

FFT 全探索ブロックマッチングに基づくサブピクセル精度高速ブロックマッチング法

植村 淳志[†] 李 禎[†] 貴家 仁志[†]

A Fast Sub-Pixel Block-Matching Algorithm Using FFT-Based Full-Search Block-Matching

Atsushi UEMURA[†], Zhen LI[†], and Hitoshi KIYA[†]

あらまし 本論文では、ピクセル精度のブロックマッチングを FFT 全探索ブロックマッチング法によって行った際に適用可能な、新たなサブピクセル精度のブロックマッチング法を提案する。一般にサブピクセル精度のブロックマッチングは、サブピクセルの画素を補間によって生成し、実行される。一方、提案法は、ピクセル精度の推定の際に算出した種々の値を直接補間することで、サブピクセル精度のブロックマッチングを実行する。提案法が従来法と完全に同精度であり、かつ低演算量であることを示す。

キーワード サブピクセルブロックマッチング, SSD, 双線形補間

1. ま え が き

ブロックマッチングは、動画像符号化や画像認識、消失領域の復元 [1], [2] など多くの分野で応用されている技術である。その計算は単純であるが、多くの計算量が必要である。そのため、推定精度をできるだけ維持したまま、計算量を削減する方法についての研究が盛んに行われている [3] ~ [10]。その中でも、FFT による相関計算に基づく方法 [7] ~ [10] は全探索の精度を維持しており、かつ低演算量である。更に、計算量が信号に依存しないという特徴をもつ。

多くのマッチング応用においては、整数ピクセル精度を上回るマッチング精度が求められている。ピクセル精度の推定同様、いくつかの推定法が提案されている [11] ~ [16]。第一の方法は、画像を補間によって部分的に高解像度化し、画像間の相違度を計算するという方法 [11], [12] である。しかし、高精度要求時に計算コストが高いという欠点がある。第二の方法は、ピクセル精度の推定で得られた相違度の値をある関数でフィッティングし、解析的に最適位置を求める方法 [13] ~ [15] である。しかし、フィッティング関数はブロックマッ

チングで使用される絶対誤差和、二乗誤差和などの誤差尺度に対し、最適である保証はない。FFT を用いるブロックマッチング法では、整数ピクセル精度においてはその有効性が確認されている [7] ~ [10]。しかし、サブピクセル精度では考察が行われていない。

従来法は空間領域で画素を補間するという特徴をもつ。それに対し、本論文で提案する方法では、[8] ~ [10] の方法でピクセル精度のマッチングを行った際に算出した値を直接補間することで、サブピクセル精度のマッチングを行う。この方法は、サブピクセルの画素を補間する方法と比較して低演算量であり、かつ画像を補間して得られた誤差尺度と完全一致する。つまり、ピクセル精度のブロックマッチングにおいて、低演算量かつ高精度である FFT に基づく方法を用いることで、サブピクセル精度においても、低演算量かつ高精度なブロックマッチングが可能である。更に、サブピクセル精度の参照点が多いほど提案法は有効である。シミュレーションにより提案法の有効性を確認する。

2. 準 備

ここでは準備として、ブロックマッチングの定義と、FFT を用いたブロックマッチングの高速アルゴリズムについて説明する。

[†] 首都大学東京大学院システムデザイン研究科, 日野市
Faculty of System Design, Tokyo Metropolitan University,
Hino-shi, 191-0065 Japan

2.1 ブロックマッチング

ある二次元信号 $b(x, y)$ を対象ブロック, $f(x, y)$ を探索領域と呼ぶことにする．ここで,

$$b(x, y), x = 0, 1, \dots, A-1, y = 0, 1, \dots, B-1 \quad (1)$$

$$f(x, y), x = 0, 1, \dots, M-1, y = 0, 1, \dots, N-1 \quad (2)$$

$$M > A \quad \text{and} \quad N > B$$

$$x, y, M, N, A, B \in \mathbf{Z}$$

とする．ただし, \mathbf{Z} は整数全体の集合である．本論文では, 探索領域内から対象ブロックと一番誤差が小さい領域を探し出すことをブロックマッチングと定義する．一般に誤差尺度は絶対誤差和 (sum of absolute differences : SAD) と二乗誤差和 (sum of squared differences : SSD) の二つの基準が使われている．それぞれ

$$\text{SAD}_{b,f}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} |b(x, y) - f(u+x, v+y)| \quad (3)$$

$$\text{SSD}_{b,f}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} (b(x, y) - f(u+x, v+y))^2 \quad (4)$$

$$(u, v) \in \mathbf{R}_{\text{IN}}$$

$$\mathbf{R}_{\text{IN}} = \{(u, v) | u = 0, 1, \dots, M-A, \\ v = 0, 1, \dots, N-B\}$$

と定義される．ブロックマッチングとは, 誤差が一番小さいときのシフト量 (u, v) を探し出すことである．誤差が最小となるシフト量を (u_0, v_0) とすると, それぞれ

$$\text{SAD}_{b,f}(u_0, v_0) = \min_{(u,v) \in \mathbf{R}_{\text{IN}}} (\text{SAD}_{b,f}(u, v)) \quad (5)$$

$$\text{SSD}_{b,f}(u_0, v_0) = \min_{(u,v) \in \mathbf{R}_{\text{IN}}} (\text{SSD}_{b,f}(u, v)) \quad (6)$$

と表現される．特に, 可能なシフト量すべてについて調べることを全探索と呼ぶ．本論文で論ずる FFT によるブロックマッチング法は, SSD を誤差尺度とする全探索ブロックマッチング法である．

2.2 整数ピクセル精度のブロックマッチング法

文献 [9] を例にして, FFT 全探索ブロックマッチン

グ法について簡単に説明する．式 (4) より, 誤差尺度 SSD は以下のように展開できる．

$$\begin{aligned} \text{SSD}_{b,f}(u, v) &= \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} (b(x, y))^2 \\ &\quad - 2 \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} b(x, y) f(u+x, v+y) + P_{f^2}(u, v) \end{aligned} \quad (7)$$

ここで,

$$P_{f^2}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \{f(u+x, v+y)\}^2 \quad (8)$$

である．また, 式 (7) の右辺第 2 項は以下のように表現できる．

$$\text{cor}_{g_b,f}(u, v) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g_b(x, y) f(u+x, v+y) \quad (9)$$

$$= \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} b(x, y) f(u+x, v+y) \quad (10)$$

ここで,

$$(u, v) \in \mathbf{R}_{\text{IN}}$$

$$g_b(x, y) = \begin{cases} b(x, y) & (x, y) \in \mathbf{S} \\ 0 & (x, y) \in \mathbf{T} \end{cases} \quad (11)$$

$$\mathbf{S} = \{(x, y) | x = 0, 1, \dots, A-1 \text{ and} \\ y = 0, 1, \dots, B-1\}$$

$$\mathbf{T} = \{(x, y) | x = A, A+1, \dots, M-1 \text{ or} \\ y = B, B+1, \dots, N-1\}$$

であり, $g_b(x, y)$ と $f(x, y)$ は周期 (M, N) を仮定している．また, $\text{cor}_{g_b,f}(u, v)$ は $g_b(x, y)$ と $f(x, y)$ との周期的相互相関を意味する．

次に, 別の誤差尺度 $E_{b,f}(u, v)$ を以下のように導入する．

$$E_{b,f}(u, v) = P_{f^2}(u, v) - 2\text{cor}_{g_b,f}(u, v) \quad (12)$$

式 (7) の右辺第 1 項は (u, v) と独立であるため,

$E_{b,f}(u, v)$ が最小となる (u, v) と $SSD_{b,f}(u, v)$ が最小となる (u, v) は等しい . つまり ,

$$SSD_{b,f}(u_0, v_0) = \min_{(u,v) \in \mathbf{R}_{IN}} (SSD_{b,f}(u, v)) \quad (13)$$

$$E_{b,f}(u'_0, v'_0) = \min_{(u,v) \in \mathbf{R}_{IN}} (E_{b,f}(u, v)) \quad (14)$$

$$u'_0 = u_0 \quad \text{and} \quad v'_0 = v_0 \quad (15)$$

が成立する . したがって , 式 (12) の最小値を求めることによって , ブロックマッチングを実行することができる .

$P_{f^2}(u, v)$ は次のように再帰的な計算により効率的に算出可能である [9] . まず ,

$$\bar{P}_{f^2}(u, v) = \sum_{x=0}^{A-1} \{f(u+x, v)\}^2 \quad (16)$$

$$= \begin{cases} \sum_{x=0}^{A-1} \{f(x, v)\}^2 & u = 0 \\ \bar{P}_{f^2}(u-1, v) + \{f(u+A-1, v)\}^2 \\ \quad - \{f(u-1, v)\}^2 & u \geq 1 \end{cases} \quad (17)$$

$$u = 0, 1, \dots, M-A, v = 0, 1, \dots, N-1$$

を求める . 次に , $P_{f^2}(u, v)$ を式 (8) から ,

$$P_{f^2}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \{f(u+x, v+y)\}^2 \quad (18)$$

$$= \begin{cases} \sum_{y=0}^{B-1} \bar{P}_{f^2}(u, y) & v = 0 \\ P_{f^2}(u, v-1) + \bar{P}_{f^2}(u, v+B-1) \\ \quad - \bar{P}_{f^2}(u, v-1) & v \geq 1 \end{cases} \quad (19)$$

$$u = 0, 1, \dots, M-A, v = 0, 1, \dots, N-B$$

と計算する . 以上のように , $\{f(x, y)\}^2$ から一つの (u, v) における $P_{f^2}(u, v)$ を計算する場合 , 直接計算では $AB-1$ 回の実数加算が必要である . 一方 , 上式では , $4 + \frac{B-3}{N} + \frac{A-3}{M} \approx 4$ ($B \ll N, A \ll M$ のとき) 回の実数加算のみでよい . 更に , これらは $b(x, y)$ とは独立であり , $f(x, y)$ によってのみ決まる値である .

また , $(u, v) \in \mathbf{R}_{IN}$ における $\text{cor}_{g_b, f}(u, v)$ は FFT を用いて高速に算出可能である . ただし , 推定精度は

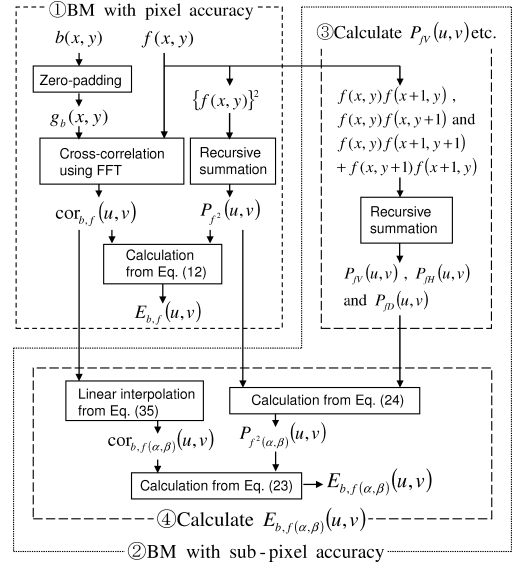


図 1 $E_{b,f,\alpha,\beta}(u, v)$ の計算手順
Fig. 1 Flow of calculation of $E_{b,f,\alpha,\beta}(u, v)$.

ピクセル精度に限定されている . 以上の手順は , 図 1 の①において要約されている .

2.3 サブピクセルブロックマッチング

サブピクセル精度のブロックマッチングでは , 一般に画素間に新たな画素を補間により生成し , それを用いてマッチングが実行される . 図 2 のように , $(u, v) \sim (u+1, v+1)$ 間を $\alpha : (1-\alpha), \beta : (1-\beta)$ ($0 \leq \alpha < 1, 0 \leq \beta < 1$) に分割した点での補間値を

$$f_{(u,v,\alpha,\beta)}(x, y), x=0, 1, \dots, A-1, y=0, 1, \dots, B-1 \quad (20)$$

$$0 \leq u + \alpha \leq M - A \quad \text{and} \quad 0 \leq v + \beta \leq N - B$$

とすると , それを用いて計算した二乗誤差和 $SSD_{b,f(\alpha,\beta)}(u, v)$ は

$$SSD_{b,f(\alpha,\beta)}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} (b(x, y) - f_{(u,v,\alpha,\beta)}(x, y))^2 \quad (21)$$

となる .

画像の補間法として , K 点を用いて線形結合により一つの点 $f_{(u,v,\alpha,\beta)}(x, y)$ を生成する場合 ,

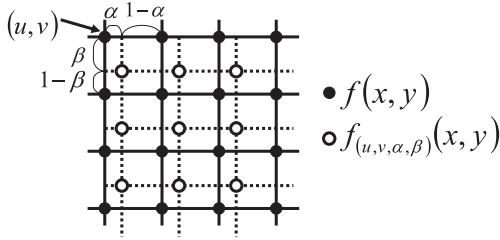


図 2 画素間の補間

Fig. 2 Generation sub-pixels.

$$f_{(u,v,\alpha,\beta)}(x, y) = \sum_{l=1}^K \gamma_l f(u + x_l + x, v + y_l + y) \quad (22)$$

$$\gamma_1, \dots, \gamma_K \in \mathbf{R}, \quad x_1, \dots, x_K, y_1, \dots, y_K \in \mathbf{Z}$$

と表現される。ここで、 \mathbf{R} は実数の集合である。代表的な補間法である双線形補間の場合、 $K = 4$ であり、パラメータはそれぞれ

$$\begin{aligned} \{\gamma_1, x_1, y_1\} &= \{(1-\alpha)(1-\beta), 0, 0\} \\ \{\gamma_2, x_2, y_2\} &= \{\alpha(1-\beta), 1, 0\} \\ \{\gamma_3, x_3, y_3\} &= \{(1-\alpha)\beta, 0, 1\} \\ \{\gamma_4, x_4, y_4\} &= \{\alpha\beta, 1, 1\} \end{aligned} \quad (23)$$

と与えられる。

3. 提案法

ここでは、補間法として双線形補間を用いた際のサブピクセルブロックマッチングの高速アルゴリズムを提案する。

3.1 サブピクセルブロックマッチングの高速化
補間した画素値 $f_{(u,v,\alpha,\beta)}(x, y)$ を用いた SSD は、式 (21) から、以下のように展開できる。

$$\begin{aligned} \text{SSD}_{b,f_{(\alpha,\beta)}}(u, v) &= \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} (b(x, y))^2 - 2\text{cor}_{g_b, f_{(\alpha,\beta)}}(u, v) \\ &\quad + P_{f^2(\alpha,\beta)}(u, v) \end{aligned} \quad (24)$$

ここで、

$$\text{cor}_{g_b, f_{(\alpha,\beta)}}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} b(x, y) f_{(u,v,\alpha,\beta)}(x, y) \quad (25)$$

$$P_{f^2(\alpha,\beta)}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \{f_{(u,v,\alpha,\beta)}(x, y)\}^2 \quad (26)$$

である。誤差尺度 $E_{b,f_{(\alpha,\beta)}}(u, v)$ を

$$E_{b,f_{(\alpha,\beta)}}(u, v) = P_{f^2(\alpha,\beta)}(u, v) - 2\text{cor}_{g_b, f_{(\alpha,\beta)}}(u, v) \quad (27)$$

とすると、ピクセル精度の場合と同様に、 $\text{SSD}_{b,f_{\alpha,\beta}}(u, v)$ を最小にする (u, v) と $E_{b,f_{(\alpha,\beta)}}(u, v)$ を最小にする (u, v) は等しい。

(1) $P_{f^2(\alpha,\beta)}(u, v)$ の計算

空間領域で画素を補間して $f_{(u,v,\alpha,\beta)}(x, y)$ を生成し、 $P_{f^2(\alpha,\beta)}(u, v)$ を直接計算する方法が考えられる。しかし、本論文では、 $P_{f^2}(u, v)$ の補間によって $P_{f^2(\alpha,\beta)}(u, v)$ を計算できることを述べる。

$P_{f^2(\alpha,\beta)}(u, v)$ は、式 (22)、(23)、(26) から以下のように展開できる。

$$\begin{aligned} P_{f^2(\alpha,\beta)}(u, v) &= \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \{f_{(u,v,\alpha,\beta)}(u+x, v+y)\}^2 \\ &= \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \left[(1-\alpha)(1-\beta)f(u+x, v+y) \right. \\ &\quad \left. + \alpha(1-\beta)f(u+x+1, v+y) \right. \\ &\quad \left. + (1-\alpha)\beta f(u+x, v+y+1) \right. \\ &\quad \left. + \alpha\beta f(u+x+1, v+y+1) \right]^2 \\ &= (1-\alpha)^2 \left[(1-\beta)^2 P_{f^2}(u, v) + \beta^2 P_{f^2}(u, v+1) \right. \\ &\quad \left. + 2\beta(1-\beta) P_{fH}(u, v) \right] \\ &\quad + \alpha^2 \left[(1-\beta)^2 P_{f^2}(u+1, v) + \beta^2 P_{f^2}(u+1, v+1) \right. \\ &\quad \left. + 2\beta(1-\beta) P_{fH}(u+1, v) \right] \\ &\quad + 2\alpha(1-\alpha) \left[(1-\beta)^2 P_{fV}(u, v) + \beta^2 P_{fV}(u, v+1) \right. \\ &\quad \left. + \beta(1-\beta) P_{fD}(u, v) \right] \end{aligned} \quad (28)$$

ここで、

$$P_{f^2}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \{f(u+x, v+y)\}^2 \quad (29)$$

$$P_{fV}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} f(u+x, v+y) f(u+x+1, v+y) \quad (30)$$

$$P_{fH}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} f(u+x, v+y) f(u+x, v+y+1) \quad (31)$$

$$P_{fD}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} \left[f(u+x, v+y) f(u+x+1, v+y+1) + f(u+x+1, v+y) f(u+x, v+y+1) \right] \quad (32)$$

である．ピクセル精度の推定を 2.2 の方法によって実行した場合， $P_{f^2}(u, v)$ は計算済みである．一方， $P_{fV}(u, v) \sim P_{fD}(u, v)$ はサブピクセル精度の推定時に計算する必要がある．しかし，これらは $P_{f^2}(u, v)$ と同様の再帰的計算（2.2 参照）により，効率的に計算できる．

(2) $\text{cor}_{g_b, f(\alpha, \beta)}(u, v)$ の計算

$\text{cor}_{g_b, f(\alpha, \beta)}(u, v)$ は，式 (22)，(25) から，

$$\text{cor}_{g_b, f(\alpha, \beta)}(u, v) = \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} b(x, y) f(u, v, \alpha, \beta)(x, y) \quad (33)$$

$$\begin{aligned} &= \sum_{y=0}^{B-1} \sum_{x=0}^{A-1} b(x, y) \sum_{l=1}^K \gamma_l f(u+x_l+x, v+y_l+y) \\ &= \sum_{l=1}^K \gamma_l \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g_b(x, y) f(u+x_l+x, v+y_l+y) \\ &= \sum_{l=1}^K \gamma_l \text{cor}_{g_b, f}(u+x_l, v+y_l) \end{aligned} \quad (34)$$

となる．ゆえに，ピクセル精度の推定で既に計算済みの $\text{cor}_{g_b, f}(u, v)$ を補間することで， $\text{cor}_{g_b, f(\alpha, \beta)}(u, v)$ を計算できることが分かる．補間法として双線形補間を用いる場合， $\text{cor}_{g_b, f}(u, v)$ から 1 組の (u, v, α, β) における $\text{cor}_{g_b, f(\alpha, \beta)}(u, v)$ を計算するのに必要な計算量

は，5 回の実数加算と 8 回の実数乗算のみである．

3.2 提案法のフローチャート

以上の議論をまとめ，提案法の手順を説明する．図 1 は，ある (u, v, α, β) におけるサブピクセル精度の誤差尺度 $E_{b, f(\alpha, \beta)}(u, v)$ を求めるまでのフローチャートである．以下では， $i, \bar{u}_i, \bar{v}_i, L \in \mathbf{Z}$ ， $\alpha_i, \beta_i \in \mathbf{R}$ ， $0 \leq \alpha_i < 1, 0 \leq \beta_i < 1$ とする．

1. 2.2 の方法でピクセル精度の推定を実行，得られたシフト量を (u_0, v_0) とする
2. $P_{fV}(u, v) \sim P_{fD}(u, v)$ を計算
3. サブピクセル精度で推定を行う (u, v, α, β) の組を複数組 $(\bar{u}_i, \bar{v}_i, \alpha_i, \beta_i)$ ， $(i = 1, 2, \dots, L)$ 決定
4. $i = 0$ のとき， $(\bar{u}_0, \bar{v}_0, \alpha_0, \beta_0) = (u_0, v_0, 0, 0)$ ， $E_{b, f(\alpha_0, \beta_0)}(\bar{u}_0, \bar{v}_0) = E_{b, f}(u_0, v_0)$ とする
5. $i = 1, \dots, L$ について以下のように $E_{b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ を計算
 - 5-1. 式 (28) から $P_{f^2(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ を計算
 - 5-2. $\text{cor}_{g_b, f}(u, v)$ を補間し $\text{cor}_{g_b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ を計算
 - 5-3. 式 (27) から $E_{b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ を計算
6. $E_{b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ が最小となる i を求める

最終的なシフト量は，

$$E_{b, f(\alpha_{i_0}, \beta_{i_0})}(\bar{u}_{i_0}, \bar{v}_{i_0}) = \min_i (E_{b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)) \quad (35)$$

のとき， $(\bar{u}_{i_0} + \alpha_{i_0}, \bar{v}_{i_0} + \beta_{i_0})$ である．

L はサブピクセル精度の推定点数である．図 3 のように，ピクセル精度の推定結果から ± 0.5 画素以内を $1/2, 1/4, 1/8$ 画素精度で全探索する場合， L はそれぞれ 8, 24, 80 となる． L はサブピクセル精度のマッチングがどれだけ細かく行われるかの尺度と考えることができ，マッチング精度は L とサブピクセル精度の推定位置の配置によって決まる．

図 1 において，③の部分の計算は L によらず 1 度だけ計算すればよい．それぞれの (u, v, α, β) の組において必要な計算は④の部分のみである．以上により，空間領域で画素を補間する従来法 [11], [12] と比較して， L が大きいほど提案法はより有効となる．

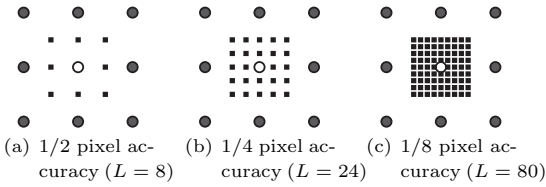


図3 サブピクセル精度の推定位置 (●:ピクセル精度の推定位置, ○:ピクセル精度の推定結果, ■:サブピクセル精度の推定位置)

Fig. 3 Matching points with sub-pixel accuracy (●: matching points with pixel accuracy, ○: conclusive points with pixel accuracy, ■: matching points with sub-pixel accuracy).

3.3 計算量

ここでは、フレーム間でブロックマッチングを実行し、予測画像を生成する場合を例にして、計算量について考察する。フレームサイズを $M' \times N'$ 、ブロックサイズを $A \times B$ 、探索領域のサイズを $M \times N$ とする。今、議論を簡単にするために、 M, N はともに2のべき乗、 $M'/A, N'/B$ はともに整数とする。このとき、総ブロック数は $D = \frac{M'N'}{AB}$ となる。また、補間には双線形補間を用い、各ブロックでのブロックマッチングにおけるサブピクセル精度の推定点数を L とする。サブピクセル精度の推定に必要な計算量を、補間によって画素を生成し二乗誤差和を計算していく従来法と、提案法それぞれについて計算する。

表1は従来法でのサブピクセル精度のブロックマッチングに必要な計算量を示している。一つのマクロブロック、1組の (u, v, α, β) における二乗誤差和を計算するには、画素の補間と誤差の計算が必要である。これを一つのマクロブロックにつき L 組、1フレームにつき D ブロック計算する必要がある。したがって、1フレームに対し、従来法を用いてサブピクセルブロックマッチングを行うのに必要な実数加算回数 C_{sum} 、実数乗算回数 C_{multi} はそれぞれ、

$$C_{sum} = DL(5AB + 1) \quad (36)$$

$$C_{multi} = DL(5AB + 4) \quad (37)$$

となる。

表2は提案法でのサブピクセル精度のブロックマッチングに必要な計算量を示している。2.2の方法でピクセル単位の推定結果を得た場合、 $P_{f_2}(u, v)$ 、 $cor_{g_b, f}(u, v)$ は計算済みである。次に、すべてのマクロブロック、すべての (u, v, α, β) の組に対し共通して

表1 従来法における計算量 (“a set” は1組の (u, v, α, β) を表す)

Table 1 Complexities for the general method (“a set” means a set of (u, v, α, β)).

	Additions	Multiplications
Interpolation per a set	$3AB + 2$	$4AB + 4$
Calculating SSD per a set	$2AB - 1$	AB
Total number per a set	$5AB + 1$	$5AB + 4$
Total number per a frame	$DL(5AB + 1)$	$DL(5AB + 4)$

表2 提案法における計算量 (“a set” は1組の (u, v, α, β) を表す)

Table 2 Complexities for the proposed method (“a set” means a set of (u, v, α, β)).

	Additions	Multiplications
Calculating $f(x, y)f(x + 1, y)$ etc per a frame	$M'N'$	$4M'N'$
Calculating $P_{fV}(u, v)$ etc per a frame	$12M'N' + 3(B-3)M' + 3(A-3)N'$	0
Calculating E per a set	16	30
Total number per a frame	$16DL + 13M'N' + 3(B-3)M' + 3(A-3)N'$	$30DL + 4M'N'$

いる $P_{fV}(u, v) \sim P_{fD}(u, v)$ と、それらを生成するための $f(x, y)f(x + 1, y)$ などの計算が必要である。また、一つのマクロブロック、1組の (u, v, α, β) における誤差尺度 $E_{b, f(\alpha, \beta)}(u, v)$ の計算を、一つのマクロブロックにつき L 組、1フレームにつき D ブロック計算する必要がある。したがって、1フレームに対し、提案法を用いてサブピクセルブロックマッチングを行うのに必要な実数加算回数 P_{sum} 、実数乗算回数 P_{multi} はそれぞれ、

$$P_{sum} = 16DL + 13M'N' + 3(B-3)M' + 3(A-3)N' \quad (38)$$

$$P_{multi} = 30DL + 4M'N' \quad (39)$$

となる。

図4は $M' = 640, N' = 480, A = B = 16, M = N = 32$ とし、サブピクセル精度の推定点数 L を変化した場合の計算量の比較である。ここで、計算量は実数加算回数と実数乗算回数の和としている。提案法では L によらず $P_{fV}(u, v)$ の算出などに計算量が必要である。しかし、空間領域で画素を補間する従来法と比較して、 L に対する増分が非常に少ない。そのため、 L が大きいほど計算量の差も大きくなる。例えば1/4ピクセル精度においては、提案法の計算量は従来法の

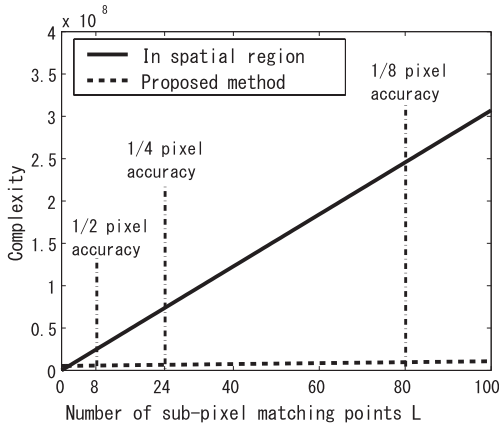


図 4 計算量の比較 ($M' = 640, N' = 480, A = B = 16, M = N = 32$)

Fig. 4 Comparison of the complexities ($M' = 640, N' = 480, A = B = 16, M = N = 32$).

1/10 程度である。

4. シミュレーション

ここでは、提案法の有効性を確認するために、フレーム間でブロックマッチングを実行し、予測画像を生成する場合を例にして、シミュレーションを行う。

4.1 シミュレーション条件

シミュレーションでは動画 caltrain (図 5(a), 8 ビット濃淡, 512×400), forest (図 5(b), 8 ビット濃淡, 640×480) を使用した。2 フレーム目を対象フレーム, 1 フレーム目を参照フレームとして予測フレームを生成し、PSNR とブロックマッチングに要した処理時間 (サブピクセル精度計算のみ) を測定した。また、参考として、ピクセル精度のブロックマッチングに要した処理時間も測定した。ブロックサイズは $A = B = 16, 8$ の 2 種類とし、探索領域のサイズは $M = N = 32$ とした。サブピクセル精度の推定は、双線形補間を用い、図 3 のように、ピクセル精度の推定結果の ± 0.5 ピクセル内を $1/2, 1/4, 1/8$ ピクセル精度で推定した。補間により画素を生成し、SSD を計算する従来法と提案法との処理時間を比較した。シミュレーション環境を表 3 に示す。なお、提案法の計算量は、画像の種類によらないことに注意する。

4.2 シミュレーション結果

表 4 は、caltrain, forest を用い、ブロックサイズが $A = B = 16$ の場合の処理時間を比較したものである。この表より、まず整数ピクセル精度のブロック



(a) caltrain (1 frame). (b) forest (1 frame).

図 5 使用動画

Fig. 5 Video sequences.

表 3 シミュレーション環境

Table 3 Simulation environments.

Language	C
CPU	intel Core2Duo 3.6 GHz
Memory	2.0GB

表 4 処理時間 (milli-second) の比較, フレームサイズ: 512×400 (caltrain), 640×480 (forest), マクロブロックサイズ: 16×16 , 探索領域サイズ: 32×32

Table 4 Comparison of the processing speed (milli-second), frame size: 512×400 (caltrain), 640×480 (forest), block size: 16×16 , search window size: 32×32 .

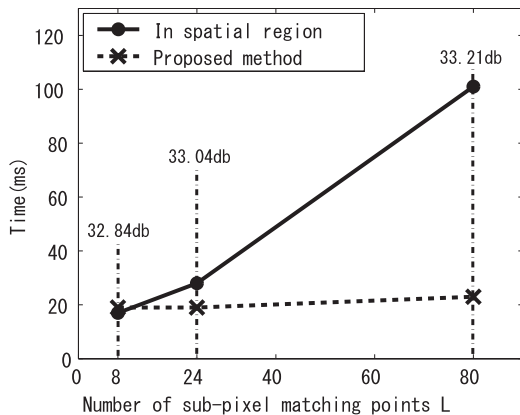
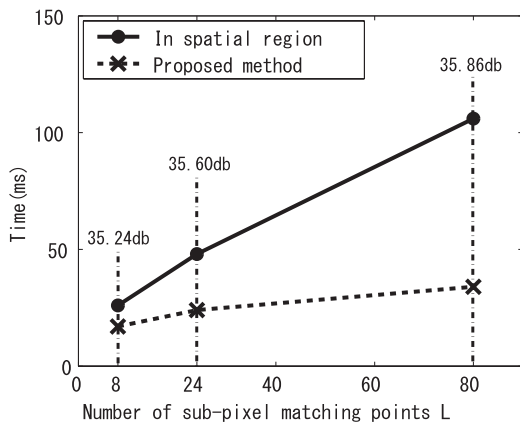
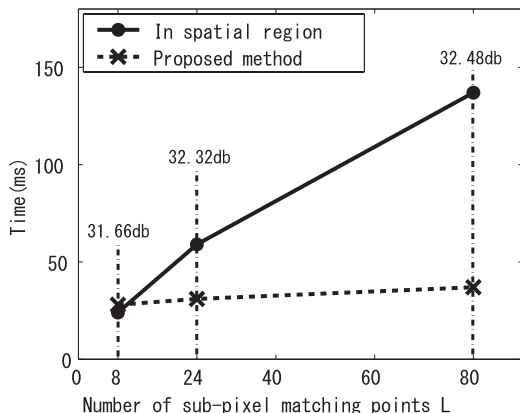
Movie		caltrain			forest		
Integer pixel	①Direct full search	751			1137		
	②Lit. [9]'s	163			246		
	②/①	0.22			0.22		
Sub-pixel	Accuracy	1/2	1/4	1/8	1/2	1/4	1/8
	③In spatial region	17	28	101	24	59	137
	④Proposed	19	19	23	28	31	37
	④/③	1.12	0.68	0.23	1.17	0.53	0.27

マッチングにおいては、[9] の方法は空間領域における全探索と比較して 5 倍程度高速であることが分かる。また、サブピクセル精度のブロックマッチングにおいて、従来法の処理時間に対する提案法の処理時間の割合は、それぞれ $1/4$ ピクセル精度で 68%, 53%, $1/8$ ピクセル精度で 23%, 27% であった。

図 6(a) は表 4 の “caltrain”, “Sub-pixel” の部分をグラフにしたものである。これより、 L が大きいほど提案法が有効であることが分かる。

図 6(b) は caltrain を用い、ブロックサイズが $A = B = 8$ の場合の処理時間を比較したものである。これより、異なるブロックサイズにおいても提案法が有効であることが分かる。

図 6(c) は表 4 の “forest”, “Sub-pixel” の部分をグラフにしたものである。これより、入力信号に依存せず、提案法が有効であることが分かる。

(a) caltrain, block size: $A = B = 16$.(b) caltrain, block size: $A = B = 8$.(c) forest, block size: $A = B = 16$.図 6 従来法と提案法の比較 (探索領域サイズ $M = N = 32$)Fig. 6 Comparison between general method and proposed method (search window size: $M = N = 32$).

5. むすび

本論文では、二乗誤差和 SSD を誤差尺度とするサブピクセルブロックマッチングの高速アルゴリズムを提案した。提案法は、整数精度のブロックマッチング時に計算されたいくつかのパラメータを直接補間することでサブピクセル精度のブロックマッチングを実行する方法である。双線形補間を例にして、補間により画素を生成し SSD を計算する従来法と提案法が同精度であることを解析的に証明した。また、計算量の評価やシミュレーションを行い、従来法と比べ提案法が低演算量であること、サブピクセル精度の推定点数が多いほど有効であること、画像の内容によらないことを確認した。

今後は、他の補間法や異なる誤差尺度についても検討を進める予定である。

文 献

- [1] 河合紀彦, 佐藤智和, 横矢直和, “テクスチャ明度変化と局所性を考慮したパターン類似度を用いたエネルギー最小化による画像修復,” 信学論 (D), vol. J91-D, no. 9, pp. 2293–2304, Sept. 2008.
- [2] Y. Wexler, E. Shechtman, and M. Irani, “Space-time completion of video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, March 2007.
- [3] T. Koga, K. Iinuma, A. Hirano, and Y. Iijima, “Motion compensated interframe coding for video conferencing,” *NTC 81 Conf. Record, IEEE*, pp. G5. 3. 1–G5. 3. 5, 1981.
- [4] S. Zhu and K. Ma, “A new diamond search algorithm for fast block motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, 2000.
- [5] W. Li and E. Salari, “Successive elimination algorithm for motion estimation,” *IEEE Trans. Image Process.*, vol. 4, no. 1, pp. 105–107, Jan. 1995.
- [6] Y. Noguchi, J. Furukawa, and H. Kiya, “A fast full search block matching algorithm for MPEG-4 video,” *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 61–65, 1999.
- [7] F. Essannouni, R.O.H. Thami, D. Aboutajdine, and A. Salam, “Simple noncircular correlation method for exhaustive sum square difference matching,” *Opt. Eng.*, vol. 46, no. 10, 107004, Oct. 2007.
- [8] S.L. Kilthau, M.S. Drew, and T. Moller, “Full search content independent block matching based on the fast fourier transform,” *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 669–672, 2002.
- [9] Z. Li, A. Uemura, and H. Kiya, “An FFT-based full-search block-matching algorithm with SSD criterion,” *Proc. APSIPA ASC2009*, pp. 457–460, Oct. 2009.
- [10] Z. Li and H. Kiya, “Double-search-window block

matching using the fast fourier transform,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, no.IVMSP-P11.PG8, Dallas, TX, the U.S., March 2010.

- [11] H. Nisar and T. Choi, “Fast and efficient fractional pixel motion estimation for H.264/AVC video coding,” Proc. IEEE Int. Conf. Image Process., pp.1561–1564, 2009.
- [12] P.R. Hill and D.R. Bull, “Kernel based sub-pixel motion estimation,” Proc. IEEE Int. Conf. Image Process., pp.1557–1560, 2009.
- [13] Q. Zhang, Y. Dai, and C. Kuo, “Fast sub-pel motion vector prediction via block classification,” Proc. IEEE Int. Conf. Image Process., pp.1569–1572, 2009.
- [14] M. Sayed, W. Badawy, and G. Jullien, “Low-complexity algorithm for fractional-pixel motion estimation,” Proc. IEEE Int. Conf. Image Process., pp.1565–1568, 2009.
- [15] L. Bo, Z. Jinyin, and S. Jianbin, “A fast sub-pel motion estimation algorithm based on best position calculation,” Int. J. Comput. Sci. Netw. Security, vol.6, no.2B, pp.21–28, Feb. 2006.
- [16] 植村淳志, 李 禎, 貴家仁志, “FFT 全探索ブロックマッチング法のサブピクセル精度への拡張” 第 24 回信号処理シンポジウム講演論文集, pp.447–452, Nov. 2009.

(平成 22 年 1 月 26 日受付, 5 月 16 日再受付)



貴家 仁志 (正員:フェロー)

1982 長岡技科大大学院修士課程了。同年, 都立大・工・電気助手。2000 同大学院工学研究科電気教授。改組により, 2005 首都大・システムデザイン・情報通信システム教授, 現在に至る。工博。1995–1996 豪シドニー大 Visiting Fellow。デジタル信号処理, メディア工学, 情報セキュリティの研究・教育に従事。1998–2002 本会論文誌編集委員。1998–2000 IEEE 論文誌 Signal Processing 編集委員。2008 本会論文賞受賞。2008–2009 本会基礎・境界ソサイエティ誌編集委員長, 次期ソサイエティ会長(理事)。アジア太平洋・信号情報処理学会 (APSIPA) 副会長(理事)。



植村 淳志 (正員)

2008 都立大・工・電子情報卒。2010 首都大学東京大学院システムデザイン研究科博士前期課程了。同年テクノマセマティカル(株)入社。デジタル信号処理に関する研究について興味をもつ。



李 禎 (学生員)

2009 首都大東京・システムデザイン・情報通信システム工学コース卒。現在, 同大学院システムデザイン研究科博士前期課程に在学中。デジタル信号処理に関する研究について興味をもつ。