

位相限定相関を用いた映像の位置ずれ補正の DSPによる実現

A DSP Implementation of Frame Displacement Correction Using Phase-Only Correlation

福田康宏, 阿部正英, 川又政征

Yasuhiro FUKUDA, Masahide ABE, Masayuki KAWAMATA

東北大学大学院工学研究科

Graduate School of Engineering, Tohoku University.

キーワード: 位相限定相関 (phase-only correlation), DSP (Digital Signal Processor), スケーリング (scaling)

連絡先: 〒980-8579 仙台市青葉区荒巻字青葉05 東北大学大学院 工学研究科 電子工学専攻 川又研究室
福田康宏, Tel.: (022)217-7095, Fax.: (022)263-9169, E-mail: fukuda@mk.ecei.tohoku.ac.jp

1. はじめに

映画はその誕生からすでに100年が経過している。そのため経年劣化により傷んだ映画フィルムも多く存在する。そのフィルムをデジタル画像処理により修復し、元の画像品質を復元する技術が必要とされている。

しかし、フィルムのデジタル修復にかかるコストは安いわけではない。このような状況下では商業的に成功した映画しかデジタル損傷修復されない。よって商業的利用価値が見込めないと考えられている映像でも、デジタル損傷修復が受けられるようにするためには、損傷修復コストを大幅に下げるための技術の開発が必要である^{1, 2)}。損傷修復コストの低減の方法としては、汎用の Digital Signal Processor (DSP) を用いて処理を行うことなどが考えられる。

本論文ではフィルムの位置ずれを補正する。フィ

ルムの損傷・劣化にはいくつかの種類があり、そのなかにフレームの不規則な揺れがある。これはフレーム内での大域的な位置ずれとして次のようにモデル化される。

$$I_n(n_1, n_2) = E_n(n_1 + \tau_{1n}, n_2 + \tau_{2n}) \quad (1)$$

ここで、 $E_n(n_1, n_2)$ と $I_n(n_1, n_2)$ は、原映像と位置ずれを含む映像のフレーム n の中の座標 (n_1, n_2) における輝度値を表す。フレーム n の (τ_{1n}, τ_{2n}) は位置ずれ量を表す。フレームごとの位置ずれ推定とは、与えられた $I_n(n_1, n_2)$ から映像中のそれぞれのフレームに対して (τ_{1n}, τ_{2n}) を推定することである。この位置ずれを推定する手法として、萩原らの位相限定相関を用いた手法がある^{3, 4, 5)}。

この位相限定相関を用いた手法を DSP を用いて実現することが本論文の目的である。ただし、今回は実際にフィルム映像の位置ずれ補正は行わず、ビデオカメラから取得した通常の映像を用い

て補正の動作を確認するにとどまった．また，萩原らの手法にあるサブピクセル精度の位置ずれ検出は行っておらず，位置ずれは1ピクセル単位で検出している．

本論文の構成は以下のとおりである．第1章では背景と目的を説明した．第2章では位相限定相関の特徴や算出の方法を述べる．実際に位置ずれ補正の処理を実装した環境や実装方法を第3章に示す．第4章で実装時に起こる演算の精度の問題とその解決策を考察し，第5章で実装結果を示す．最後に第6章でむすびとなる．

2. 位相限定相関の概要

本論文では位置ずれの検出に位相限定相関を用いた手法を採用する．位相限定相関とは，相関を計算する2つの画像の正規化クロスパワースペクトルのフーリエ逆変換として定義される．画像のスペクトルを振幅スペクトルで正規化しているため，求める位相限定相関がフレームごとの不規則な輝度値の変化に影響を受けにくく，より高精度でマッチングを行うことができる．以下では位相限定相関について説明する．

2.1 位相限定相関

本節では位相限定相関の算出方法を説明する．相関を計算する2つの画像 $a(n_1, n_2)$, $b(n_1, n_2)$ の位相限定相関 $g_{ab}(n_1, n_2)$ は次式のように表される．

$$g_{ab}(n_1, n_2) = \text{IDFT} \left[\frac{A(k_1, k_2) B^*(k_1, k_2)}{|A(k_1, k_2)| |B(k_1, k_2)|} \right] \quad (2)$$

ここで，IDFT は離散フーリエ逆変換を表す．また， $A(k_1, k_2)$ と $B(k_1, k_2)$ は，それぞれ $a(n_1, n_2)$ と $b(n_1, n_2)$ の離散フーリエ変換を表し，“*” は複素共役を表す．

位相限定相関は以下の順序で計算する．はじめに画像のスペクトル $A(k_1, k_2)$, $B(k_1, k_2)$ を求める．次にそのスペクトルを振幅スペクトルで正規化し，

$A_p(k_1, k_2)$, $B_p(k_1, k_2)$ を求める．

$$A_p(k_1, k_2) = \frac{A(k_1, k_2)}{|A(k_1, k_2)|} \quad (3)$$

$$B_p(k_1, k_2) = \frac{B(k_1, k_2)}{|B(k_1, k_2)|} \quad (4)$$

2つの正規化されたスペクトルのうち片方の複素共役を求め，それを用いて $G_{ab}(k_1, k_2)$ を求める．

$$G_{ab}(k_1, k_2) = A_p(k_1, k_2) B_p^*(k_1, k_2) \quad (5)$$

最後に合成位相を離散フーリエ逆変換することで位相限定相関 $g_{ab}(n_1, n_2)$ を求める．

$$g_{ab}(n_1, n_2) = \text{IDFT} [G_{ab}(k_1, k_2)] \quad (6)$$

2.2 位相限定相関の特徴

位相限定相関の特徴として，Fig. 1 のように自己相関関数がデルタ関数になるというものがある．位相限定相関ではどのような画像に対しても自己相関関数 $g_{aa}(n_1, n_2)$ がデルタ関数になることが大きな特徴の一つである．

$$g_{aa}(n_1, n_2) = \delta(n_1, n_2) \quad (7)$$

もう一つの大きな特徴として，相関を計算する画像にずれがあった場合そのずれ量だけ位相限定相関が移動することがあげられる^{3, 4, 5}．ある画像 $a(n_1, n_2)$ と，その画像を (τ_1, τ_2) だけずらした画像 $b(n_1, n_2)$ の位相限定相関 $g_{ab}(n_1, n_2)$ は以下のようになる．

$$b(n_1, n_2) = a(n_1 - \tau_1, n_2 - \tau_2) \quad (8)$$

$$g_{ab}(n_1, n_2) = g_{aa}(n_1 + \tau_1, n_2 + \tau_2) \quad (9)$$

$$= \delta(n_1 + \tau_1, n_2 + \tau_2) \quad (10)$$

相関を計算する画像にずれがある場合の例を Fig. 2 に示す．

以上の二つの特徴から，位相限定相関のピークの位置を求めることで，二つの画像のずれ量を求めることができる．

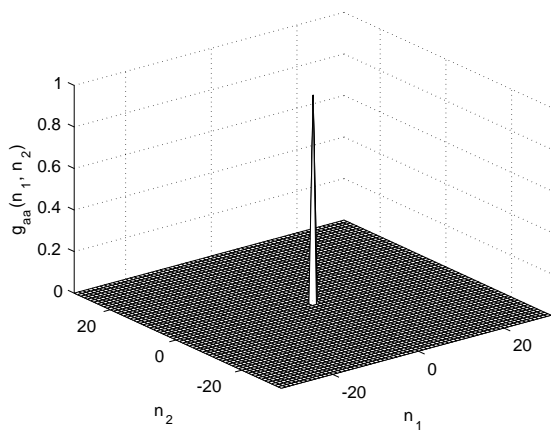


Fig. 1 位相限定相関の自己相関関数

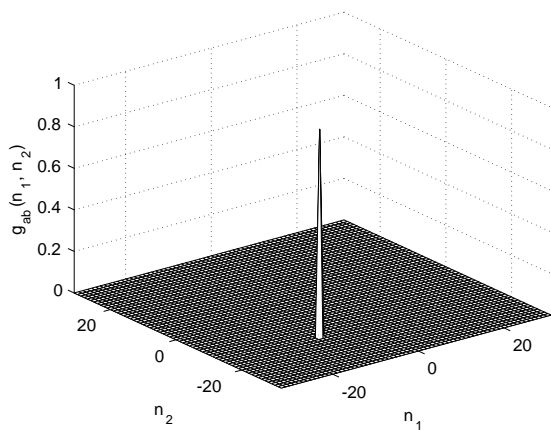


Fig. 2 画像にずれがある場合の位相限定相関

ただし、実際の位置ずれ検出においてこのようなデルタ関数がでることはほとんどない。実際に得られる位相限定相関はデルタ関数にノイズをのせたようなものになる。これは位相限定相関をとる画像がまったく同一であることがほとんどないためである。しかし、位相限定相関は鋭いピークを持つため、そのピークの検出が可能、すなわち、画像のずれの検出が可能である。

3. 実装方法

本章では使用した機器や実装環境を示す。また、基本的な処理の流れや処理方法もあわせて示す。

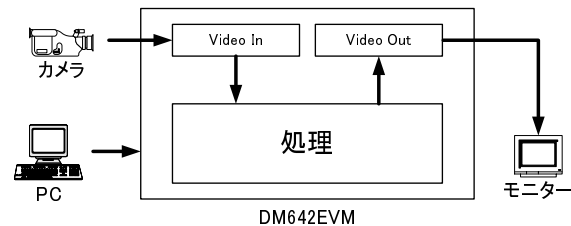


Fig. 3 実装環境

3.1 使用プラットフォーム

本論文ではアルゴリズムの実装に Texas Instruments (TI) 社の評価用モジュールDM642 Evaluation Module (EVM) を用いる。この EVM はビデオ・イメージング、およびネットワーク・アプリケーション向けの開発プラットフォームで、TI のデジタル・メディア・プロセッサ TMS320DM642 を搭載している。TMS320DM642 は動作周波数 600MHz のデジタルメディア向けの固定少数点 DSP である。また、DM642EVM は 64MB の SDRAM をそなえている。

3.2 実装環境

位置ずれ補正の処理を実装する環境は Fig. 3 である。

まずビデオカメラから画像を取得する。取得した画像は入力バッファに格納される。その画像を用いて位置ずれの検出および補正を行う。補正後の画像は出力バッファに格納され、モニターに表示される。

また、総合開発環境として TI 社の Code Composer Studio (CCS) を使用している。CCS を用いてプログラミングやそのコンパイル、データのロードなどを行い、その制御は PC で行っている。

プログラムは画像の出力のための関数など、一部アセンブリ言語を用いて記述された関数が用意されているが、自作した部分はすべて C 言語を用

いて記述している．

3.3 処理条件

本研究は古い映画フィルムのデジタル修復という研究課題が背景にあるため，処理対象は白黒画像とした．そのため，実際に処理を行うのは画像の輝度値成分のみである．

今回は 480×720 pixel の YUV4:2:2 形式の画像を入力として取得する．処理対象は輝度値であるので，Y 成分のみを処理する．その後，処理した画像を RGB に変換し VGA 出力する．このとき，変換係数を操作することで白黒画像を作り出すことにした．

また， 480×720 pixel の画像を取得し，その画像を補正した後出力するが，実際に位相限定相関を求めるのは画像の左上 128×128 pixel の部分だけとしている．位相限定相関を求める処理範囲は画像全体で行うのが望ましいが，FFT を行う際に処理をする画像のサイズが 2 のべき乗である方が都合がよいことと，処理にかかる計算時間が膨大になるのを防ぐことを考慮した結果，現段階ではこの処理範囲が適切であると判断した．前述の通り，位置ずれの補正および補正後の画像の出力は 480×720 pixel で行っている．

3.4 処理の流れ

画像の処理は Fig. 4 のように行う．はじめに，取得した画像の 2 次元 FFT (First Fourier Transform) を求める．今回 2 次元 FFT には行-列分解に基づく方法⁶⁾を用いた．行-列分解に基づく方法を用いて $x(n_1, n_2)$ を離散フーリエ変換すると次式のようになる．

$$\begin{aligned} X(k_1, k_2) &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} \\ &= \sum_{n_2=0}^{N_2-1} F(k_1, n_2) W_{N_2}^{k_2 n_2}, \end{aligned}$$

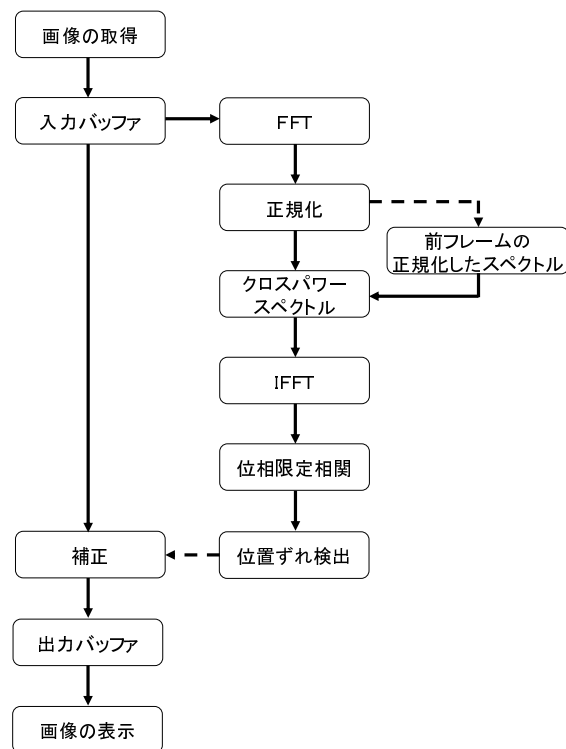


Fig. 4 画像の位置ずれ補正処理の流れ

$$k_1 = 0 \sim N_1 - 1, k_2 = 0 \sim N_2 - 1 \quad (11)$$

ただし

$$\begin{aligned} F(k_1, n_2) &= \sum_{n_1=0}^{N_1-1} x(n_1, n_2) W_{N_1}^{k_1 n_1}, \\ k_1 &= 0 \sim N_1 - 1, n_2 = 0 \sim N_2 - 1 \end{aligned} \quad (12)$$

とおいている． $F(k_1, n_2)$ の N_2 個の行は，それぞれ $x(n_1, n_2)$ の n_1 方向の N_1 点 1 次元離散フーリエ変換である．また， W_N は

$$W_N = \exp\left(-j \frac{2\pi}{N}\right) \quad (13)$$

で，回転因子である．1 次元 FFT には TI 社が提供する関数ライブラリ DSPLIB の FFT 関数を使用している．この FFT 関数は Mixed-Radix (混合基数) アルゴリズムを用いている．画像のスペクトルを求めた後，それを振幅スペクトルで正規化する．現在のフレームの画像の正規化したスペクトルと，1 フレーム前の画像の正規化したスペ

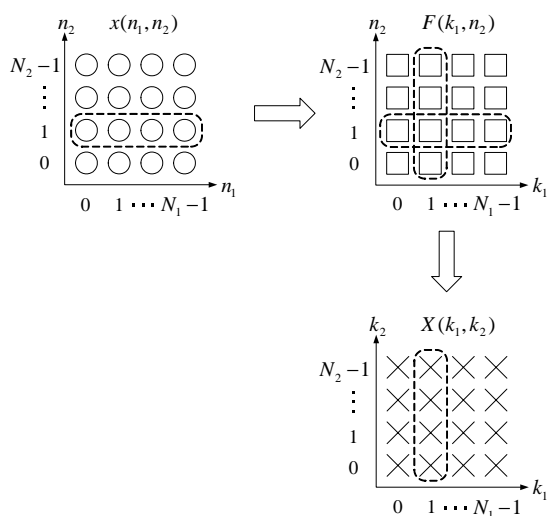


Fig. 5 行-列分解に基づく離散フーリエ変換

クトルを用いて、正規化クロスパワースペクトルを求める。このとき用いた現在のフレームの画像の正規化したスペクトルは、次フレームの処理時に使用するのを確保しておく。求めた正規化クロスパワースペクトルを IFFT (Inverse First Fourier Transform) して位相限定相関を求める。最後に位相限定相関のピークの位置を求め、位置ずれ量を検出し、それにもとづき位置ずれを補正する。

4. 計算精度の問題

上記の処理を行うにあたり、計算の精度の問題が起こる。

今回演算に用いたデータタイプは 16bit 固定少数点で小数点以下 14bit である。画像のデータは 8bit, 256 階調で、Fig. 7 に示す位置に格納する。この位置にデータを格納したのは、演算に用いる値の絶対値を 1 以下でかつできる限り大きい値にするためである。この状態で FFT を行うとスペクトルの値が大きくなり、オーバーフローが起きてしまう。そこでオーバーフローを防ぐため、FFT を行う前に処理する値をスケールし、値をあ

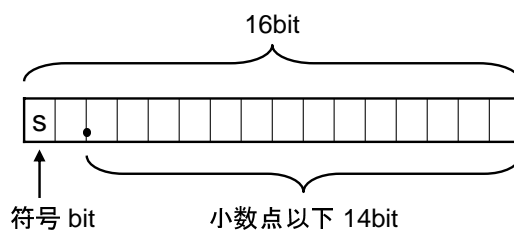


Fig. 6 演算に用いたデータタイプ

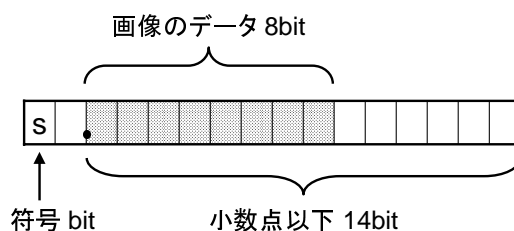


Fig. 7 画像のデータの格納位置

らかじめ小さくした。これによりオーバーフローが起きることはなくなるが、演算に用いる値が全体的に小さくなってしまう。

ここで問題が起こる。演算に用いるデータタイプは小数点以下 14bit であるので、絶対値が 2^{-14} より小さい値を表現することができない。しかし、位相限定相関を求める過程において絶対値が 2^{-14} より小さい値がでてきてしまう。これは画像のスペクトルが直流成分に集中し、その他の値が非常に小さくなってしまふことが原因である。このとき、絶対値が 2^{-14} より小さい値をすべてゼロにしているため、処理する値の大部分がゼロになってしまい、位相限定相関の正しい演算結果を求めることができない。

そこで、画像の 2 次元 FFT を行う部分からスペクトルを振幅スペクトルで正規化する部分までに、以下の二つの方法を用いてこの問題を解決した。

- 1) 直流成分の除去
- 2) スペクトルの拡大

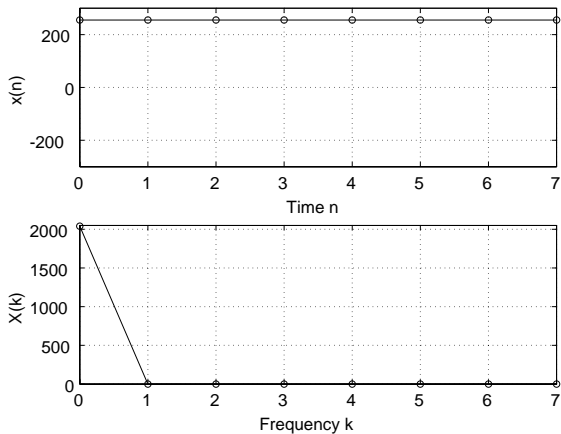


Fig. 8 振幅スペクトルが最大値を取る場合の例

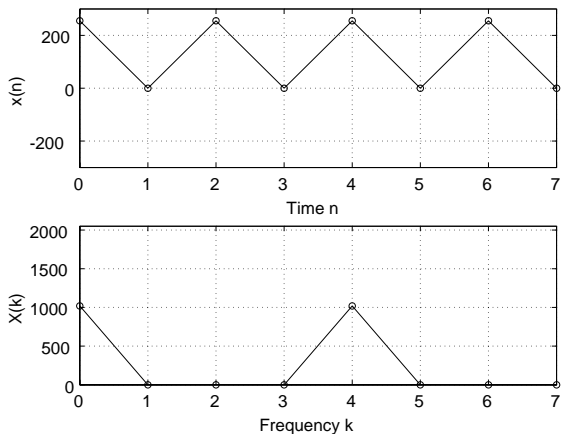


Fig. 9 直流成分以外での最大値の例

この二つの方法について順に説明する。

4.1 直流成分の除去

画像には直流成分が含まれており、その直流成分の除去を行う。直流成分を除去することで、行列分解に基づく FFT の行方向、列方向それぞれの計算において、演算の精度を上げることができる。以下ではその方法および効果を説明する。

4.1.1 行方向の FFT

直流成分の除去は、行-列分解による 2 次元 FFT のなかの行方向の 1 次元 FFT、つまり (12) 式を行う部分で行う。本節では例として 8 点の 1 次元

FFT を用いるが、実際に実装したのは 128 点の 1 次元 FFT である。また、画像の輝度値は 0 から 256 が 0 から 1 に対応するようにスケールを行っており、オーバーフローが起こらないようなスケールとは、演算に用いる値の絶対値が常に 1 以下となるようなスケールのことである。

今回取得する画像のデータは輝度値成分 256 階調である。すなわち画素の取り得る値の範囲が 0 から 255 である。この画像のデータの 1 次元の 8 点 FFT を行った場合、振幅スペクトルの取り得る値の最大値 X_{max} は 255×8 になる (Fig. 8)。よって、 X_{max} を取った場合でもオーバーフローが起こらないようなスケールでスケールを行うことにしている。つまり、この場合は $1/8$ にスケールリングすることになる。しかし、今回用いている 256 階調の画像において、 X_{max} を取り得るのは直流成分だけであり、直流成分以外の振幅スペクトルの取り得る値の最大値 \bar{X}_{max} はこの値よりも小さい。 \bar{X}_{max} は、今回用いている 256 階調の画像においては $(256 \times 8)/2$ である (Fig. 9)。つまり、画像の直流成分を除去すれば振幅スペクトルの取り得る値の最大値が半分になる。よって、スケールリングするスケールを $1/8$ から $1/4$ に変更し、演算の制度を 1bit 上げることが可能となる。

直流成分の除去方法は以下の通りである。画像の 256 階調の輝度値成分から 128 を差し引いた。直流成分を完全に除去するには画像から直流成分を計算しなければならない。よってこの方法では不十分であるが、直流成分を完全に除去する必要がないことと、計算の手間を省けることを理由に今回はこの方法を用いた。このことで画素の取り得る範囲が 0 から 255 だったのが -128 から 127 になり、直流成分の一部または全部が除去できる。直流成分の除去の例を Fig. 10, Fig. 11 に示す。このとき除去しきれなかった直流成分の取り得る値の最大値は 128×8 になる。この値は直流成分除去前の値の半分よりも大きい、この値を取るの

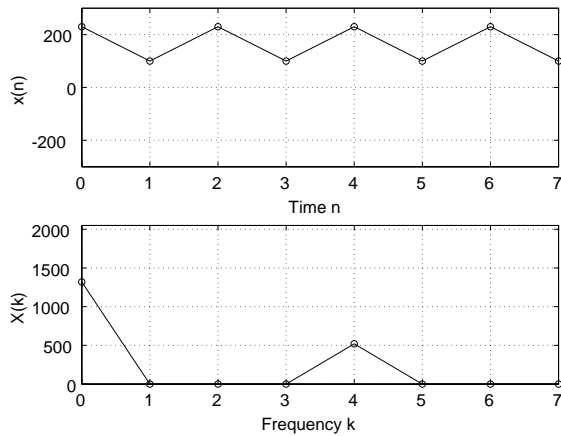


Fig. 10 直流成分の除去の例（除去前）

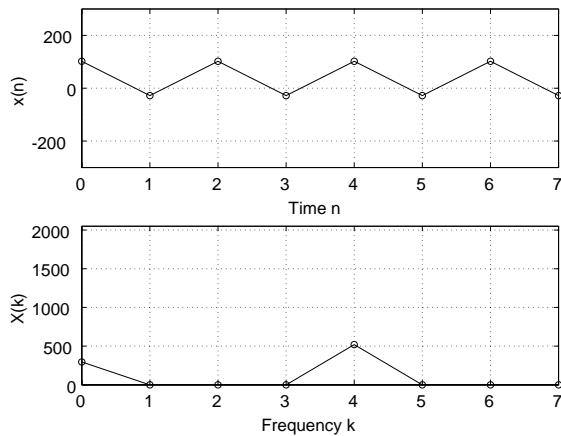


Fig. 11 直流成分の除去の例（除去後）

は画素のほとんどが 0 である画像の場合であるので考慮の対象から外す。また，振幅スペクトルで正規化した直流成分のスペクトルは常に 1 であることが分かっているので，この方法を用いた場合でも位相限定相関を求めるにあたり正しい演算結果を得ることができる。

4.1.2 列方向の FFT

本節では Fig. 12 の画像 lena を用いて考える。画像 lena のサイズは 128×128 pixel である。

行-列分解による 2 次元 FFT のなかの列方向，つまり (11) 式の 1 次元 FFT を行う場合，先に計算された行方向の 1 次元 FFT の計算結果 $F(k_1, n_2)$



Fig. 12 画像 lena

が入力になる。ここで，(11) 式の計算結果がオーバーフローしないように，入力 $F(k_1, n_2)$ をあらかじめ $1/128$ にスケールしておく。しかし，入力 $F(k_1, n_2)$ の列成分にはあらかじめ値が小さいものもあり，それらに関しては $1/128$ にスケールしなくてもオーバーフローすることはない。そこで，あらかじめ十分に値が小さいと判断できる部分はどこであるかを考える。

Fig. 13 に画像 lena の行方向の 1 次元 FFT の計算結果 $F(k_1, n_2)$ を示す。これを見て分かる通り，直流成分以外は非常に小さい。直流成分の値の平均が 17,206 であるのに対し，直流成分以外の値の最大値は 2,695 であった。よって直流成分以外はスケールのスケールを変更することが可能である。

しかし，今回は行方向の 1 次元 FFT (12) 式において，直流成分の除去を行っている。直流成分の除去を行った場合の FFT 結果は Fig. 14 のようになる。これで全体の値が十分に小さくなり，すべての帯域においてスケールのスケールを変更することが可能になる。具体的には $1/128$ にスケールするはずのところを， $1/64$ に変更した。よって演算の精度が 1bit 上がったことになる。

以上の方法は画像 lena のように，スペクトルが直流成分に集中する画像に限り有効である。本論文で処理対象としている自然画像も同様の特徴を

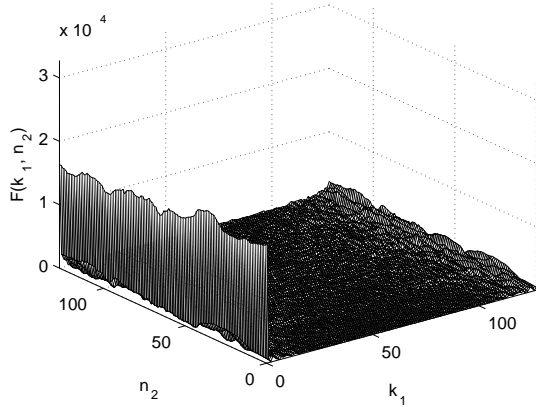


Fig. 13 画像 lena の 1 次元 FFT 結果

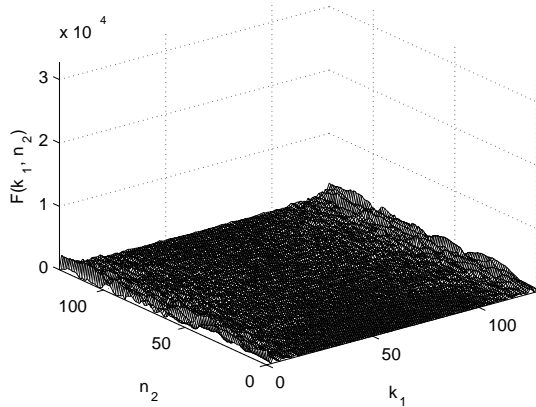


Fig. 14 画像 lena の 1 次元 FFT 結果 (直流成分除去)

持つものと考えているので、この方法を用いることができる。

4.2 スペクトルの拡大

位相限定相関を求める際、処理する値の絶対値が小さくなり、処理する値が表現できなくなってしまう問題が起こる主な処理はスペクトルの正規化を行う処理である。ここでは例としてスペクトル

$$X_n(k_1, k_2) = a_n(k_1, k_2) + jb_n(k_1, k_2) \quad (14)$$

を振幅スペクトル $|X_n(k_1, k_2)|$ で正規化する場合について考える。

前述のとおり、位相限定相関を求める過程においてスペクトルを振幅スペクトルで正規化する。ここで振幅スペクトルを求めるにあたり、複素数の絶対値を求める必要がある。そのとき次式のように実数部と虚数部それぞれの 2 乗をとることになる。

$$|X_n(k_1, k_2)| = \sqrt{a_n^2(k_1, k_2) + b_n^2(k_1, k_2)} \quad (15)$$

この $a_n^2(k_1, k_2)$ と $b_n^2(k_1, k_2)$ を求める部分が非常に小さい値どうしの乗算になり、乗算結果はさらに小さい値となってしまふ。このため、値が表現できなくなり、正しい演算結果がでない。

この問題を解決するために、スペクトルの値各々に対してスケーリングを行う。問題になっているのは、小さい値どうしを乗算することなので、値を拡大すればよい。拡大したスペクトルを次式に示す。

$$X_n'(k_1, k_2) = \alpha \{a_n(k_1, k_2) + jb_n(k_1, k_2)\} \quad (16)$$

ここで拡大率 α は

$$\alpha = \frac{1}{\max\{|a_n(k_1, k_2)|, |b_n(k_1, k_2)|\}} \quad (17)$$

である。

この方法はスペクトルの値一つ一つに対して異なる拡大率を用いてスペクトルを拡大する。この方法を用いると振幅スペクトルの値それぞれの比率が変化し、FFT の計算結果として正しいとは言えない値になる。しかし、この処理で変更を加えたのはスペクトルの振幅についてだけである。よって、位相限定相関を求めるにあたり振幅は正規化してしまうため、この処理をすることで位置ずれの検出に影響が出ることはない。このことは次式からも明らかである。

$$\frac{X_n(k_1, k_2)}{|X_n(k_1, k_2)|} = \frac{a_n(k_1, k_2) + jb_n(k_1, k_2)}{\sqrt{a_n^2(k_1, k_2) + b_n^2(k_1, k_2)}} \quad (18)$$

Table 1 平均 2 乗誤差

適用した処理	MSE
1) のみ適用	1.4752×10^{-17}
1) と 2) を適用	1.4754×10^{-17}

$$\frac{X_n'(k_1, k_2)}{|X_n'(k_1, k_2)|} = \frac{\alpha\{a_n(k_1, k_2) + jb_n(k_1, k_2)\}}{\alpha\sqrt{a_n^2(k_1, k_2) + b_n^2(k_1, k_2)}} \quad (19)$$

4.3 演算結果への影響

上記の方法を用いた場合の演算結果への影響を考察する。処理の流れは Fig. 15 のようになる。前述の通り、演算結果への影響はないはずである。そこで MATLAB を用いて確認を行った。MATLAB では浮動小数点型のデータタイプを用いて演算を行い、上記の方法を用いた場合と用いない場合では結果にどれだけの違いが現れるかを確認した。Fig. 16 は通常の方法で求めた位相限定相関である。Fig. 17 と Fig. 18 はそれぞれ第 4 章の 1) を適用して求めた位相限定相関、1) と 2) を適用して求めた位相限定相関である。また、Table 1 には上記の二つの位相限定相関と通常の方法で求めた位相限定相関の平均 2 乗誤差を示した。

図と表から分かる通り、通常の方法で求めた場合との誤差は非常に小さく、位置ずれの検出に影響がないことが確認できた。

5. 結果

以上の方法を用いてアルゴリズムを実装した。プログラムは C 言語で 513 行になった。このプログラムを CCS のコンパイラオプション -o3 で最適化し、ビルドした。これはファイルレベルでの最適化である。コンパイルの結果、ソースコードにおいて画像の FFT から補正までの処理の部分のプログラム容量は 40,440 [Byte] であった。これは、DSPLIB の FFT 関数のプログラム容量 23,304

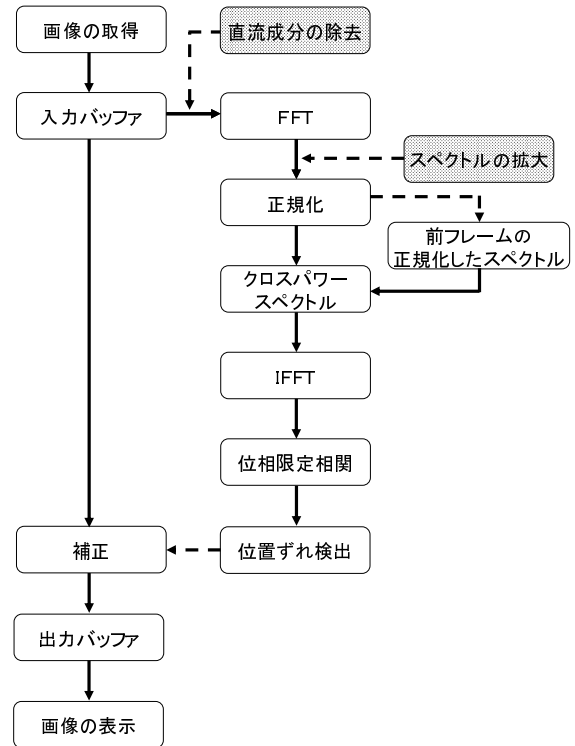


Fig. 15 演算精度対策追加後の流れ

[Byte] を含む。動作の確認は以下のように行った。

ビデオカメラからの映像には位置ずれは含まれていない。そこで取得した画像に人工的に位置ずれを発生させ、その画像の位置ずれ量を検出し、ずらした量と比較した。その結果、正しい検出ができていたことを確認できた。

次に、カメラをゆっくり動かすことで取得する画像を少しずつずらし、そのずれの補正を行った。この場合も正しく補正ができていたことを視覚的に確認できた。

計算時間の測定は TI 社の TMS320C6000 スタンドアロン・シュミレータを使用して行った。測定の結果から、演算精度対策の処理も含め画像の FFT から補正までの処理に 18 [ms] かかっていることが分かった。画像の入出力や、画像のフォーマットの変換など、映像の素通しの処理部分には、入出力デバイスの情報の取得が必要な関数が使われているため、スタンドアロン・シュミレータを

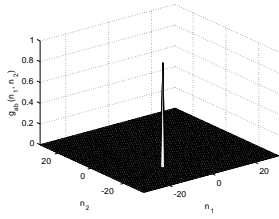


Fig. 16 通常の方法

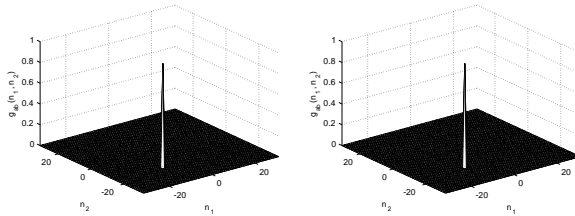


Fig. 17 1) の方法 Fig. 18 1) と 2) の方法

用いた計算時間の測定が行えない。よって、処理を 30 [frame/s] で行うには 1 frame を 33 [ms] 以内に処理できればよいが、この処理速度を達成できているかどうかは確認できていない。しかし、映像の素通しでは 30 [frame/s] で画像が更新されていることが分かっているので、素通しの処理、具体的には画像の FFT から補正までの処理以外の処理の処理時間は 33 [ms] 以内である。これに測定した処理時間 18 [ms] を加えると、位置ずれ補正にかかる処理時間は 51 [ms] 以内となる。よって、出力画像の更新が 33 [ms] の間隔で行われることから、15 [frame/s] 以上の速さで処理が行えているものと考えている。

6. おわりに

位相限定相関を用いた映像の位置ずれ補正を DSP 上で実現した。しかし、処理の計算時間が多くかかってしまうため、目標とする 30 [frame/s] の処理速度は達成できていない。また、処理範囲も 128×128 pixel であるので、取得する 480×720 pixel の画像のほんの一部でしかない。よって、移動物体などの映像の部分的な動きに位置ずれ検出が影響を受けやすい。この影響を少なくするには

処理範囲を画像全体にするのが望ましい。当然処理範囲が広がるのでより多くの計算時間がかかる。以上の点の解決法の一つとして処理の高速化がある。よって、アルゴリズムの改良や命令の並列化による処理の高速化が今後の課題である。

参考文献

- 1) 齊藤 隆弘: デジタル画像処理によるフィルム映像の復元 (1), 映像情報メディア学会誌, 55-12, 1599/1604 (2001)
- 2) 齊藤 隆弘: デジタル画像処理によるフィルム映像の復元 (2), 映像情報メディア学会誌, 56-1, 85/92 (2002)
- 3) M. Hagiwara, M. Abe and M. Kawamata: Estimation method of frame displacement for old films using phase-only correlation, Journal of Signal Processing (採録決定), (2004)
- 4) M. Hagiwara and M. Kawamata: Detection of frame displacement for old films using phase-only correlation, ISPACS (2002)
- 5) 萩原 瑞木, 阿部 正英, 川又 政征: カメラワークを考慮した古いフィルム映像の位置ずれ補正, 電子情報通信学会論文誌 (投稿中)
- 6) 川又 政征, 樋口 龍雄: 多次元デジタル信号処理, 61/63, 朝倉書店(1995)