

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

An Extension of FFT-Based Full-Search Block-Matching Method to Sub-pixel Accuracy

植村 淳志[†]

李 禎[†]

貴家 仁志[†]

[†] 首都大学東京大学院 システムデザイン研究科 情報通信システム工学専修

Atsushi UEMURA[†]

Zhen LI[†]

Hitoshi KIYA[†]

[†]Dept. of Information and Communication Systems Engineering, Tokyo Metropolitan University

アブストラクト

本稿では、ピクセル精度の推定に限定されていたFFT全探索ブロックマッチング法を、サブピクセル精度へと拡張する。従来法がサブピクセルの画素を補間によって生成して実行されるのに対して、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、計算量の削減を実現している。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シミュレーションにより確認した。

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など多くの分野で応用されている技術である。特に動画像符号化においては、圧縮効率はブロックマッチングの推定精度に依存し、符号化処理全体の処理時間のうち大部分をブロックマッチングの処理時間が占めている。このため、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 [1]–[5]。その中でも、FFTによる相関計算に基づく方法 [4], [5] は全探索の精度を維持しており、かつ低演算量である。しかし、この方法は推定精度がピクセル精度に限定されていた。

マッチングにおいては、精度向上のため、1/2画素精度、1/4画素精度などサブピクセル精度のマッチングが求められている。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 [6]–[9]。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 [6] であるが、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 [7]–[9] 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。

本稿では、ピクセル精度の推定に限定されていた [4], [5] の方法をサブピクセル精度へと拡張する。具体的には、[4], [5] 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方法を提案する。この方法は、画像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。シミュレーションにより提案法の有効性を確認するとともに、計算量についても言及する。

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。

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。但し、 \mathbf{Z} は整数全体の集合である。本稿では、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 difference, SAD) と二乗誤差和 (sum of squared difference, SSD) の2つの基準が使われている。それぞれ

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

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

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

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

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, 可能なシフト量全てについて調べることを全探索と呼ぶ. 本稿では, SSD を誤差尺度とする全探索ブロックマッチングについて考察する.

2.2 全探索ブロックマッチングの高速アルゴリズム [4]

周期 (M, N) を持つ二次元信号 $c(x, y)$ と $g(x, y)$ との周期的相互相関 $\widetilde{\text{cor}}_{c,g}(u, v)$ は, DFT, IDFT を用いて以下のように表現される.

$$C(k_1, k_2) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} c(x, y) W_M^{xk_1} W_N^{yk_2} \quad (7)$$

$$G(k_1, k_2) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g(x, y) W_M^{xk_1} W_N^{yk_2} \quad (8)$$

$$(k_1 = 0, 1, \dots, M-1, k_2 = 0, 1, \dots, N-1)$$

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

$$= \frac{1}{MN} \sum_{k_2=0}^{N-1} \sum_{k_1=0}^{M-1} \overline{C(k_1, k_2)} G(k_1, k_2) W_M^{-uk_1} W_N^{-vk_2} \quad (10)$$

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

ここで, $\overline{C(k_1, k_2)}$ は $C(k_1, k_2)$ の複素共役であり, $W_M = \exp(-j2\pi/M)$, $W_N = \exp(-j2\pi/N)$ である. 従って, $\widetilde{\text{cor}}_{c,g}(u, v)$ は, FFT を用いて高速に演算することができる.

次に, 以下のように領域を定義する.

$$\mathbf{Q} = \{(x, y) | x = 0, 1, \dots, M-1, y = 0, 1, \dots, N-1\} \quad (11)$$

$$\mathbf{S} = \{(x, y) | x = 0, 1, \dots, A-1, y = 0, 1, \dots, B-1\} \quad (12)$$

$$\mathbf{T} = \mathbf{Q} \setminus \mathbf{S} \quad (13)$$

さらに, 対象ブロック $b(x, y)$ に対し, 以下のような関数を定義する.

$$g_b(x, y) = \begin{cases} b(x, y) & (x, y) \in \mathbf{S} \\ 0 & (x, y) \in \mathbf{T} \end{cases} \quad (14)$$

式 (4) の $\text{SSD}_{b,f}(u, v)$ は, 以上を用いて,

$$\begin{aligned} \text{SSD}_{b,f}(u, v) &= \sum_{(x,y) \in \mathbf{Q}} \{g_b(x, y)\}^2 - 2\widetilde{\text{cor}}_{g_b,f}(u, v) \\ &+ \sum_{(x,y) \in \mathbf{S}} \{f(u+x, v+y)\}^2 \end{aligned} \quad (15)$$

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

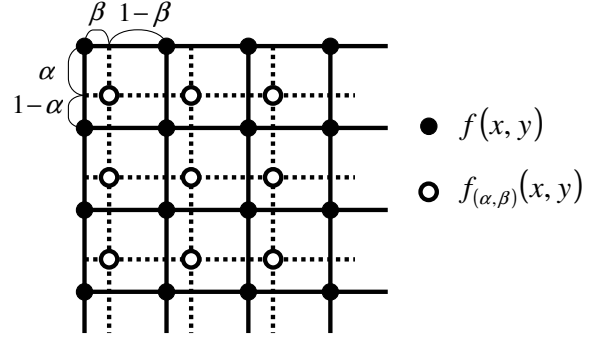


図 1: 画素間の補間

と表現できる. $\widetilde{\text{cor}}_{g_b,f}(u, v)$ の定義域は $(u, v) \in \mathbf{Q}$ であるが, 式 (15) は \mathbf{R}_{IN} を定義域とする. 式 (15) の右辺第三項は, 再帰的な計算あるいは FFT を用いた相互相関として効率的に算出が可能である. また, 右辺第一項は u, v によらず一定である. 従って,

$$E_{b,f}(u, v) = P_{f^2}(u, v) - 2\widetilde{\text{cor}}_{g_b,f}(u, v) \quad (16)$$

$$P_{f^2}(u, v) = \sum_{(x,y) \in \mathbf{S}} \{f(u+x, v+y)\}^2 \quad (17)$$

とおき,

$$\text{SSD}_{b,f}(u_0, v_0) = \min_{(u,v) \in \mathbf{R}_{\text{IN}}} (\text{SSD}_{b,f}(u, v)) \quad (18)$$

$$E_{b,f}(u'_0, v'_0) = \min_{(u,v) \in \mathbf{R}_{\text{IN}}} (E_{b,f}(u, v)) \quad (19)$$

を比較すると,

$$u'_0 = u_0 \quad \text{and} \quad v'_0 = v_0 \quad (20)$$

が成立する. $\text{SSD}_{b,f}(u, v)$ の代わりに誤差尺度 $E_{b,f}(u, v)$ を計算することによって, 全探索ブロックマッチングを実行できることがわかる.

2.3 サブピクセルブロックマッチング

前節までのブロックマッチングでは, 推定精度はピクセル精度に限定されていた. 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 1 のように, $(x, y) \sim (x+1, y+1)$ 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 $f_{(\alpha, \beta)}(x, y)$ とすると, それを用いて計算した二乗誤差和 $\text{SSD}_{b, f_{(\alpha, \beta)}}(u, v)$ は

$$\text{SSD}_{b, f_{(\alpha, \beta)}}(u, v) = \sum_{(x,y) \in \mathbf{S}} (b(x, y) - f_{(\alpha, \beta)}(u+x, v+y))^2 \quad (21)$$

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 $f_{(\alpha,\beta)}(x,y)$ を生成する場合、

$$f_{(\alpha,\beta)}(x,y) = \sum_{l=1}^K \gamma_l f(x+i_l, y+j_l) \quad (22)$$

$$\gamma_1, \dots, \gamma_K \in \mathbf{R}, \quad i_1, \dots, i_K, j_1, \dots, j_K \in \mathbf{Z}$$

と表現される。ここで、 \mathbf{R} 、 \mathbf{Z} はそれぞれ実数、整数全体の
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

$$\begin{aligned} \{\gamma_1, i_1, j_1\} &= \{(1-\alpha)(1-\beta), 0, 0\} \\ \{\gamma_2, i_2, j_2\} &= \{\alpha(1-\beta), 1, 0\} \\ \{\gamma_3, i_3, j_3\} &= \{(1-\alpha)\beta, 0, 1\} \\ \{\gamma_4, i_4, j_4\} &= \{\alpha\beta, 1, 1\} \end{aligned} \quad (23)$$

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につい
て説明する。また、計算量についても考察する。

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 $f_{(\alpha,\beta)}(x,y)$ を用いた場合の誤差尺度
 $E_{b,f_{(\alpha,\beta)}}(u,v)$ は、式 (16) より、

$$E_{b,f_{(\alpha,\beta)}}(u,v) = P_{f^2_{(\alpha,\beta)}}(u,v) - 2\widetilde{\text{cor}}_{g_b, f_{(\alpha,\beta)}}(u,v) \quad (24)$$

となる。但し、

$$P_{f^2_{(\alpha,\beta)}}(u,v) = \sum_{(x,y) \in \mathbf{S}} \{f_{(\alpha,\beta)}(u+x, v+y)\}^2 \quad (25)$$

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

であり、 $\widetilde{\text{cor}}_{g_b, f_{(\alpha,\beta)}}(u,v)$ は $g_b(x,y)$ と $f_{(\alpha,\beta)}(x,y)$ との周期的
相互相関である。

まず、式 (24) 右辺第一項について考える。その計算と
しては、まず補間により $f_{(\alpha,\beta)}(x,y)$ を生成し、直接計算す
る方法が考えられる。しかし、本稿では、 $P_{f^2_{(\alpha,\beta)}}(u,v)$ を
 $P_{f^2}(u,v)$ の補間によって計算できることを指摘し、補間
によって $f_{(\alpha,\beta)}(x,y)$ を生成する必要がないことを述べる。

$P_{f^2_{(\alpha,\beta)}}(u,v)$ は、式 (22)、(23)、(25) によって以下のよう

$$\begin{aligned} P_{f^2_{(\alpha,\beta)}}(u,v) &= \sum_{(x,y) \in \mathbf{S}} \{f_{(\alpha,\beta)}(u+x, v+y)\}^2 \\ &= \sum_{(x,y) \in \mathbf{S}} \left[(1-\alpha)(1-\beta)f(u+x, v+y) \right. \\ &\quad + \alpha(1-\beta)f(u+x+1, v+y) \\ &\quad + (1-\alpha)\beta f(u+x, v+y+1) \\ &\quad \left. + \alpha\beta f(u+x+1, v+y+1) \right]^2 \quad (26) \\ &= (1-\alpha)^2 \left[(1-\beta)^2 P_{f^2}(u,v) + \beta^2 P_{f^2}(u,v+1) \right. \\ &\quad \left. + 2\beta(1-\beta)P_{fH}(u,v) \right] + \alpha^2 \left[(1-\beta)^2 P_{f^2}(u+1,v) \right. \\ &\quad \left. + \beta^2 P_{f^2}(u+1,v+1) + 2\beta(1-\beta)P_{fH}(u+1,v) \right] \\ &\quad + 2\alpha(1-\alpha) \left[(1-\beta)^2 P_{fV}(u,v) + \beta^2 P_{fV}(u,v+1) \right. \\ &\quad \left. + \beta(1-\beta)P_{fD}(u,v) \right] \quad (27) \end{aligned}$$

ここで、

$$P_{f^2}(u,v) = \sum_{(x,y) \in \mathbf{S}} \{f(u+x, v+y)\}^2 \quad (28)$$

$$P_{fV}(u,v) = \sum_{(x,y) \in \mathbf{S}} f(u+x, v+y)f(u+x+1, v+y) \quad (29)$$

$$P_{fH}(u,v) = \sum_{(x,y) \in \mathbf{S}} f(u+x, v+y)f(u+x, v+y+1) \quad (30)$$

$$P_{fD}(u,v) = \sum_{(x,y) \in \mathbf{S}} \left[f(u+x, v+y)f(u+x+1, v+y+1) \right. \\ \left. + f(u+x+1, v+y)f(u+x, v+y+1) \right] \quad (31)$$

である。第 2.2 節の方法でピクセル精度の推定を行った場
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が、再
帰的計算などを用いて効率的に計算できる。さらに、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。

次に、式 (24) の右辺第二項について考える。 $g_b(x,y)$ と
線形結合により生成された $f_{(\alpha,\beta)}(x,y)$ との周期的相互相関

$\widetilde{\text{cor}}_{g_b, f(\alpha, \beta)}(u, v)$ は、式 (22) より、

$$\widetilde{\text{cor}}_{g_b, f(\alpha, \beta)}(u, v) = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g_b(x, y) f(\alpha, \beta)(u+x, v+y) \quad (32)$$

$$= \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g_b(x, y) \sum_{l=1}^K \gamma_l f(u+i_l+x, v+j_l+y) \quad (33)$$

$$= \sum_{l=1}^K \gamma_l \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} g_b(x, y) f(u+i_l+x, v+j_l+y) \quad (34)$$

$$= \sum_{l=1}^K \gamma_l \widetilde{\text{cor}}_{g_b, f}(u+i_l, v+j_l) \quad (35)$$

となる．これによって、ピクセル精度の推定で得ている $\widetilde{\text{cor}}_{g_b, f}(u, v)$ を補間することで、 $\widetilde{\text{cor}}_{g_b, f(\alpha, \beta)}(u, v)$ を計算できることが分かる．

以上の議論をまとめ、提案法の手順を説明する．以下では、 $u, v, u_0, v_0, i, \bar{u}_i, \bar{v}_i, L \in \mathbf{Z}$, $\alpha_i, \beta_i \in \mathbf{R}$, $0 \leq \alpha_i < 1, 0 \leq \beta_i < 1$ とする．

1. 第 2.2 節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 (27) によって $P_{f^2(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ を計算
 - 5-2. $\widetilde{\text{cor}}_{g_b, f}(u, v)$ を補間し $\widetilde{\text{cor}}_{g_b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ を計算
 - 5-3. 式 (24) によって $E_{b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)$ 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、

$$E_{b, f(\alpha_{i_0}, \beta_{i_0})}(\bar{u}_{i_0}, \bar{v}_{i_0}) = \min_i (E_{b, f(\alpha_i, \beta_i)}(\bar{u}_i, \bar{v}_i)) \quad (36)$$

のとき、 $(\bar{u}_{i_0} + \alpha_{i_0}, \bar{v}_{i_0} + \beta_{i_0})$ である．ここで、 L はサブピクセル精度の推定点数である．図 2 のように、ピクセル精度の推定結果から ± 0.5 画素以内を $1/2, 1/4, 1/8$ 画素精度で推定する場合、 L はそれぞれ 8, 24, 80 となる．

3.2 計算量

ここでは、動き補償に利用されることを想定し、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．い

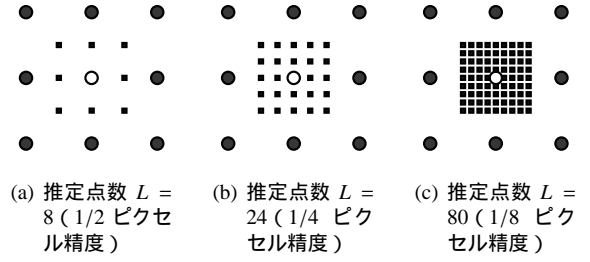


図 2: サブピクセル精度の推定位置 (●: ピクセル精度の推定位置, ○: ピクセル精度の推定結果, ■: サブピクセル精度の推定位置)

ま、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．また、 $P_{f^2}(u, v) \sim P_{fD}(u, v)$ の計算には再帰的な計算を用いる場合の計算量を算出した．全てのブロックにおいてピクセルレベルの推定結果が得られている状態から、サブピクセル精度の推定に必要な計算量を、補間によって画素値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．

まず、従来法における計算量を評価する．1 つの対象ブロック、一組の (u, v, α, β) における二乗誤差和を計算するには、 AB 点を補間により生成し、対象ブロックとの二乗誤差和を計算する必要がある．近傍 4 点を用いた双線形補間によって AB 点生成するには、式 (22) から、実数加算が $3AB$ 回、実数乗算が $4AB$ 回必要である．次に、 AB 点の二乗誤差和を計算するには、式 (21) から、実数加算が $2AB - 1$ 回、実数乗算が AB 回必要である．これを 1 ブロックにつき L 組、さらに 1 フレームにつき D ブロック計算する必要がある．従って、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、

$$C_{\text{sum}} = DL(5AB - 1) \quad (37)$$

$$C_{\text{multi}} = 5DLAB \quad (38)$$

となる．

次に、提案法の計算量を評価する．2.2 節の方法でピクセル単位の推定結果を得た場合、 $P_{f^2}(u, v)$ 、 $\widetilde{\text{cor}}_{g_b, f}(u, v)$ は計算済みである．式 (29)~(31) から、 $P_{fV}(u, v) \sim P_{fD}(u, v)$ を生成するための $f(x, y)f(x+1, y)$ 、 $f(x, y)f(x, y+1)$ 、 $\{f(x, y)f(x+1, y+1) + f(x+1, y)f(x, y+1)\}$ の生成に、実数加算が $M'N'$ 回、実数乗算が $4M'N'$ 回必要である．次に、 $f(x, y)f(x+1, y)$ から再帰的な計算を用いて $P_{fV}(u, v)$ を計算するのに、1 フレームにつき、実数加算が $4M'N' +$

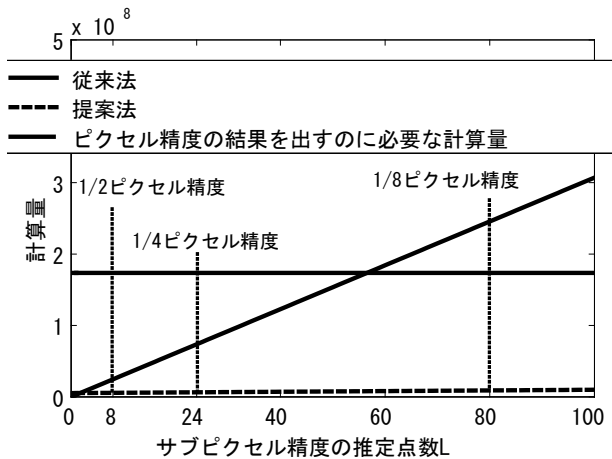


図 3: 計算量の比較 ($M' = 480, N' = 640, A = B = 16, M = N = 32$)

$(B - 3)M' + (A - 3)N'$ 回必要である。これは $P_{fH}(u, v)$, $P_{fD}(u, v)$ においても同様である。また、1 ブロック、一組の (u, v, α, β) において、式 (24), (27), (35) によって誤差尺度 $E_{b,f(\alpha,\beta)}(u, v)$ を計算するのに、実数加算が 14 回、実数乗算が 25 回必要である。これを 1 ブロックにつき L 組、さらに 1 フレームにつき D ブロック計算する必要がある。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、

$$P_{\text{sum}} = 14DL + 13M'N' + 3(B - 3)M' + 3(A - 3)N' \quad (39)$$

$$P_{\text{multi}} = 25DL + 4M'N' \quad (40)$$

となる。

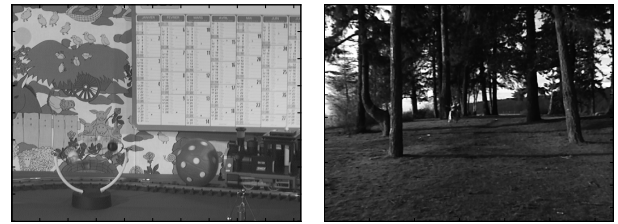
図 3 は $M' = 480, N' = 640, A = B = 16, M = N = 32$ とし、サブピクセル精度の推定点数 L を変化させた場合の計算量の比較である。ここで、計算量は実数加算回数と実数乗算回数の和としている。この図より、従来法では $L = 60$ 以上になると、ピクセル精度の推定を超える計算量が必要であることが分かる。従来法と提案法とを比較すると、 $L = 2$ 以上で提案法の方が計算量が少なく、 $L = 20$ 以上になると $1/10$ 以下に削減できることが分かる。

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、動き補償による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 4(a), 8 ビット濃淡, 400×512), forest (図 4(b), 8 ビット濃淡, 480×640) 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 と生



(a) 動画 caltrain (1 フレーム目) (b) 動画 forest (1 フレーム目)

図 4: 使用動画

成時間を測定した。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 2 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。画像を補間により高精度化し、SSD を計算する方法を従来法とし、提案法と処理時間を比較した。また、FFT の演算が高速な MATLAB を使用したため、 $P_{f2}(u, v) \sim P_{fD}(u, v)$ の計算には相互相関を用いた。シミュレーション環境を表 1 に示す。

表 1: シミュレーション環境

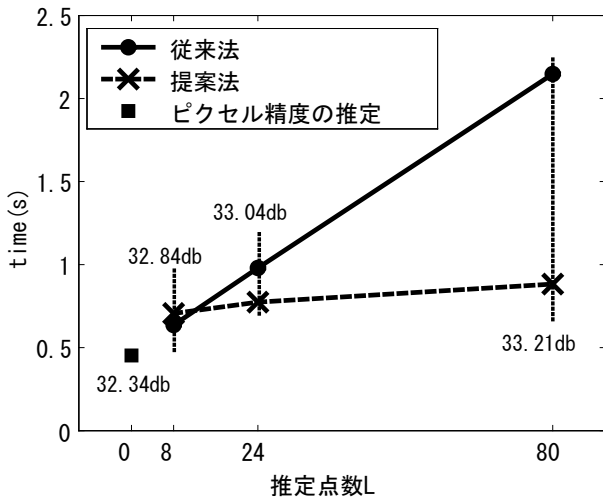
使用言語	MATLAB 6
CPU	intel Core2Duo 2GHz
メモリ	2.5GB

4.2 シミュレーション結果

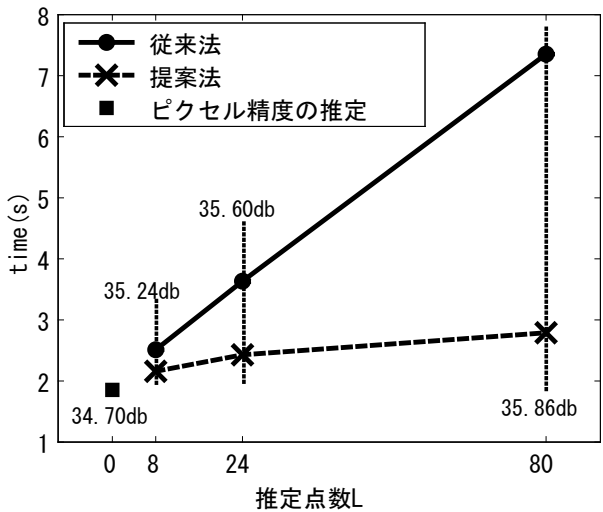
図 5(a), 5(b) は caltrain を用い、それぞれブロックサイズが $A = B = 16, 8$ の場合のサブピクセル精度の推定点数 L に対する処理時間の比較である。図中の数字は PSNR であり、提案法と従来法が同精度であることが分かる。これらの図より、提案法が従来法と比較して、高速で同精度の推定が可能であることが分かる。また推定点数が多いほど二つの差が大きく、提案法が有効であることが分かる。図 5(c) は forest を用いて実験した結果である。これにより、データに依存せず提案法が有効であることが分かる。

5 まとめ

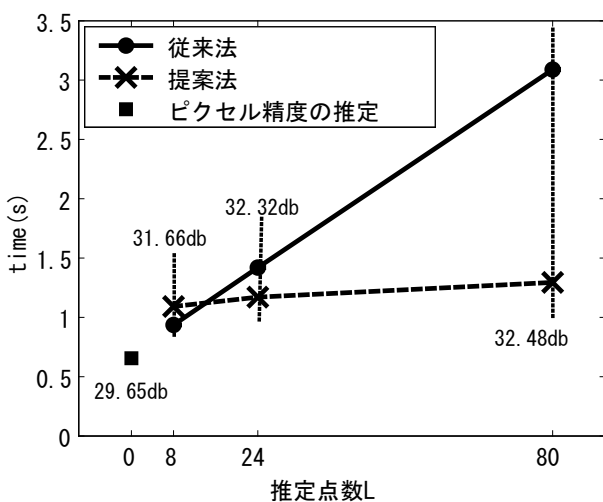
本稿では、二乗誤差和 SSD を誤差尺度とするサブピクセルブロックマッチングの高速アルゴリズムを提案した。補間により画素を生成し実際に SSD を計算する従来法と提案法とが同精度になることを解析的に証明し、計算量が少ないことを示した。シミュレーションにより、以上のことを確認した。



(a) 動画: caltrain, ブロックサイズ $A = B = 16$



(b) 動画: caltrain, ブロックサイズ $A = B = 8$



(c) 動画: forest, ブロックサイズ $A = B = 16$

図 5: 従来法と提案法の比較 (探索領域サイズ $M = N = 32$)

参考文献

- [1] T. Koga, K. Iinuma, A. Hirano and Y. Iijima, "Motion compensated interframe coding for video conferencing," in NTC 81 Conf. Record, IEEE pp. G5. 3. 1-G5. 3. 5, 1981.
- [2] S. Zhu and K. Ma, "A new diamond search algorithm for fast block motion estimation," IEEE Trans. Image Proc., 9(2), pp. 287-290, 2000.
- [3] Y. Noguchi, J. Furukawa and H. Kiya, "A fast full search block matching algorithm for MPEG-4 video," in Proc. IEEE Int. Conf. Image Proc., Vol. 1, pp. 61-65, 1999.
- [4] 李禎, 植村淳志, 貴家仁志, "FFT を用いた全探索 SSD ブロックマッチングの高速化," 映像学技報, vol. 33, no. 6, pp. 25-30, Feb. 2009.
- [5] Z. Li, A. Uemura and H. Kiya, "An FFT-based full-search block-matching algorithm with SSD criterion," in Proc. APSIPA, Oct. 2009, to be published.
- [6] W. Choi, J. Lee, S. Sungmo and B. Jeon, "Fast motion estimation and mode decision with variable motion block sizes," Visual Communications and Image Processing 2003, vol. 5150, pp. 1561-1572, June 2003.
- [7] 沼徳仁, 青木孝文, "映像のサブピクセル動き推定の実験的検討," 信学技報, SIP2007-128, vol. 107, no. 286, pp. 31-36, Oct. 2007.
- [8] 新井元基, 鷲見和彦, 松山隆司, "画像のブロックマッチングにおける相関関数とサブピクセル推定方式の最適化," 情報処理学会研究報告, 2004-CVIM-144(5), vol. 2004, no. 40, pp. 33-40, May 2004.
- [9] 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, Feb. 2006.
- [10] 佐川雅彦, 貴家仁志, "高速フーリエ変換とその応用," 昭晃堂, 1992.