

## 値域フーリエ級数展開による冗長ウェーブレット変換 Redundant Wavelet Transform by Range-Fourier Series Expansion

林 晃平<sup>1)</sup> 本田 宗一郎<sup>1)</sup> 亀井 宏和<sup>1)</sup> 前田 慶博<sup>2)</sup> 福嶋 慶繁<sup>1)</sup>

Kohei Hayashi Soichirou Honda Hirokazu Kamei Yoshihiro Maeda Norishige Fukushima

### 1 はじめに

スケールスペースフィルタリング [1,2] は、スケールに応じて画像をベース信号と詳細信号に分解する画像解析の基本的な処理の一つである。スケールスペース処理は様々な領域で使われており、デノイジング、詳細強調、ぼけ除去、超解像、コントラスト強調、テクスチャ除去、霞除去、HDR、センサー合成、画質評価など幅広く利用されている。このスケールスペース表現には、Difference of Gaussian (DoG) などのフィルタの差分表現 [3]、ラプラシアンピラミッドなどのピラミッド表現 [4] や、ウェーブレット変換 [5] などがある。

これらのスケールスペースフィルタリングを画像強調に用いるには、詳細信号の係数を増幅させて再構成すればよい。しかしながら急峻なエッジ付近では詳細係数が大きくなるため、線形に増幅するとオーバーフロー・アンダーフローが発生し、それが halo と呼ばれる「もや」として生じる。また、スケールスペース処理中にアップサンプルやダウンサンプルを行うと、ギブス現象によりもやが更に大きくなる。そこで、halo の問題を解決するために、様々な手法が提案されてきた。

古典的方法として、様々な非線形変換により詳細信号の増幅を抑える方法が取られてきた [6-10]。係数の大きさに応じて増幅率を変えるもの [6-8] やグラディエントの大きさで変えるもの [9] が提案されている。近年でも、モルフォロジ演算の応答により増幅率を変える方法が提案されている [10]。しかし、輪郭のエッジは非常に大きな応答なため完全な halo 抑制は難しい。

そこでエッジ保存平滑化フィルタを用いることで、エッジ付近の詳細信号自体を小さくする方法も提案されている。バイラテラルフィルタ [11] による 2 層処理や、繰り返しバイラテラルフィルタによるマルチスケール処理と間引きによる高速化 [12] などが提案されている。しかし、多くのエッジ保存型マルチスケールフィルタは、繰り返しの強調プロセスを持つため、パラメータの扱いが難しい。また、近似による高速化 [13] が必要不可欠となる。

一方、local Laplacian filter (LLF) [14] などの局所コントラストを考慮して、画像強調を事前に行ってからマルチスケール処理する手法はこの問題を解決する。事前に行われる画像強調では、非線形なコントラスト変換を行うことで halo を抑制することができる。LLF は、事前コントラスト変換により Laplacian pyramid [4] を edge-aware な処理へと拡張する。ある注目画素のコントラストを改善するために画像全体のダイナミックレンジを非線形変換で操作し、その後ラプラシアンピラミッドを構築し、その係数一つを出力として採用する。つまり、1 画素あたり画像全体のピラミッド構築が必要なため計算コストが膨大である。そこで、その高速化手法 fast LLF [15] が

提案され、これは線形補間を用いることでピラミッドの構築数を激減させる。更に線形補間をフーリエ級数展開に拡張した Gaussian Fourier pyramid [16] は更にピラミッド構築数を減らし、またその DoG 拡張は畳み込みからダウンサンプリングを削除できる [17]。さらにはガイド画像付き拡張などが提案されている [18]。

しかしながら、局所コントラスト手法で取り扱うことができる現状のピラミッド表現だけでは多様な信号をとらえて解析することは難しい。そこで本論文では、信号解析の中核となるウェーブレット変換を対象として、その局所コントラスト変換拡張方法を提案する。そして、その形式が新しいウェーブレット変換表現の一つであることを示す。本論文の貢献は以下である。

- 局所コントラストアダプティブなウェーブレット変換 (local contrast adaptive wavelet transform; LCA-WT) を提案し、さらに多様なウェーブレット変換をサポート (離散ウェーブレット変換 (DWT), アトラスウェーブレット変換 (Å-trous WT), 定常ウェーブレット変換 (SWT))
- LCA-WT を値域のフーリエ級数展開を用いた冗長ウェーブレット変換で表す高速化計算方法を提案し、それは画像固有のウェーブレット変換であり、パラメータに依存しない変換である証明
- ウェーブレット係数操作を用いた画像処理拡張 (ガイド付き処理, ウェーブレット合成)

### 2 従来研究

#### 2.1 古典的な係数調整によるマルチスケール強調

従来では、マルチスケール分解により入力信号  $I \in \mathbb{R}$  を、ベース信号  $B \in \mathbb{R}$  と詳細信号  $D \in \mathbb{R}$  に再帰的に分解する ( $B_l = D_{l+1} + B_{l+1}$ )。  $I$  をレベル  $L$  分解するとき、初期分解の詳細成分は 0 ( $I = B_0 + 0$ ) とすると次となる。

$$I = B_0 = D_1 + B_1 = D_1 + D_2 + B_2 = \sum_{l=1}^L D_l + B_L \quad (1)$$

強調処理では、詳細信号を関数  $f \in \mathbb{R}$  で増幅して一つ上のベース信号に加算して復元する処理を繰り返す ( $r(B_l, B_{l+1}) = D_{l+1}f(D_{l+1}) + B_l$ 。なお、関数  $r$  はリマップ関数と呼ばれる。)。強調画像  $I_e$  は下記で定義される。

$$I_e = \sum_{l=1}^L D_l f(D_{l+1}) + B_L \quad (2)$$

関数  $f$  は、アンシャープマスクは定数値が用いられるが、halo を抑制するためには、減衰型の偶関数が使われることが多く、アプリケーションによって具体的に定義される。本研究では、関数  $f$  を  $w_r(i)$  として、強調度合いを制御する要素  $m$  を含んだガウス関数を用いる。

$$w_r(i) = m \exp\left(\frac{-i^2}{2\sigma_r^2}\right). \quad (3)$$

1) 名古屋工業大学。Nagoya Institute of Technology.

2) 芝浦工業大学。Shibaura Institute of Technology.

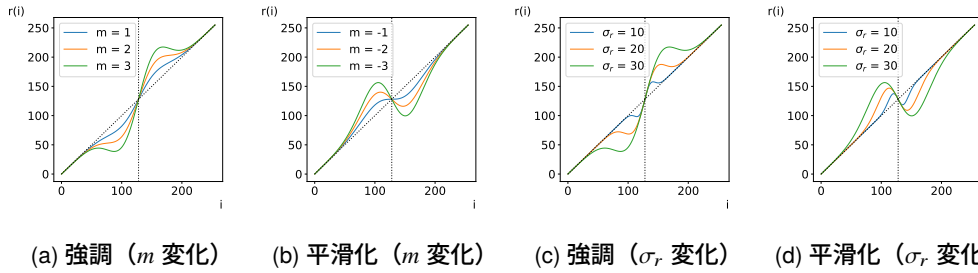


図 1  $m$  や  $\sigma_r$  を変えた時の強調, 平滑化リマップ関数 ( $g = 128$ , 変化させないパラメータは  $m = \pm 3, \sigma_r = 30$  で固定)

そして, リマップ関数は次のように定義される. ここで,  $i$  は入力画素値,  $g$  は参照画素値である.

$$r(i, g) = i + (i - g)w_r(i - g). \quad (4)$$

なお,  $m > 0$  の時, この関数は画像を強調し,  $m < 0$  の時, 画像を平滑化する. また,  $\sigma$  が大きいほど強調する輝度値域が広くなり, 小さいと逆となる. 図 1 に,  $g = 128$  の時の  $m, \sigma_r$  を変化させたリマップ関数を示す.

定数値  $m$  のリマップ関数を用いる古典的な強調手法であるアンシャープマスクは下記式で表現される.

$$O_p = I_p + m \sum_{q \in N_p} w_s(p, q)(I_p - I_q) \quad (5)$$

これを非線形関数  $w_r$  付きに変換すると, 下記となる.

$$O_p = I_p + m \sum_{q \in N_p} \exp\left(\frac{|\sum_{r \in N_q} w_s(p, q)(I_q - I_r)|^2}{-2\sigma_r^2}\right) w_s(p, q)(I_p - I_q) \quad (6)$$

このように後から係数を非線形リマップする方法は, 点  $p, q$  間以外の要素にも影響され, 大きな特徴を持つエッジの詳細信号だけを後から点で抑えると周囲の影響も失われて大きな変化がとらえづらくなる.

## 2.2 エッジアボイディングウェーブレット

エッジを考慮したウェーブレット変換として Edge-Avoiding Wavelets (EAW) [19, 20] がある. EAW ではエッジでの極端な応答を抑制するために, ウェーブレット係数を計算する予測や更新のステップでエッジストップ関数 (リマップ関数中の関数  $f$  に等しい) による重みづけを行う. このエッジストップ関数は, ガウス関数や絶対値の逆数などを用いる. 係数を計算する際にこの重みを各画素にかけることで, 画像に固有のエッジを考慮したウェーブレット変換の式が定義される. 例えば, 1次元の重み付きの CDF5/3 ウェーブレット変換は以下で表される.

$$\begin{aligned} O_{2k+1} &\leftarrow I_{2k+1} - P_{2k+1} \\ P_{2k+1} &= \frac{w_r(I_{2k+1} - I_{2k+0})I_{2k+0} + w_r(I_{2k+1} - I_{2k+2})I_{2k+2}}{w_r(I_{2k+1} - I_{2k+0}) + w_r(I_{2k+1} - I_{2k+2})}, \end{aligned} \quad (7)$$

$$\begin{aligned} O_{2k+0} &\leftarrow I_{2k+0} + U_{2k+0} \\ U_{2k+0} &= \frac{w_r(I_{2k+0} - I_{2k-1})O_{2k+0} + w_r(I_{2k+0} - I_{2k+1})O_{2k+2}}{w_r(I_{2k+0} - I_{2k-1}) + w_r(I_{2k+0} - I_{2k+1})} \end{aligned} \quad (8)$$

通常のウェーブレット変換に重みが付くため既存ライブラリが使えず, 加えて, 多くのウェーブレット変換が可分性を持つが, エッジ保存を考慮すると可分性を失う. そのため論文では可分性のない畳み込みのみを専

用に実装している. また, ガイド画像を用いた EAW による高周波成分を考慮しない補間は定義されているが, 通常のフィルタは定義されておらず, 複数画像を入力にとる画像合成等の処理を適用できない.

## 3 提案手法

本論文では, 局所コントラスト変換を用いたウェーブレット変換フィルタである LCA-WT フィルタを提案する. 古典的な処理では, 入力画像のウェーブレット係数を増減して詳細強調や平滑化を行うが, 提案手法では, 係数ひとつひとつに合わせて適切にリマップした画像全体から係数一つを計算する. つまり, ある位置の画素についてリマップされた画像をウェーブレット変換した係数のうち, 参照位置の係数一つのみを保持する操作を, 全てのレベルの層, 全ての画素に対して行うことで, 画像の全ての位置についてコントラスト強調された画像のウェーブレット係数をえる.

また, その近似高速化実装も示す. 提案手法を素朴に実装すると, リマップで参照するピラミッドの全層, 全画素数回分の離散ウェーブレット変換を行う必要がある. つまり, 画素数オーダーの離散ウェーブレット変換が必要となり, 計算コストが膨大となる. そこで局所コントラスト変換にフーリエ級数展開を用いて近似高速化した. この近似によって, 原画像を少数の  $\sin$  と  $\cos$  の周波数に分解し, それらをウェーブレット変換することで, 数回の冗長なウェーブレット変換によってエッジを考慮したウェーブレット変換を行うことができる.

更に, 近似により, リマップ関数のパラメータアダプティブ化が容易に実現できると共に, 処理前にリマップ関数を定めるのではなく, ウェーブレット変換後に任意のリマップ関数を決定できることを示す.

### 3.1 Naïve 実装

まず提案手法を素朴に実装すると, 次の流れになる (図 2). 画像  $I$  をウェーブレット変換した係数のうち, レベル  $\ell$  の層の画素位置  $p$  の係数を  $W_x^\ell[I]_p$  とする. ここで,  $x \in \{s, v, h, d\}$  はウェーブレット係数の種類を表し, 順に低周波, 垂直, 水平, 対角成分を表す.

- (1) 入力画像をウェーブレット変換し, 低周波成分の係数からなるピラミッド状のウェーブレット係数  $W_s^\ell[I]$  を構築する.
- (2) リマップ原点を指すピラミッド  $G_\ell[I]$  を  $G_{\ell-1}[I] = W_s^\ell[I]$  として構築する. ただし,  $G_0[I] = I$  である.
- (3) 以下 (4)–(5) を全層  $\ell$  の全位置  $p$  に対して繰り返す.
- (4)  $\ell$  層の位置  $p$  の値を参照してリマップ関数を適用し, コントラスト変換画像  $\tilde{I} = r(I, G_\ell[I]_p)$  を作る.

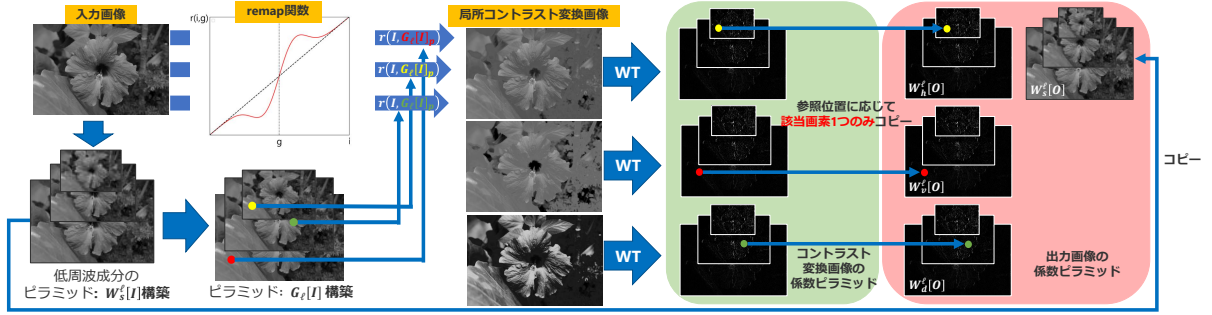


図 2 Naive 実装の処理フロー

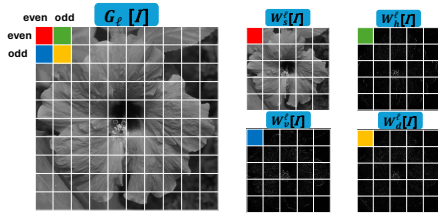


図 3 リマップ時参照位置関係。左) 参照画像  $G_\ell[I]$ , 右) 計算されるウェーブレット係数の該当位置  $W_{s,h,v,d}^\ell[I]$ 。ウェーブレット係数  $W_x^{\ell-1}$ ,  $x \in \{s, v, h, d\}$  の位置  $(i, j)$  の係数を求める際, リマップ原点の位置は垂直成分は  $(2i+0, 2j+1)$ , 水平成分は  $(2i+1, 2j+0)$ , 対角成分は  $(2i+1, 2j+1)$  の  $G$  の係数を参照

- (5)  $\tilde{I}$  をウェーブレット変換し, 垂直, 水平, 対角方向の高周波成分の係数  $W_v^\ell[\tilde{I}]$ ,  $W_h^\ell[\tilde{I}]$ ,  $W_d^\ell[\tilde{I}]$  を計算し, 位置  $p$  におけるそれぞれの係数  $W_v^\ell[\tilde{I}]_p$ ,  $W_h^\ell[\tilde{I}]_p$ ,  $W_d^\ell[\tilde{I}]_p$  を出力画像のウェーブレット係数  $W_v^\ell[O]_p$ ,  $W_h^\ell[O]_p$ ,  $W_d^\ell[O]_p$  としてコピーする。リマップ原点はレベル  $\ell$  や位置  $p$  によって変わる (図 3)。また, 出力画像の低周波成分の最上層  $W_s^{\ell_{\max}-1}[O]$  は  $W_s^{\ell_{\max}-1}[I]$  をそのままコピーする。
- (6) ウェーブレット逆変換を行い, 出力画像  $O$  を得る。

つまり Naive 実装のウェーブレット係数は次で定義され, この係数を逆変換することで出力画像を得る。

$$W_s^{\ell_{\max}-1}[O]_p = W_s^{\ell_{\max}-1}[I]_p, \quad (9)$$

$$W_{x \in \{v, h, d\}}^\ell[O]_p = W_x^\ell[r(I, G_\ell[I]_p)]_p \quad (10)$$

### 3.2 フーリエ級数展開を用いた近似高速化実装

Naive 実装のウェーブレット係数の式 (10) を近似することで, 高速化する。リマップ関数中のエッジストップ関数  $f$  は偶関数が多く, 本論文もガウス関数 (式 (3)) を用いるため, フーリエ級数展開により有限の  $\cos$  関数で近似できる。任意の位置  $p, q$  にある画素値  $I_q$  をリマップ原点  $g_p$  でリマップするときの重み関数は,

$$f(I_q - g_p) \approx m(\alpha_0 + 2 \sum_{k=1}^K \alpha_k \cos(\omega_k (I_q - g_p))) \quad (11)$$

となる。これを加法定理によって引数を分離し, 式中の  $I_q, g_p$  をそれぞれ別の  $\sin, \cos$  関数の引数にする。 $\alpha_k$  は, 区分積分法で求められる [21]。

$$f(I_q - g_p) \approx m\alpha_0 + 2m \sum_{k=1}^K \alpha_k (\cos(\omega_k I_q) \cos(\omega_k g_p) + \sin(\omega_k I_q) \sin(\omega_k g_p)) \quad (12)$$

リマップ関数は式 (12) により次で近似される。

$$r(I_q, g_p) \approx I_q + m(I_q - g_p) \left( \alpha_0 + 2 \sum_{k=1}^K \alpha_k (\cos(\omega_k I_q) \cos(\omega_k g_p) + \sin(\omega_k I_q) \sin(\omega_k g_p)) \right) \quad (13)$$

このように近似されたリマップ関数をウェーブレット係数の式 (10) に代入すると以下ようになる。

$$\begin{aligned} W_x^\ell[r(I, G_\ell[I]_p)]_p &= (1 - m\alpha_0) W_x^\ell[I]_p \\ &+ 2m \sum_{k=1}^K \alpha_k \cos(\omega_k G_\ell[I]_p) \tilde{C}_x^{\ell,k}[I]_p \\ &+ 2m \sum_{k=1}^K \alpha_k \sin(\omega_k G_\ell[I]_p) \tilde{S}_x^{\ell,k}[I]_p \\ &- 2m G_\ell[I]_p \sum_{k=1}^K \alpha_k \cos(\omega_k G_\ell[I]_p) \tilde{C}_x^{\ell,k}[I]_p \\ &- 2m G_\ell[I]_p \sum_{k=1}^K \alpha_k \sin(\omega_k G_\ell[I]_p) \tilde{S}_x^{\ell,k}[I]_p, \quad (14) \end{aligned}$$

where  $\tilde{C}_x^{\ell,k}[I] = W_x^\ell[\cos(\omega_k I)]$ ,  $\tilde{S}_x^{\ell,k}[I] = W_x^\ell[\sin(\omega_k I)]$ ,

$\tilde{C}_x^{\ell,k}[I] = W_x^\ell[I \cos(\omega_k I)]$  and  $\tilde{S}_x^{\ell,k}[I] = W_x^\ell[I \sin(\omega_k I)]$ .

入力画像の係数ピラミッド  $W_x^\ell[I]$  と, リマップで参照する係数ピラミッド  $G_\ell[I]$ , 周波数変換された画像の係数ピラミッド  $\tilde{C}_x^{\ell,k}[I]$ ,  $\tilde{S}_x^{\ell,k}[I]$ ,  $\tilde{C}_x^{\ell,k}[I]$ ,  $\tilde{S}_x^{\ell,k}[I]$  の合計  $4K+2$  個の係数ピラミッドによる積和演算で提案手法の全層, 全画素位置の係数を計算できることがわかる。

ガウス関数の導関数でリマップ関数全体を近似することでさらに計算量を半減することができる。ガウス関数を微分して以下のように式変形すると, リマップ関数の 2 項目がガウス関数の導関数を用いて表現できる。

$$f'(I_q, g_p) = -\sigma^2 (I_q - g_p) f(I_q - g_p) \quad (15)$$

リマップ関수에式 (15) を代入すると,

$$r(I_q, g_p) = I_q - \sigma^2 f'(I_q - g_p) \quad (16)$$

となり, 式 (16) 中の  $f'(I_q - g_p)$  をフーリエ級数展開し, 加法定理を用いて変形すると, リマップ関数は以下のように近似できる。ただし,  $\tilde{\alpha}_k = 2\sigma^2 \alpha_k \omega_k$  である。

$$r(I_q, g_p) \approx I_q - m \sum_{k=1}^K \tilde{\alpha}_k (\cos(\omega_k I_q) \sin(\omega_k g_p) + \sin(\omega_k I_q) \cos(\omega_k g_p)) \quad (17)$$

式 (10) に式 (17) を代入して整理すると, 提案手法の式は以下のようにリマップ関数が削除されて複数の  $\sin$  と

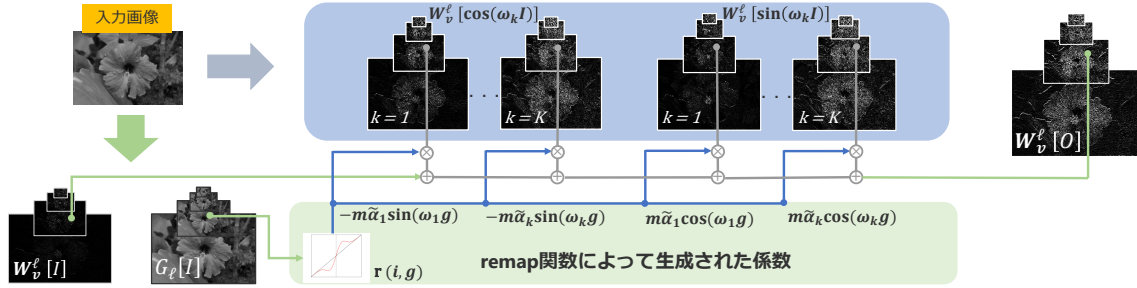


図 4 近似実装のフロー。図は垂直方向の係数であり水平，対角方向も実際は求める。

$\cos$  で変換された画像のウェーブレット変換係数の総和として表される。

$$W_x^\ell[r(I, G_\ell[I]_p)]_p = W_x^\ell[I]_p - m \sum_{k=1}^K \tilde{\alpha}_k (\sin(\omega_k G_\ell[I]_p) \tilde{C}_x^{\ell,k}[I]_p - \cos(\omega_k G_\ell[I]_p) \tilde{S}_x^{\ell,k}[I]_p), \quad (18)$$

where  $\tilde{C}_x^{\ell,k}[I] = W_x^\ell[\cos(\omega_k I)]$  and  $\tilde{S}_x^{\ell,k}[I] = W_x^\ell[\sin(\omega_k I)]$

式 (14) と比べて，周波数変換された画像に対する係数ピラミッドの数が  $2K$  個に半減している。

このように，近似実装では式 (9)，(18) によって， $2K+2$  個の係数ピラミッドのみで局所コントラストを考慮したウェーブレット係数を求めることができる。 $W_x^\ell[I]$  と  $G_\ell[I]$  は同時に求めることができるため，このピラミッド群を構築するには  $2K+1$  回の冗長なウェーブレット変換が必要となる。画素数に比例する回数のウェーブレット変換が必要な Naive 実装に比べると，近似実装では  $2K+1$  回のウェーブレット変換を行うだけでよく，計算コストが大幅に減っている。なお，代入するリマップ関数の近似式は，すべて同じである必要がなく，方向や画素位置によって，式 (3) の  $m$  や  $\sigma_r$  を変更できる。つまり，水平，垂直，対角方向のそれぞれに対して，リマップ関数の強調度合いを変えられるため，各方向ごとに異なる強さでコントラスト変換を行った画像のウェーブレット係数を求めることができる。近似実装で 1 方向のウェーブレット係数を求めるフローを図 4 に示す。

### 3.3 事後コントラスト評価

リマップ関数を画素ごとに変え，パラメータアダプティブに拡張 (PA 拡張) することで，平坦な領域でノイズのような模様が強調されることを防ぐなどフィルタ応用を広げられる。Naive 実装では，画素ごとにリマップ関数を定義することで PA 拡張を行うことができるが，計算コストの問題が残る。ここで，フーリエ級数展開したウェーブレット変換係数群の式 (18) に着目すると，リマップ関数の形状に依存せず，入力画像  $I$  のみに依存し，リマップ関数に依存しているのは，リマップ原点  $G$  と重み  $\tilde{\alpha}_k$  となる。そのため近似高速化実装における PA 化は，入力画像をフーリエ級数展開したウェーブレット変換を一度行えば，適切な係数を乗ずるだけで実現できるため計算コストを削減できる。これは，この変換が画像固有の変換であり，後から画像コントラストを表現できることも示している。

## 4 係数操作による LCA-WT の拡張

提案手法で得られたウェーブレット係数を操作することで，以下 4 つの拡張を考える。

### 4.1 エッジ付近の極端な応答を抑制する重み実装

提案手法は，使用するウェーブレット変換の種類によってエッジ付近での応答が大きくなる場合がある。例えば，リフティングウェーブレットである CDF9/7 ウェーブレットは，信号を偶数番地と奇数番地に分けて間引きながら処理するため，シフト不変性を持たず，逆変換ではアップサンプルを伴いながら係数が逆変換される。提案手法は係数の位置によってそれぞれ異なる位置の画素値を参照してリマップする。そのため，リフティングウェーブレット変換を用いた場合は逆変換する際に畳み込まれる各係数について，リマップで参照された値の差が大きくなる急峻なエッジ周辺で，コントラスト変換の強さが大きい場合に極端な応答が発生して halo が生じやすい。そこで，この問題を改善するためにウェーブレットの親子関係を利用した重みづけ処理を行う。

レベル  $\ell+1$  の層は，解像度がレベル  $\ell$  の層の半分である。つまり，リマップで参照する層の画素位置  $p$  の値と，一つ上の層の画素位置  $\frac{p}{2}$  の値との差分を取ることによって，位置  $p$  の係数が周囲の係数に比べて差が大きい値でリマップされているかどうかを評価できる。その差分が大きくなるほど値が小さくなる重みを  $m$  に乗ずることで，周囲の係数に比べてリマップの値の差が大きい係数の強調度合いを制御できる。

具体的には，以下のような式に基づいて，レベル  $\ell$ ，位置  $p$  の係数での重み  $w(p, \ell)$  を計算する。 $C(\ell)$  は各レベルで強調度合いを制御する差の範囲を表し，この値が小さくなるほどわずかな差でも強調を抑制するようになる。 $C(\ell)$  は，レベルが上がるほど値が小さくなるような以下の関数を用いる。また，関数  $L(\ell)$  によってレベルによって係数の強調度合いを変える。

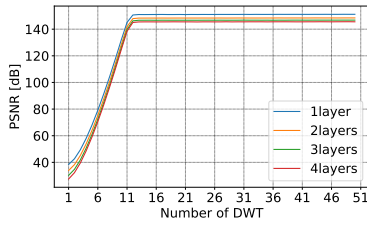
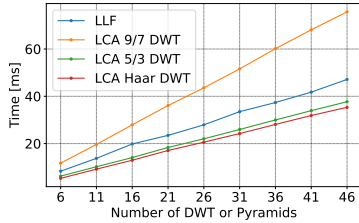
$$w_\ell(p, \ell) = L(\ell) \exp\left(\frac{-\|g_p^\ell - g_{\frac{p}{2}}^{\ell+1}\|_1}{C(\ell)}\right),$$

$$C(\ell) = \frac{A}{\ell+1}, \quad L(\ell) = \frac{1}{1 + (\ell_{\max}-1 - \ell - 1) \cdot B}. \quad (19)$$

$A, B$  はパラメータであり  $A = 80, B = 0.2$  を用いた。この処理では，重みは  $m$  に乗じるため，出力ウェーブレット係数を  $mw_\ell(p, \ell)$  で増減することに等しい。

### 4.2 ガイド画像付き強調

入力画像とは異なるガイド画像を使うことで，ジョイントフィルタのような処理が可能となる。提案手法では，ガイド画像を提案手法によってウェーブレット変換して得られた係数に対して，最上位層の低周波成分に入力画像の通常のウェーブレット変換係数で置き換えて逆変換を行うことで，ガイド付きの画像強調を行う。

図 5 LCA 9/7 DWT の PSNR ( $\sigma_r = 30, m = 2, \ell_{\max} = 1, \dots, 4$ )図 6 各 DWT 手法の計算時間 ( $\sigma_r = 30, m = 2, \ell_{\max} = 4$ )

入力画像を  $I$ , ガイド画像を  $G$  として, ガイド画像の LCA-WT フィルタの出力を  $O_G$  とすると, 入力画像  $I$  のガイド画像付き強調の出力画像  $O$  のウェーブレット係数の式は次のように表現できる.

$$W_s^\ell[\mathbf{O}] = \begin{cases} W_s^\ell[I] & (\ell = \ell_{\max} - 1), \\ \text{DC}_{x \in \mathcal{D}}(W_x^{\ell+1}[\mathbf{O}]) & (\text{else}). \end{cases} \quad (20)$$

$$W_{x \in \{v, h, d\}}^\ell[\mathbf{O}] = W_x^\ell[\mathbf{O}_G] \quad (20)$$

ここで,  $\mathcal{D}$  はウェーブレット係数の方向を表す集合  $\{s, v, h, d\}$  である. また  $\text{DC}_{x \in \mathcal{D}}$  は, 前レベルの層のウェーブレット係数から逆変換して DC 成分 (つまり方向  $s$ ) を取り出すオペレータである. この係数を逆変換することで出力画像を得る. つまり, 最上位層の低周波成分を入力画像のウェーブレット変換の低周波成分に置き換えたものを逆変換した処理である.

### 4.3 画像合成 (Wavelet fusion)

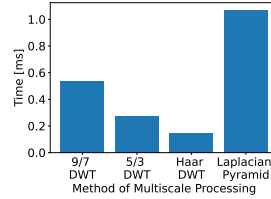
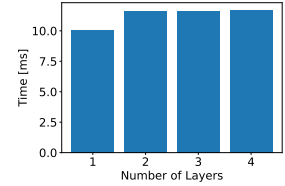
Wavelet fusion による画像合成では, 複数枚の画像のウェーブレット係数について, 値の大きい係数を採用したり, 係数の平均をとったりするなどして合成した係数を逆変換して入力画像を合成した出力画像を得る. 提案手法でも同様の処理が行える. 2 枚の入力画像  $I_1, I_2$  を合成する場合の出力のウェーブレット係数の式は次のように表現できる. なお,  $\text{fusion}(\cdot, \cdot)$  は, 係数操作の関数であり, 平均や最大値など様々な処理を取りうる.

$$W_s^\ell[\mathbf{O}] = \begin{cases} \text{fusion}(W_s^\ell[I_1], W_s^\ell[I_2]) & (\ell = \ell_{\max} - 1), \\ \text{DC}_{x \in \mathcal{D}}(W_x^{\ell+1}[\mathbf{O}]) & (\text{else}). \end{cases} \quad (21)$$

$$W_{x \in \{v, h, d\}}^\ell[\mathbf{O}] = \text{fusion}(W_x^\ell[\mathbf{O}_{I_1}], W_x^\ell[\mathbf{O}_{I_2}]) \quad (21)$$

## 5 実験および結果

実験では, 近似精度評価と計算時間, ウェーブレット基底による出力画像の違いを評価した. また, Core i9-9900K 3.60GHz (8 コア 16 スレッド) 上で, C++ 言語を用いて OpenMP による並列化と AVX によるベクトル化を用いたコードを Visual Studio 2019 でコンパイルした. 三角関数計算は 1 層目のみテーブル参照による高速化を行った [22].

図 7 マルチスケール処理単体の計算時間 ( $\ell_{\max} = 4$ )図 8 LCA 9/7 DWT の各レイヤー数の計算時間 ( $K = 6$ )

### 5.1 近似の精度および速度評価

近似精度を評価するために, Naïve 実装の出力画像を正解として, 1 から 4 までのレイヤー数について近似次数  $K$  を変えながら出力画像との peak signal-to-noise ratio (PSNR) をそれぞれ計測し, Kodak が提供する 24 枚の画像で PSNR の平均を評価した. なお, 提案手法は 1 チャネル画像のみ対応しているため, YUV 色空間の Y チャネルにフィルタすることでカラー出力を実現した.

図 5 より, 近似次数が 6 を超えると, どのレイヤー数でも PSNR が 60 dB に達する. 59 dB を超えると 8bit 画像では同じ出力である. つまり, ウェーブレット変換を  $2 * 6 + 1 = 13$  回行うと, Naïve と同等な結果が得られる.

図 6 に CDF9/7, CDF5/3, Haar 基底による DWT を用いた提案手法と, ガウシアンフーリエ LLF [16] の計算時間を示す. どの手法も近似次数にほぼ線形である. マルチスケール処理単体の実行時間を測定すると, 図 7 のようになり, CDF9/7 DWT を除き, 各手法間の計算時間の差は使用されるマルチスケール処理の計算コストに依存するものとわかる. CDF9/7 DWT を用いた提案手法の計算時間が長い原因は, ウェーブレット変換時の画像スキャン回数が多くキャッシュ効率が悪いことが考えられ高効率実装が望まれる. また, 図 8 に CDF9/7 DWT を用いた提案手法のレイヤー数ごとの計算時間を示す. レイヤー数を増やすと計算時間が増加することがわかる.

### 5.2 パラメータアダプティブ化 (PA 化)

図 9 に PA 化した結果を示す. リマップで参照する値はウェーブレット係数の低周波成分をもとにしているため, リマップで参照する係数と一つ上のレベルの層の係数の差は, 平坦な領域では小さくなる. そこで, リマップで参照する画素値と一つ上のレベルの層の画素値の差が 3 以下の場合には  $m$  および  $\sigma_r$  を半分にする事で平坦な領域でのコントラスト変換の強さを抑えた. 入力画像 (図 9a) を提案手法で強調した画像 (図 9b) とその PA 拡張 (図 9c) を比較すると, PA 拡張では背景の平坦な領域での過強調を抑えつつ, 岩の部分は通常の場合と同等に強調できている. また, PA 拡張でも通常実装でも, まったく同一の  $\sin, \cos$  で周波数変換されたウェーブレット係数群から出力の係数を求めており (図 9d, 9e), 係数群に乗ずる値 ( $m$  や  $\tilde{\alpha}_k$ ) のみが異なる. つまり,  $m$  や  $\tilde{\alpha}_k$  によって, どのようなリマップ関数を用いて入力画像のコントラストを操作するかを, ウェーブレット変換を行った後に決定できることを示している.

### 5.3 ウェーブレット基底による違い

提案手法は様々なウェーブレット変換に対応するため, 入力 (図 10a) に対して比較のために LLF (図 10b) を, DWT である CDF9/7 (図 10c) に加えて, À-trous WT (図 10d), SWT (図 10e) を提案手法に適用した.



(a) 入力画像 (768×512) (b) 提案強調 (c) PA 提案強調 (d) WT 係数 (sin,  $K = 1$ ) (e) WT 係数 (sin,  $K = 6$ )  
 図 9 パラメータアダプティブ (PA) 拡張結果 ( $\sigma_r = 20, \ell_{\max} = 4, K = 6, m = 5$ )



(a) 入力画像 (768×512) (b) LLF [14] (c) LCA CDF9/7 DWT (d) LCA À-trous WT (e) LCA SWT  
 図 10 異なるウェーブレット基底を用いて強調した場合の比較 ( $\sigma_r = 30, \ell_{\max} = 4, K = 6, m = 3$ )



(a) 入力拡大 (b) 全方向 (c) 重み付き全方向 (d) 水平方向 (e) 垂直方向 (f) 対角方向

図 11 方向性強調 ( $\sigma_r = 25, \ell_{\max} = 4, K = 6, m = 4$ )

信号を間引く DWT と比べて、間引きを行わない À-trous WT や SWT はエッジでの極端な応答が抑制されている。高速処理やデータ圧縮を目的とする場合は間引きを行い、コンパクトな係数が得られる CDF9/7 を用い、計算時間はかかるが halo を発生させることなく強調することを目的とする場合は À-trous WT や SWT を用いるなど、目的によるウェーブレット基底の使い分けができると考えられる。

## 6 提案手法を用いたアプリケーション

次に、ウェーブレット変換を用いる 4 つのアプリケーション (方向性強調, RGB-IR 画像合成, 多焦点画像合成, ガイド付き処理) の例を示す。

### 6.1 方向性のある強調

提案手法はウェーブレット変換を扱うため、水平、垂直、対角方向のそれぞれに対して異なる強さで強調処理を行うことができる。そこで、図 10a の拡大図 11a に対して、全方向 (図 11b)、重み付き実装の全方向 (図 11c) 水平 (図 11d)、垂直 (図 11e)、対角 (図 11f)、に詳細強調を行い、エッジでの応答を比較した。全方向強調した場合で、図 11b と図 11c を比べると、重みによって画像中の帽子と空の境界のようなエッジ付近でのもやが抑えられていることがわかる。図 11d~11f より、特定エッジ方向に対して詳細強調が確認できる。

### 6.2 赤外線カメラ画像と可視光カメラ画像の合成

可視光カメラ画像と赤外線カメラ画像を合成することで、可視画像の明部や暗部で見づらい物体を合成画像で可視化できる。この処理は、可視と赤外線画像のウェーブレット係数を合成することで実現できる [23]。

実験では、2 枚の入力画像をウェーブレット変換した

後、係数の平均をとって出力画像の係数を計算した。また、可視画像の係数を強調することで物体などの視認性を高めた。入力画像には位置合わせされた可視画像と赤外線画像のデータセット [24,25] に含まれる画像を用いた。可視画像 (図 12a, 12f) と赤外線画像 (図 12b, 12g) について、通常のウェーブレット変換 (図 12c, 12h) と提案手法を用いて画像合成を実行した (強調無し: 図 12d, 12i), 強調有り: 図 12e, 12j)。図 12a と 12b の合成では、暗部で見えなくなっていた人が視認できるようになっており、図 12f と 12g の合成では、明部で見えなくなっていた雲や路面のひびが視認できるようになっている。また、単純なウェーブレット強調と比べて提案手法は急峻なエッジ付近での極端な応答を抑えられている。例として、図 12h の右上の白い建物のエッジでみられるような halo が、図 12j では抑制されている。

### 6.3 合焦画像の合成

焦点位置の異なる複数画像を合成することで多領域で焦点の合った画像を生成できる。これは、複数枚の画像をウェーブレット変換した後、各画像の係数から焦点の合った領域の係数を出力の係数として選択して逆変換することで実現できる [26,27]。

実験では、高周波は絶対値の大きい係数を合焦領域として選択し、低周波は係数の平均を用いた。低周波は係数の大きさに基づいて合焦領域を判定することが難しいため、出力画像に低周波が原因のぼけが現れることがある。そこで、入力画像をウェーブレット変換した係数を強調して合焦画像を生成することで出力でぼけが目立たないようにした。合焦画像生成のためのデータセット [28] に含まれる画像を入力画像として実験を行った。画像の一部で焦点のぼけた画像 2 枚 (図 13a,

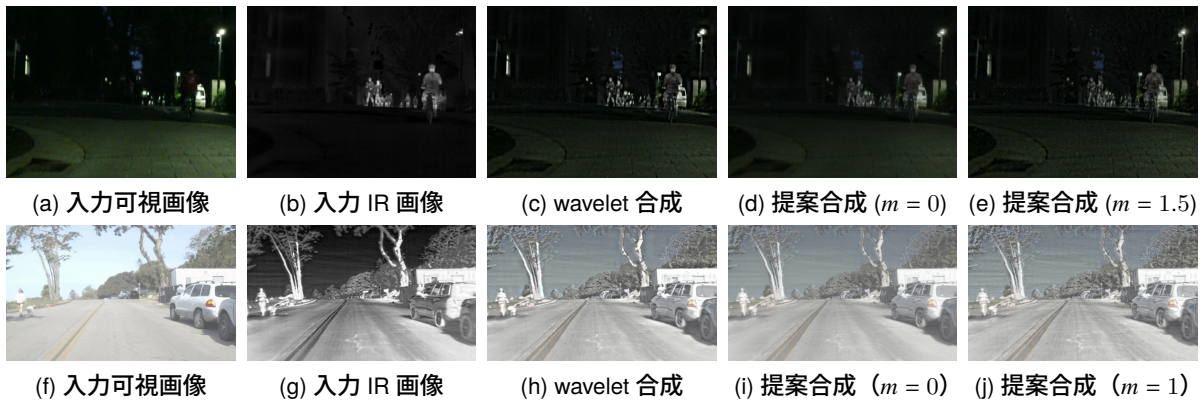


図 12 赤外線画像と可視画像の合成

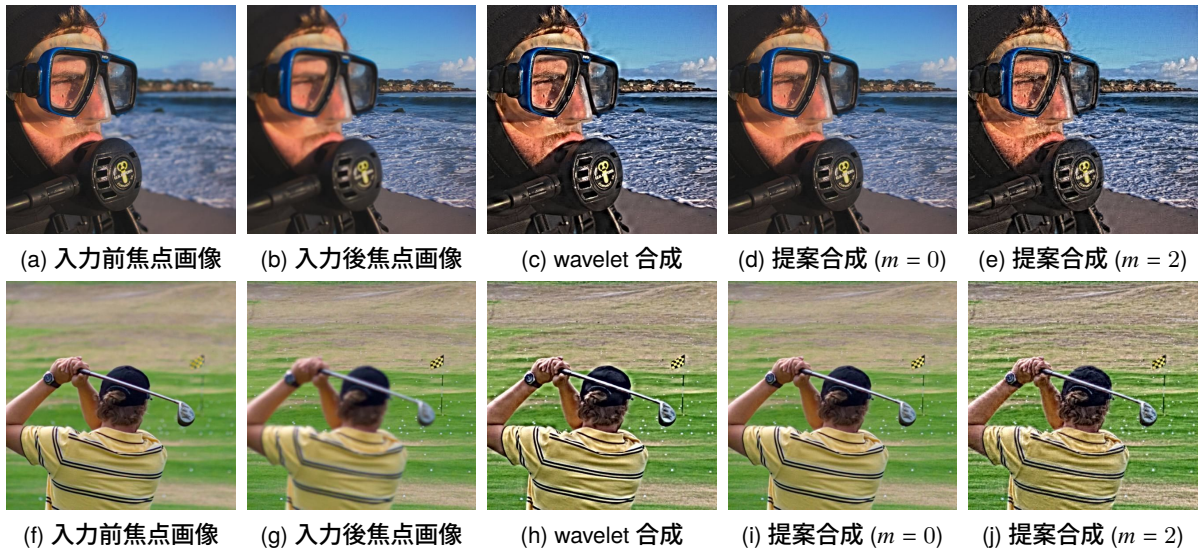


図 13 合焦画像の生成

13b および図 13f, 13g) に対して, 通常のウェーブレット変換 (図 13c, 13h) と提案手法を用いて合焦画像の合成を実行した (強調無し: 図 13d, 13i, 強調有り: 図 13e, 13j). 強調することで, より鮮明な画像が得られている. また, 単純なウェーブレット変換で強調合成した場合と比べ, 提案手法はエッジ付近の極端な応答が抑えられており, 図 13c のゴーグル上側でみられる halo が, 図 13e では抑制されている.

#### 6.4 ガイド付き強調処理

フラッシュを炊いて撮影された画像はノイズを抑えられるが色が正しく再現されず, 一方でフラッシュを炊かずに撮影された画像は色再現は正しいが, 高ノイズである. そこで, ノーフラッシュ画像の強調の際にフラッシュ画像をガイド画像として使用し, ノイズを抑えつつノーフラッシュ画像を強調する.

実験では, フラッシュ画像 (図 14a) とノーフラッシュ画像 (図 14b) に対して, ノーフラッシュ画像を単純ウェーブレット強調 (図 14c), ガイド付き単純ウェーブレット強調 (図 14d), ノーフラッシュ画像を LLF [14] (図 14e), ガイド画像でリマップ関数の重みを計算する Joint LLF (JLLF) [18] (図 14j), ノーフラッシュ画像を提案手法 (図 14h) 提案手法のガイド付き処理 (図 14i) によって詳細強調した. 図 14f, 図 14g を見ると, ノーフ

ラッシュ画像を強調した場合はノイズが多く, ガイド付きで強調した場合はノイズが少なく模様の詳細情報が視認できるようになっている. ただし, JLLF についてはノイズを抑えることができておらず, 提案手法のガイド付き処理のように係数の置き換えを行ったほうがノイズを抑えられることがわかる. また, 単純ウェーブレット強調ではエッジで halo が発生しているが, 提案手法による強調では halo が抑制されている.

#### 7 まとめ

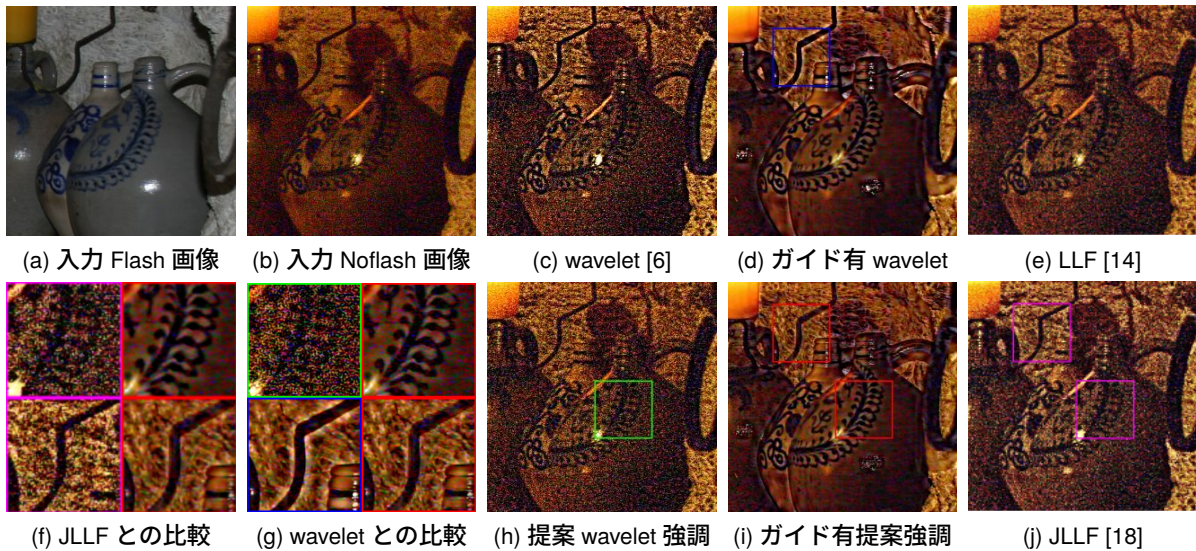
本研究では, 局所コントラスト変換マルチスケール処理をウェーブレット変換に拡張し, 様々な応用例を示した. さらに, フーリエ級数展開によって高速化し, 冗長なウェーブレット変換 10 回程度で提案手法を実行できるような新たなウェーブレット表現を提案した.

##### 謝辞

本研究は科研費 (JP24K02960) および (独) 環境再生保全機構の環境研究総合推進費 (JPMEERF2022M01) の助成による.

##### 参考文献

- [1] 飯島泰蔵, “パターン観測に関する基礎理論,” in 電気通信学会オートマトンと自動制御研究会, 1959.
- [2] A. Witkin, “Scale-space filtering,” in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 1983.
- [3] D. Marr and E. Hildreth, “Theory of edge detection,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol.

図14 フラッシュ・ノーフラッシュ画像の強調 ( $m = 3, \sigma_r = 30$  で強調)

- 207, no. 1167, pp. 187–217, 1980.
- [4] P. J. Burt and E. H. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [5] A. Grossmann and J. Morlet, “Decomposition of hardy functions into square integrable wavelets of constant shape,” *SIAM journal on mathematical analysis*, vol. 15, no. 4, pp. 723–736, 1984.
- [6] P. Vuylsteke and E. P. Schoeters, “Multiscale image contrast amplification (musica),” in *Proceedings of SPIE*, vol. 2167, 1994, pp. 551–560.
- [7] S. Dippel, M. Stahl, R. Wiemker, and T. Blaffert, “Multiscale contrast enhancement for radiographies: Laplacian pyramid versus fast wavelet transform,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 4, 2002.
- [8] Y. Li, L. Sharan, and E. H. Adelson, “Compressing and companding high dynamic range images with subband architectures,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 836–844, 2005.
- [9] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 249–256, 2002.
- [10] X. Deng, Y. Zhang, X. Zhao, and H. Li, “Halo-free image enhancement through multi-scale detail sharpening and single-scale contrast stretching,” *Signal Processing: Image Communication*, vol. 113, p. 116923, 2023.
- [11] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 1998, pp. 839–846.
- [12] R. Fattal, M. Agrawala, and S. Rusinkiewicz, “Multiscale shape and detail enhancement from multi-light image collections,” *ACM Transactions on Graphics*, vol. 26, no. 3, p. 51, 2007.
- [13] K. Sugimoto, N. Fukushima, and S. Kamata, “200 fps constant-time bilateral filter using svd and tiling strategy,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2019.
- [14] S. Paris, W. Hasinoff, and J. Kautz, “Local laplacian filters: Edge-aware image processing with a laplacian pyramid,” *ACM Transactions on Graphics*, vol. 30, no. 4, 2011.
- [15] M. Aubry, S. Paris, J. Kautz, and F. Durand, “Fast local laplacian filters: Theory and applications,” *ACM Transactions on Graphics*, vol. 33, no. 5, 2014.
- [16] Y. Sumiya, T. Otsuka, Y. Maeda, and N. Fukushima, “Gaussian fourier pyramid for local laplacian filter,” *IEEE Signal Processing Letters*, vol. 29, pp. 11–15, 2022.
- [17] K. Hayashi, Y. Maeda, and N. Fukushima, “Local contrast enhancement with multiscale filtering,” in *Proceedings of Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2023, pp. 765–770.
- [18] H. Du, X. Jin, and P. J. Willis, “Two-level joint local laplacian texture filtering,” *Visual Computer*, vol. 32, pp. 1537–1548, 2016.
- [19] R. Fattal, “Edge-avoiding wavelets and their applications,” *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 22: 1–10, 2009.
- [20] Y. Sumiya, H. Kamei, K. Ishikawa, and N. Fukushima, “Vectorized computing for edge-avoiding wavelet,” in *International Workshop on Advanced Image Technology (IWAIT)*, vol. 12177, 2022, pp. 23–28.
- [21] Y. Sumiya, N. Fukushima, K. Sugimoto, and S. Kamata, “Extending compressive bilateral filtering for arbitrary range kernel,” in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2020.
- [22] Y. Maeda, N. Fukushima, and H. Matsuo, “Effective implementation of edge-preserving filtering on cpu microarchitectures,” *Applied Sciences*, vol. 8, no. 10, 2018.
- [23] A. D. Sappa, J. A. Carvajal, C. A. Aguilera, M. Oliveira, D. Romero, and B. X. Vintimilla, “Wavelet-based visible and infrared image fusion: a comparative study,” *Sensors*, vol. 16, no. 6, p. 861, 2016.
- [24] H. Xu, J. Ma, Z. Le, J. Jiang, and X. Guo, “Fusiondn: A unified densely connected network for image fusion,” in *Proceedings of AAAI Conference on Artificial Intelligence*, 2020.
- [25] L. Tang, J. Yuan, and J. Ma, “Image fusion in the loop of high-level vision tasks: A semantic-aware real-time infrared and visible image fusion network,” *Information Fusion*, vol. 82, pp. 28–42, 2022.
- [26] P. Singh, M. Diwakar, X. Cheng, and A. Shankar, “A new wavelet-based multi-focus image fusion technique using method noise and anisotropic diffusion for real-time surveillance application,” *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 1051–1068, 2021.
- [27] 白井啓一郎, 野村和史, 池原雅章, “ウェーブレット変換による合焦画像の作成,” *電子情報通信学会論文誌 A*, vol. 88, no. 10, pp. 1154–1162, 2005.
- [28] M. Nejati, S. Samavi, and S. Shirani, “Multi-focus image fusion using dictionary-based sparse representation,” *Information Fusion*, vol. 25, pp. 72–84, 2015.