

重みの対称性と空間分解による適応的バイラテラルフィルタの高速化の検討 Acceleration of Adaptive Bilateral Filter base on Spatial Decomposition and Symmetry of Weights

真喜志 泰希[†]
Taiki Makishi

山田 親稔[‡]
Chikatoshi Yamada

市川 周一[§]
Shuichi Ichikawa

1. はじめに

画像の輪郭を保存したノイズ除去フィルタとしてガウシアンフィルタ (Gaussian Filter: 以下 GF) が従来用いられているが、近年、GF よりも有用なフィルタとしてバイラテラルフィルタ (Bilateral Filter: 以下 BF) が CG などの広い範囲で利用されてきている [1]。BF は、GF と同様にガウス関数に従った重み付けを行うことから高いノイズ除去性能を有しており、核医学画像の分野においては、汎用性の高い短時間収集を可能とする [2] ことから、核医学画像のノイズ除去フィルタとして応用が期待されている。その他にも、特徴点抽出における前処理としても有効性が示されており、核医学画像に限らず、医療用動画画像である MRI (Magnetic Resonance Imaging system) や CT (Computed Tomography)、超音波診断、内視鏡検査においても、BF によるエッジ強調が有効であると言える。しかし、特徴点抽出の前処理として BF を用いた場合、画像の主要なエッジ付近でエッジの勾配が反転し不自然なノイズが生じる事が指摘されている。その不自然なノイズが残っている状態でコントラスト強調などを行う場合、不自然な状態でエッジが強調されピーク信号対雑音比 (以下 PSNR) が低下することも指摘されている。このノイズの原因は BF のエッジ強調性にあり、このエッジ協調性を抑えることができれば、不自然なノイズの生じないフィルタ処理が可能となる。このエッジ強調性を抑える手法として、エッジ強調性を抑制する適応的バイラテラルフィルタ (Adaptive Bilateral Filter: 以下 ABF) が提案されている。この ABF は、出力の判定にエッジ保存性のない GF を用い、GF と BF の入力信号値から出力信号値への変化により、出力を適応的に判定することにより、エッジ強調性を抑制したフィルタ処理が可能となる BF となっている [3]。しかし、前処理として利用する場合には、処理時間は高速であることが望ましいと言えるが、ABF は BF に比べ計算量が多く、処理に長い時間を要するという欠点がある。

そこで本研究では、ABF の式を加重係数のテイラー展開に基づく空間分解 [4] による高速化と重みの対称性 [5] による高速化の2つの高速化手法を ABF へ適用し高速化を図り、高速化手法の適用前後での ABF の処理時間の比較を行い高速化されている事を確認した。そして、高速化手法を適用した ABF を GPU へ実装し、CPU で実行した場合と GPU (Graphics Processing Unit) で実行した場合の処理時間の比較を行った。

2. 原理

2.1. バイラテラルフィルタ

BF は、空間座標と輝度に対応し、ガウス分布に従った重み付けを行うノイズ除フィルタである。フィルタの重みの係数が2つ (bi) の側面 (lateral) によって決定される。入力画素を $f(i, j)$ 、出力画素を $b(i, j)$ とすると BF の出力は、式 (1) で表される。

$$b(i, j) = \frac{\sum_{x=-\omega}^{\omega} \sum_{y=-\omega}^{\omega} f(i+x, j+y) W(i, j : x, y)}{\sum_{x=-\omega}^{\omega} \sum_{y=-\omega}^{\omega} W(i, j : x, y)} \quad (1)$$

$$W(i, j : x, y) = e^{-\alpha \|x^2 + y^2\|} e^{-\beta \|f(i, j)^2 - f(i+x, j+y)^2\|}$$

式 (1) は画像を平坦化する項 (対空間) と輪郭の平坦化を抑制する項 (対輝度) により構成される。これにより、輝度差の大きい画素は重みが抑制され、輪郭を保存したノイズ除去が可能となる。また、空間方向による重み付けの項 $\exp(-\alpha(x^2 + y^2))$ のみに着目すると、それが GF であることが分かる。すなわち、BF は輝度を加味した GF であると考えることが可能である。ガウス関数は、フーリエ変換もガウス関数になることや、スケールスペースで因果律を満たす唯一の核関数であることなど優れた性質が多く、これによる重み付けを行う事で高いノイズ除去能力を発揮する。BF は以下のような特徴を有する [1]。

- (1) 非線形拡散や平均シフトフィルタのような反復計算を要せず、動作が直感的に把握しやすい。
- (2) 簡単な単一の式だけで表され、実装しやすい。
- (3) パラメータが α と β の2個だけであるため設定しやすい。
- (4) ガウス性白色ノイズの除去能力とエッジの保存能力の両方を兼ね備えている。
- (5) 線形フィルタと同様に入力画素値の加重平均が出力画素値となる。

しかし、GF に比べ輝度に対応する重みの計算が加味されている為、GF よりも長い処理時間を要するといった欠点がある。

[†] 沖縄工業高等専門学校 専攻科 創造システム工学コース

[‡] 沖縄工業高等専門学校 情報通信システム工学科

[§] 豊橋技術科学大学 電気・電子情報工学系

2.2. 適応的バイラテラルフィルタ

ABFは、BFのエッジ強調性を抑制するために、各入力画素におけるABFの出力 $\bar{b}(i, j)$ はエッジを強調する場合には、入力信号 $f(i, j)$ をそのまま出力し、平滑化するにはBFの出力信号 $b(i, j)$ を出力するBFとなっている。このエッジ強調と平滑化の判定にエッジの保存を必要としないGFを判定に使用する。これにより、エッジは強調されずに保存されるため、不自然なノイズの生じないフィルタ処理が可能となる。判定式に使用するGFの出力 $g(i, j)$ は式(2)で表される。

$$g(i, j) = \frac{\sum_{x=-\omega}^{\omega} \sum_{y=-\omega}^{\omega} f(i+x, j+y) e^{-\hat{\alpha} \|x^2+y^2\|}}{\sum_{x=-\omega}^{\omega} \sum_{y=-\omega}^{\omega} e^{-\hat{\alpha} \|x^2+y^2\|}} \quad (2)$$

ここで、GFの標準偏差である $\hat{\alpha}$ は、

$$\sum_{x=-q}^q \sum_{y=-p}^p (W(i, j; x, y) - e^{-\hat{\alpha} \|x^2+y^2\|}) = 0 \quad (3)$$

の解である。すなわち、各入力画素におけるBFの重みの総和とGFの重みの総和が等しくなるように $\hat{\alpha}$ を調整する。

ABFの出力 $\bar{b}(i, j)$ の式は、式(4)となる。

$$\bar{b}(i, j) = \begin{cases} f(i, j) & [b(i, j) - f(i, j)][g(i, j) - f(i, j)] < 0 \\ b(i, j) & \text{otherwise} \end{cases} \quad (4)$$

式(4)は入力信号値からBFの出力信号値への変化と入力信号値からGFの出力値への変化の向きが一致している場合にBFの出力 $b(i, j)$ を出力し、不一致の場合には、入力画素 $f(i, j)$ をそのまま出力する式となっている[3]。

512*512[pixel]の画像に標準偏差5のガウス性ノイズを付加し、ABFとBFを適用した。入力画像とノイズを付加した画像を図1と図2に、BFとABFの出力結果を図3と図4に示す。



図1: 入力画像



図2: ノイズ付加画像



図3: BF

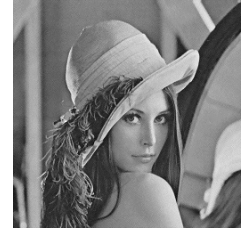


図4: ABF

図1～図4の130432～130688[pixel]までの輝度値のグラフを図5～図8に示す。

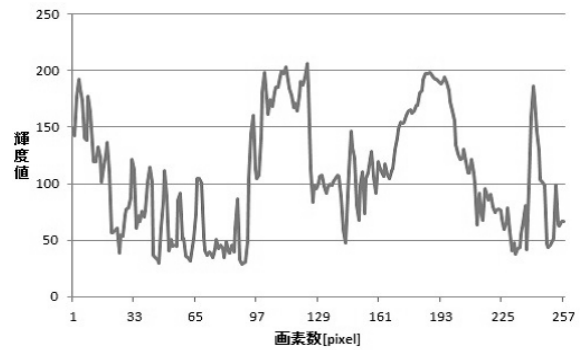


図5: 入力画像

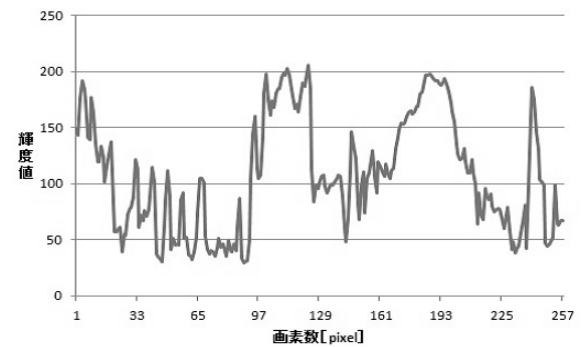


図6: ノイズ付加画像

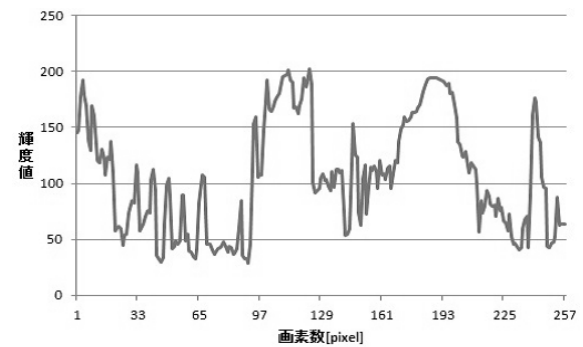


図7: BF

図7と図8の130516[pixel]付近に着目し図5と比較すると、BFのほうは輝度勾配が減少し、ABFのほうはエッジ勾配が保持されていることがわかる。

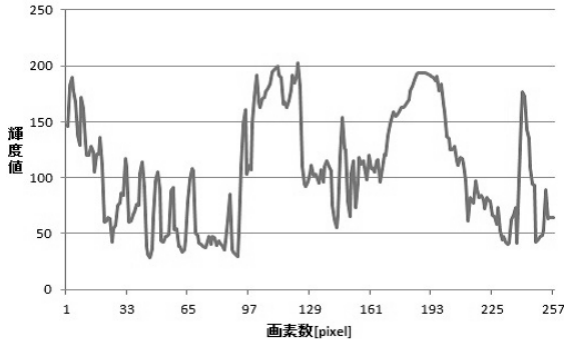


図 8: ABF

さらに、BFにより前処理を行った場合に発生する不自然なノイズを確認するために、図3に単純強調であるアンシャープマスキング (Unsharp Masking; 以下UM) を施した結果を図9に、130432~130688[pixel]までの輝度値のグラフを図10に示す。



図 9: BF に UM を適用した図

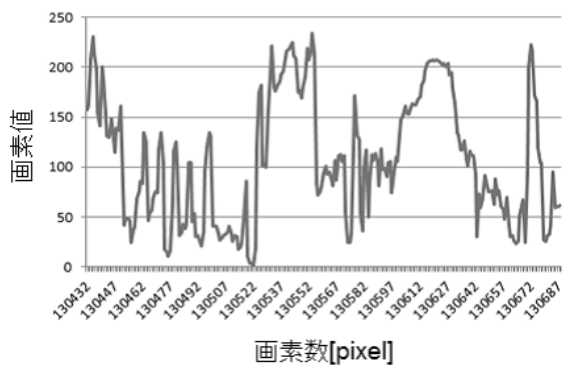


図 10: BF

130516[pixel]付近での輝度勾配がそのまま増長されたため、入力画素とは異なった出力結果となっている。これが、BFによる不自然ノイズである。

3. 高速化手法

3.1. 空間分解

式(1)の空間方向の重みの項である $\omega_\alpha(x, y) = \exp(-\alpha\|x^2 + y^2\|)$ は、指数法則により分解すると、

$$\omega_\alpha(x, y) = e^{-\alpha x^2} e^{-\beta y^2} \quad (5)$$

となる。輝度の重みの項である $\omega_\beta(f(i, j), f(i+x, j+y)) = \exp(-\beta\|f(i, j) + f(i+x, j+y)\|^2)$ は、空間方向の重みの項とは異なり、厳密には空間分解を行うことができないため、テイラー展開を用いてこの項を

$$e^{-\beta\|f(i, j) - f(i+x, j+y)\|^2} = e^{-\beta(D^2_{i,j}(x) - D^2_{i+x,y}(y) - 2D_{i,j}(x)D_{i+x,y}(y))} \quad (6)$$

$$D_{i,j}(x) = f(i, j) - f(i+x, j)$$

$$D_{i+x,j}(y) = f(i+x, y) - f(i+x, j+y)$$

と変形する。ここで、分解ができない原因である項は $\exp(-2\beta D_{i,j}(x)D_{i+x,y}(y))$ であるため、この項をテイラー展開をすると、

$$e^{-\beta\|f(i, j) - f(i+x, j+y)\|^2} = e^{-\beta(D^2_{i,j}(x) - D^2_{i+x,y}(y))} - 2\beta D_{i,j}(x) e^{-\beta D^2_{i,j}(x)} D_{i+x,y}(y) e^{-\beta D^2_{i+x,y}(y)} + 2\beta^2 D^2_{i,j}(x) e^{-\beta D^2_{i,j}(x)} D^2_{i+x,y}(y) e^{-\beta D^2_{i+x,y}(y)} + \dots \quad (7)$$

となり、式(8)の右辺の項はそれぞれ(高次の項も)すべて x の関数と y の関数の積に分解することが可能となる。ここで式(1)を

$$b_{i,j} = \frac{s_{i,j}}{t_{i,j}} \quad (8)$$

$$s_{i,j} = \sum_{x=-\omega}^{\omega} \sum_{y=-\omega}^{\omega} f(i+x, j+y) \times W(i, j; x, y) \quad (9)$$

$$t_{i,j} = \sum_{x=-\omega}^{\omega} \sum_{y=-\omega}^{\omega} W(i, j; x, y) \quad (10)$$

と置き、空間分解を行った各項を代入すると式(9)、式(10)は、式(11)、式(12)となり、式(8)の分子と分母それぞれについて、 y 方向の一次畳込みの結果を中間画像として、その中間画像に x 方向の一次畳込みを得られる。

$$s_{i,j} = \sum_{x=-q}^q e^{-\alpha x^2 - \beta(f(i,j) - f(i+x,j))^2} \quad (11)$$

$$\begin{aligned} &\times (g_{i+x,j} - 2\beta D_{i,j}(x)G_{i+x,j}) \\ t_{i,j} = &\sum_{x=-q}^q e^{-\alpha x^2 - \beta(f(i,j) - f(i+x,j))^2} \quad (12) \\ &\times (h_{i+x,j} - 2\beta D_{i,j}(x)H_{i+x,j}) \end{aligned}$$

ここで $g_{i,j}$, $G_{i,j}$, $h_{i,j}$, $H_{i,j}$ はそれぞれ, 式 (13) ~ 式 (16) となる.

$$g_{i,j} = \sum_{y=-p}^p e^{-\alpha y^2 - \beta(f(i,j) - f(i,j+y))^2} \quad (13)$$

$$\times f(i,j+y)$$

$$G_{i,j} = \sum_{y=-p}^p e^{-\alpha y^2 - \beta(f(i,j) - f(i,j+y))^2} \quad (14)$$

$$\times (f(i,j) - f(i,j+y))f(i,j+y)$$

$$h_{i,j} = \sum_{y=-p}^p e^{-\alpha y^2 - \beta(f(i,j) - f(i,j+y))^2} \quad (15)$$

$$H_{i,j} = \sum_{y=-p}^p e^{-\alpha y^2 - \beta(f(i,j) - f(i,j+y))^2} \quad (16)$$

$$\times (f(i,j) - f(i,j+y))$$

ABF の出力の判定に使用する GF についても, BF の空間方向の項と同様の分解を行う. 空間分解を行った GF は式 (17) となる.

$$g_{i,j} = \frac{S_{i,j}}{T_{i,j}} \quad (17)$$

$$S_{i,j} = \sum_{x=-\omega}^{\omega} e^{-\alpha x^2} P_{i+x,j} \quad (18)$$

$$T_{i,j} = \sum_{x=-\omega}^{\omega} e^{-\alpha x^2} Q_{i+x,j} \quad (19)$$

ここで $P_{i,j}$, $Q_{i,j}$ はそれぞれ, 式 (20), 式 (21) となる.

$$P_{i,j} = \sum_{y=-\omega}^{\omega} e^{-\alpha y^2} f(i,j+y) \quad (20)$$

$$Q_{i,j} = \sum_{y=-\omega}^{\omega} e^{-\alpha y^2} \quad (21)$$

以上から分解による手順は,

Step 1) 全ての入力画素について, 中間画像である $g_{i,j}$, $G_{i,j}$, $h_{i,j}$, $H_{i,j}$, $P_{i,j}$, $Q_{i,j}$ を計算する.

Step 2) Step 1 より得られた中間画像より全ての入力画素について, $s_{i,j}$, $t_{i,j}$, $S_{i,j}$, $T_{i,j}$ を計算し出力する.

となる [4].

3.2. 重みの対称性

式 (1) は, マスク半径を N とした場合の演算は, 注目画素 $f(i,j)$ と近傍データ $f(i-N,j-N) \sim f(i+N,j+N)$ との畳み込み演算を行う. その際に近傍データとの畳み込みの演算結果を保持しておく. $f(i,j)$ の畳み込み演算が終了し, 注目画素が $f(i+x,j+y)$ へ移る. この場合は通常どおりの演算を行うと, 近傍データ $f(i+1-N,j+1-N) \sim f(i+1+N,j+1+N)$ との畳み込みの演算を行うことになるが, ここで注目画素が $f(i,j)$ の時の演算結果を再利用することにより, $f(i+1-N,j+1-N) \sim f(i,j)$ の演算を省略することが可能となる [5].

4. 結果・考察

本実験で使用したソフトウェア開発環境及び動作環境を表 1 に示す.

表 1: ソフトウェア開発および動作環境

開発言語	C/C++
開発環境	Visual Studio 2010
使用ライブラリ	OpenCV 2.4.3/CUDA 5.0
OS	Windows 7 64bit
CPU	Core i7 2640M 2.8GHz
GPU	Geforce GTX295
メモリ (RAM)	DDR3 8GB

重みの対称性とテイラー展開を用いた空間分解の 2 つの高速化手法を適用した ABF の α , β をそれぞれ, 各パラメータを $\alpha = 0.0002$, $\beta = 0.00005$, マスク半径 2 として, $16*16 \sim 512*512$ [pixel] の lena の画像に ABF を適用し, 高速化手法の適用前後での, 処理時間の比較を行った CPU 処理にて行った結果を図 11 に示す.

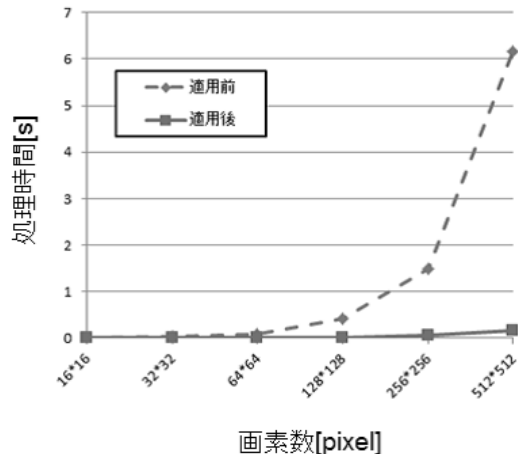


図 11: 高速化手法適用前後での比較

高速化手法の適用前後において、 $16*16\sim 64*64$ までの低解像度領域においては、CPU と GPU での処理時間に差は見られないが、 $512*512$ [pixel] での処理時間に着目すると、適用前が 6.15 [s] で、適用後は 0.48 [s] となった。適用後の時間は適用前と比べ約 $1/12$ 程度まで短縮された。しかし、信号対雑音比 (PSNR) が適用前と比べ低下が見られたが適用前後どちらも PSNR は、 30 dB 以上を保持していることを確認した。これは、テイラー展開による空間分解の際の近似誤差であると考えられる。

続いて、 $16*16\sim 4096*4096$ [pixel] の画像へ高速化を施した ABF を適用した。CPU と GPU 上での処理時間の比較結果を図 12 に示す。

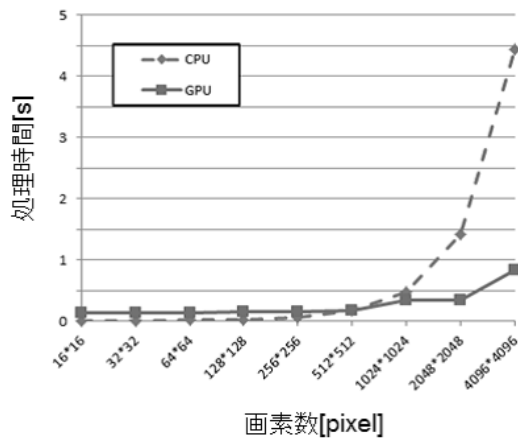


図 12: CPU と GPU での比較

$4096*4096$ [pixel] の場合、GPU での処理時間は 4.44 [s]、GPU での処理時間は 0.84 [s] となり、GPU での処理時間は CPU での処理時間に比べ約 $1/5$ まで短縮された。しかし、 $16*16\sim 512*512$ [pixel] の画像における処理時間は、GPU での処理よりも CPU での処理のほうが早い結果となっている。これは、重みの対称性による高速化手法を実装するにあたり、GPU 上でメモリを保持することができないため、マスク内の各近傍データとの演算結果をを一旦 CPU へ渡し、CPU 側のメモリ上で保持するという手法で実装したため、一ピクセルの処理中に近傍データの数だけ GPU と CPU との間での演算結果の受け渡しにより時間増大したものだと思われる。

実際の医療用画像である股関節を撮影した CT 画像を図 13、この CT 画像を入力画像としてノイズを付加した画像を図 14、ノイズの付加された画像に BF を適用した結果を図 15、高速化手法を施した ABF をノイズが付加された画像に適用した結果を図 16 に示す。

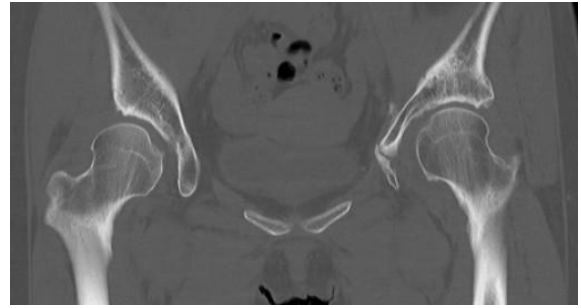


図 13: CT 画像



図 14: ノイズ付加画像

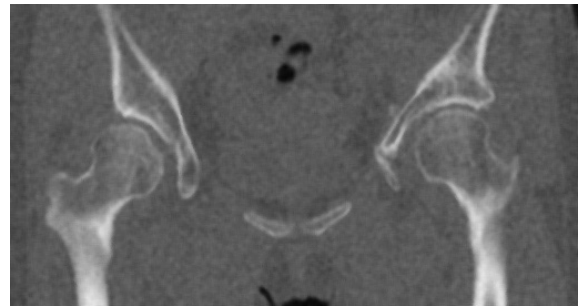


図 15: ノイズ付加画像に BF を適用した画像



図 16: ノイズ付加画像に ABF を適用した画像

図 14 に対し、図 15 と図 16 は、付加されていた白色性ノイズが除去されていることがわかる。しかし、図 15 と図 16 を比較すると図 15 のほうは、エッジが強調されているためぼやけているが、図 16 は、ぼかしが抑えられた結果となっており、この結果からエッジ強調性が抑制されているがわかる。また、図 15 と図 16 の

それぞれの入力信号値に対する出力信号値のグラフを図 17 と図 18 に示す。

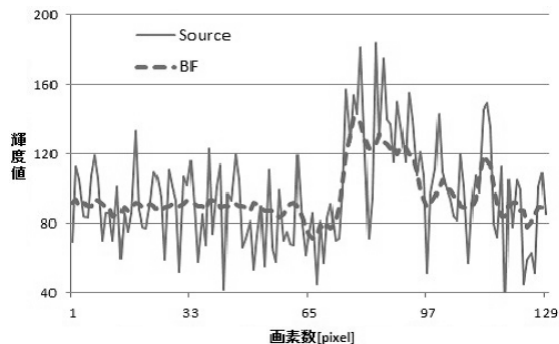


図 17: BF の入力信号値に対する出力値

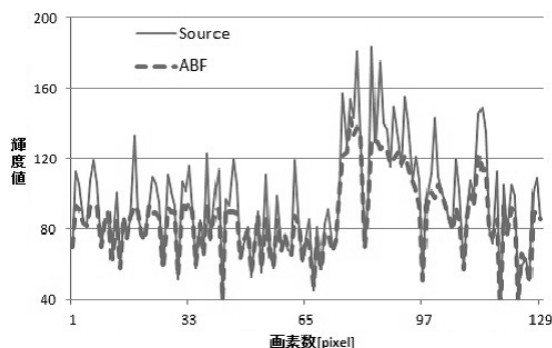


図 18: ABF の入力信号値に対する出力値

図 17 と図 18 を比較すると、図 17 は、付加された白色ノイズが除去されエッジが強調されている。図 18 は、図 17 と同様に白色ノイズが除去されているが、エッジの強調は抑制されていることがわかる。

5. まとめ

画像の輪郭を保存したノイズ除去フィルタとして GF が従来用いられているが、近年、GF よりも有用なフィルタとして BF が CG などの広い範囲で利用されてきている。BF は、GF と同様にガウス関数に従った重み付けを行うことから高いノイズ除去性能を有しており、核医学画像の分野においては、汎用性の高い短時間収集を可能とする [2] ことから、核医学画像のノイズ除去フィルタとして応用が期待されている。しかし、近年、BF によるノイズ除去を行った場合、画像の主要なエッジ付近でエッジの勾配が反転してしまい不自然なノイズが生じてしまう事が指摘されている。この欠点を解決するフィルタとして研究されているエッジ強調性を抑制する ABF は前処理として利用する場合、処理が高速であることが望ましいが、ABF には BF に比べて長い処理時間を要するという欠点がある。

そこで、本稿ではこの欠点を解決するため ABF の高速化を目的とし、ABF に加重係数のテイラー展開による空間分解による高速化と重みの対称性による高速化の 2 つの高速化手法を適用することにより高速化を図った。高速化されていることを確認するために高速化手法

の適用前後での比較を行い、処理時間が適用前と比べて約 1/12 程度まで短縮され、PSNR も 30dB 以上を保持していることを確認した。また、高速化されていることを確認した。高速化手法適用後の ABF を GPU へも実装し、CPU での処理時間との比較を行い、高解像度画像において処理時間が約 1/5 程度まで短縮されていることを確認した。しかし、低解像度領域において CPU に比べて、GPU のほうが遅いという結果となった。これは、重みの対称性による高速化を実装するにあたり、前のピクセルでの処理結果をメモリへ保持するために、1 ピクセルの計算が終了する毎に、CPU へデータを保存していた事が原因であると考えられる。今後の課題としては、重みの対称性のアルゴリズムの GPGPU へ最適な実装方法を検討することが課題となった。

今後の展開として、FPGA などの専用ハードウェアに実装し、比較検討を行う事が考えられる。

謝辞

本研究は JSPS 科研費 25871048 の助成を受けたものである。本研究の遂行にあたり、股関節 CT 画像をご提供頂いた琉球大学医学部山口浩医師、金谷文則医師、ならびに多くのご助言を頂いた琉球大学工学部長田康敬教授に深く感謝する。

参考文献

- [1] 浦浜 喜一, “第 8 回画像の本質を引き出す: バイラテラルフィルタによるノイズ除去とイラスト風画像の生成,” 映像情報メディア学会誌, 62(8), pp.1268-1273, 2008
- [2] 片山 豊, 上田 健太郎ほか, “核医学画像へのエッジ保存平滑化フィルタの適用”, 第 54 回日本放射線技術学会 近畿部会, p.28, 2010
- [3] 井上 光平, 小川 真人, 原 健二, 浦浜 喜一, “エッジ協調性を抑制する適応的バイラテラルフィルタ”, 信学論, Vol J93-D, No. 1, pp.68-70, 2010
- [4] 景 琳琳, 浦浜 喜一, “加重係数のテイラー展開に基づく空間分解バイラテラルフィルタの高速化”, 信学論, Vol J89-A, No. 7, pp.643-645, 2006
- [5] 浦浜 喜一, 井上 光平, “重みの対称性に基づくバイラテラルフィルタの高速化”, 信学論, Vol J93-D, No. 2 p.148-150, 2010