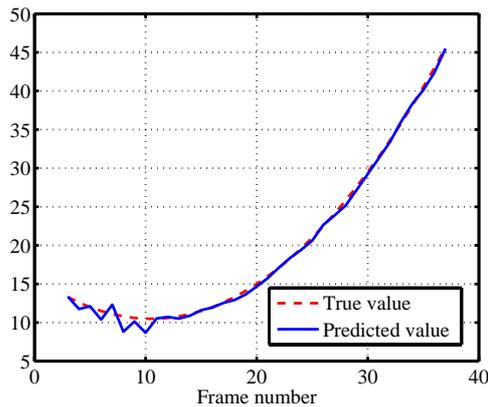
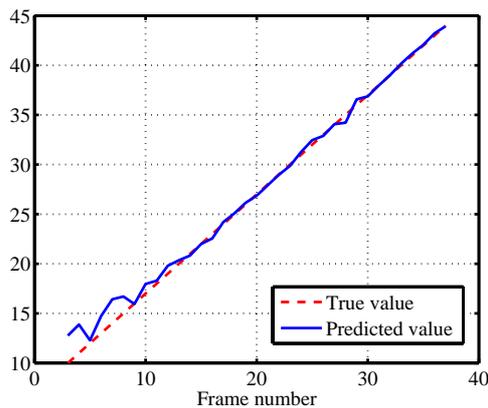
(b)  $A_{est}^{(21)}$  $A_{est}^{(37)}$ 

図 4 背景画像の推定結果 (21, 37 フレーム目)

図中の実線が提案手法, 破線が静止画復元手法, 一点鎖線が観測画像  $g^{(k)}$  と真値と RMSE である．提案手法の推定値はいずれのフレームにおいても劣化画像と比べ RMSE が改善していることがわかる．また, 提案手法は初期の数フレームを除いて静止画復元手法とほぼ変わらない結果が得られている．



(a) 垂直方向



(b) 水平方向

図 5 重心位置推定結果  $(c_{ei}^{(k)}, c_{ej}^{(k)})$

表 2 計算時間と推定精度の比較

	提案手法	静止画復元手法
計算時間	23min	5h32min
真値との RMSE(37 フレーム目)	6.2222	6.2559

表 2 には 37 フレーム目における計算時間と推定誤差の比較を示す。表 2 から提案手法が静止画復元手法と比較して大幅に計算時間が短縮されているにもかかわらず、静止画復元手法と比べてほとんど変わらない結果が得られていることが分かる。

以上のシミュレーション結果から、本稿で提案する動画復元アルゴリズムにおいて運動物体領域の抽出と予測が正しく行われ、提案手法が従来手法と比較して計算時間を大幅に短縮し、しかも従来手法と同程度の推定結果を得られることが示された。

## 5. むすび

本稿では、動画像の変化規則をモデル化してある時刻より前の復元推定値から予測した運動物体位置を次の時刻の計算に用いることで、時系列的に変化する動画像の効率的な復元方法を提案した。計算機シミュレーション結果から従来手法と比較して計算時間が大幅に短縮され、しかも従来手法と変わらない推定結果を得られることが確認できた。今後は静止画復元手法 [1] を時系列データに直接拡張する最適化計算法の検討などを行っていく予定である。

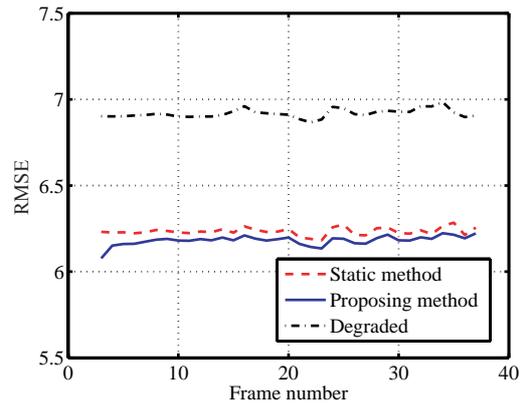


図 6 真値  $f^{(k)}$  と推定値  $f_{est}^{(k)}$  との RMSE

## 文 献

- [1] M.Belge, M.E.Kilmer, and E.L.Miller, "Wavelet domain image restoration with adaptive edge-preserving regularization," IEEE Trans. on Image Processing, vol. 9, pp. 597–608 Apr. 2000.
- [2] P.Charbonnier, L.Blanc-Feraud, G.Aubert and M.Barlaud, "Deterministic edge-preserving regularization in computed imaging," IEEE Trans. on Image Processing, vol. 6, pp. 298–311, Feb. 1997.
- [3] I.Daubechies, "Ten Lectures on Wavelet," SIAM, 1992.
- [4] D.Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," IEEE Trans. on Image Processing, vol. 4, pp. 932–946, July 1995.
- [5] S.Mallat, "A Theory for Multiresolution signal decomposition: the wavelet representation," IEEE Trans. on PAMI, vol. 11, no. 7, pp. 674–693, 1989.
- [6] S.Osher, L.I.Ruden and E.Fatemi, "Nonlinear total variation based noise removal algorithms," Phis. D, vol. 60, pp. 259–268, 1992.
- [7] S.A.Stansfield, "ANGY: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms," IEEE Trans. on PAMI, vol. 8, no. 2, pp. 188–199, 1986.
- [8] L.Vincent and P.Sopille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," IEEE Trans. on PAMI, vol. 13, no. 6, pp. 583–598, June 1991.
- [9] C.R.Vogel and M.E.Oman, "Fast, robust total variation-based reconstruction of noisy blurred images," IEEE Trans. on Image Processing, vol. 7, pp. 813–824, July 1998.