

# 3D画像フィルタースイート 技術リファレンス

## BL47XU SPring-8 マイクロ/ナノCT画像処理

## 第1章 フィルタリングの基礎概念

### 1.1 CT画像におけるノイズの性質

シンクロトロン放射光CTで取得された画像のノイズは、主に以下の成分からなる。

**ショットノイズ（ポアソンノイズ）** X線光子の統計的揺らぎに起因する。低フォトン計数条件（高解像度・短露光）で支配的になる。画素値が大きいほどノイズが小さくなる（信号依存性）。

**読み出しノイズ** 検出器（シンチレータ+カメラ）の電子回路に起因する加算的ガウスノイズ。

**リング・アーティファクト** 回転軸に対して同心円状に現れるアーティファクト。検出器の感度むらに起因する。

**実用上の近似** 再構成後のCT像では、ショットノイズは対数変換と再構成処理を経てほぼガウス分布に近似でき、加算的白色ガウスノイズ（AWGN）モデルが多くのフィルタで仮定される。

$$f(\mathbf{x}) = u(\mathbf{x}) + n(\mathbf{x}), \quad n \sim \mathcal{N}(0, \sigma^2)$$

ここで  $f$  は観測画像、 $u$  は真の画像、 $n$  はノイズである。

### 1.2 フィルタリングのトレードオフ

理想的なフィルタは以下を同時に達成するが、完全な両立は不可能である。

- **ノイズ除去能力**：ノイズを効果的に抑制する
- **エッジ保存性**：構造境界をシャープに保つ
- **テクスチャ保存性**：細かい内部構造を失わない

各フィルタはこのトレードオフの異なる点を選んでいる。

### 1.3 ノイズ標準偏差 $\sigma$ の自動推定

本ツール群では、指定がない場合に以下の方式でノイズ標準偏差を自動推定する。

#### 隣接画素差分法

画像中央50%の領域から隣接画素差分を計算し、i.i.d. ガウスノイズの仮定のもとで推定する。

$$\sigma_{\text{est}} = \sqrt{\frac{1}{2N} \sum_x \left( f(x+1) - f(x) \right)^2}$$

独立な2つのノイズ項の差の分散が  $2\sigma^2$  になることを利用する。信号の勾配成分も含まれるため、この推定値は真の  $\sigma$  よりやや大きくなる傾向があるが、各フィルタのパラメータ自動設定においてその特性を考慮した係数が乗じられている。

# 第2章 各フィルタの詳細

## 2.1 Bilateral Filter（双方向フィルタ）

### 概要

Bilateral Filter（BF）は Tomasi & Manduchi (1998) によって提案された、空間距離と輝度差の両方を考慮する重み付き平均フィルタである。空間的に近く、かつ輝度が近い画素を優先的に平均化することで、エッジを保存しながら平滑化を行う。

### 数式（3D版）

出力画素  $u(\mathbf{x})$  は以下のように定義される。

$$u(\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} w_s(\mathbf{x}, \mathbf{y}) \cdot w_r(f(\mathbf{x}), f(\mathbf{y})) \cdot f(\mathbf{y})}{\sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} w_s(\mathbf{x}, \mathbf{y}) \cdot w_r(f(\mathbf{x}), f(\mathbf{y}))}$$

### 空間重み（Gaussian kernel）

$$w_s(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2\sigma_s^2}\right)$$

### 輝度重み（Range kernel）

$$w_r(f(\mathbf{x}), f(\mathbf{y})) = \exp\left(-\frac{(f(\mathbf{x}) - f(\mathbf{y}))^2}{2\sigma_r^2}\right)$$

$\mathcal{N}(\mathbf{x})$  はカーネルサイズで定義される近傍領域。

### パラメータ

パラメータ	記号	定義	単位	デフォルト
kernel_size	$2k+1$	カーネルの一辺のサイズ（奇数）	pixel	5
spatial_sigma	$\sigma_s$	空間方向のガウス標準偏差。大きいほど広い範囲を平均化	pixel	2.0
intensity_sigma	$\sigma_r$	輝度方向のガウス標準偏差。 <b>エッジ保存の感度を制御する最重要パラメータ</b>	[画素値と同単位]	自動

### intensity\_sigma の自動設定

ビット深度	デフォルト値
8-bit	$256 / 3 \approx 85$
16-bit	$65536 / 3 \approx 21845$
32-bit float	$\text{dynamic range} / 3$

パラメータ調整指針

- `intensity_sigma` が小さい → エッジ保存が強い（ノイズ除去は弱い）
- `intensity_sigma` が大きい → ノイズ除去が強い（エッジがぼける）
- `kernel_size` は `spatial_sigma` に対して  $2 \times \lceil 3\sigma_s \rceil + 1$  以上が望ましい
- `spatial_sigma` は典型的に 1.0～5.0 の範囲

適した試料

骨、繊維、多孔質材料など、エッジが明確な中程度のテクスチャを持つ試料。

2.2 Gaussian Filter（ガウシアンフィルタ）

概要

最も基礎的な線形平滑化フィルタ。3次元ガウス関数を畳み込みカーネルとして使用する。エッジを保存する機能はないが、計算が高速で結果が予測しやすい。

数式（3D版）

$$u(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} G_{\sigma}(\mathbf{x} - \mathbf{y}) \cdot f(\mathbf{y})$$

3Dガウスカーネル（正規化済み）

$$G_{\sigma}(\mathbf{d}) = \frac{1}{Z} \exp\left(-\frac{d_x^2 + d_y^2 + d_z^2}{2\sigma^2}\right)$$

$Z$  は総和が1になるよう正規化する定数。

カーネルサイズの自動計算

$$\text{kernel\_size} = 2 \times \lceil 3\sigma \rceil + 1$$

この式により、カーネルはガウス関数の  $\pm 3\sigma$ （全面積の99.7%）をカバーする。

パラメータ

パラメータ	記号	定義	単位	デフォルト
<code>sigma</code>	$\sigma$	ガウス標準偏差。唯一の調整パラメータ	pixel	2.0

カーネルサイズは `sigma` から自動計算されるため、ユーザーが指定する必要はない。

パラメータ調整指針

- $\sigma = 0.5 \sim 1.0$  : 微細ノイズのみ除去、シャープネスをほぼ維持
- $\sigma = 1.0 \sim 2.0$  : 標準的な軽い平滑化
- $\sigma = 2.0 \sim 4.0$  : 顕著な平滑化、細かい構造が失われ始める
- $\sigma > 4.0$  : 強い平滑化、微細構造は消失

適した試料

均質な材料、ノイズの軽い除去、前処理（他の解析の前段）。

## 2.3 Median Filter（メディアンフィルタ）

### 概要

カーネル内の全画素値を並べてその中央値（メジアン）を出力する非線形フィルタ。統計的外れ値（スパイクノイズ、ソルト&ペッパーノイズ）の除去に特に有効。

### 数式（3D版）

$$u(\mathbf{x}) = \text{median}\{f(\mathbf{y}) \mid \mathbf{y} \in \mathcal{N}(\mathbf{x})\}$$

$\mathcal{N}(\mathbf{x})$  は  $\mathbf{x}$  を中心とする立方体カーネル内の全画素。カーネルサイズが  $(2k+1)^3$  の場合、ソートして  $\lfloor (2k+1)^3 / 2 \rfloor + 1$  番目の値を取る。

### パラメータ

パラメータ	記号	定義	単位	デフォルト
<code>kernel_size</code>	$2k+1$	カーネルの一辺のサイズ（奇数）。 <b>唯一の調整パラメータ</b>	pixel	3

GPU版では効率的なデバイス内ソート（insertion sort）を使用。

### パラメータ調整指針

- `kernel_size = 3` :  $3 \times 3 \times 3 = 27$ 画素のメジアン。標準的な孤立ノイズ除去
- `kernel_size = 5` :  $5 \times 5 \times 5 = 125$ 画素。やや大きな外れ値クラスタに対応
- `kernel_size = 7` : 大きなアーティファクトの除去（計算コスト大）
- 奇数のみ指定可能（3～21）

### 適した試料

スパイクノイズ、ゴースト、孤立したアーティファクトを含む画像。構造がシャープに保たれる。

## 2.4 Non-Local Means Filter（非局所平均フィルタ）

### 概要

Buades et al. (2005) により提案されたフィルタ。画像全体（または探索範囲内）から、対象画素周辺のパッチと類似したパッチを探し出し、それらの中心値を重み付き平均する。「類似した構造は類似した値を持つべき」という仮定に基づく。繰り返し構造（骨梁、繊維など）に特に有効。

### 数式（3D版）

$$u(\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \sum_{\mathbf{y} \in \mathcal{S}(\mathbf{x})} w(\mathbf{x}, \mathbf{y}) \cdot f(\mathbf{y})$$

正規化定数

$$Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{S}(\mathbf{x})} w(\mathbf{x}, \mathbf{y})$$

重み（パッチ類似度に基づく）

$$w(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{d^2(\mathbf{x}, \mathbf{y})}{h^2}\right)$$

正規化パッチ距離（SSD）

$$d^2(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{p} \in \mathcal{P}} \left[ f(\mathbf{x} + \mathbf{p}) - f(\mathbf{y} + \mathbf{p}) \right]^2$$

$\mathcal{P}$  はパッチ領域（半径  $r_p$  の立方体）、 $\mathcal{S}(\mathbf{x})$  は探索範囲（半径  $r_s$  の立方体）、 $|\mathcal{P}|$  は有効比較画素数。

自分自身（ $\mathbf{x} = \mathbf{y}$ ）の重みは、全探索で見つかった最大重みを使用する（標準的NLM実装）。

パラメータ

パラメータ	記号	定義	単位	デフォルト
patch_radius	$r_p$	パッチの半径。パッチサイズ $= (2r_p+1)^3$	pixel	1 (→ 3×3×3)
search_radius	$r_s$	探索範囲の半径。探索サイズ $= (2r_s+1)^3$	pixel	3 (→ 7×7×7)
h	$h$	<b>フィルタリング強度</b> 。ノイズ標準偏差 $\sigma_n$ と同程度の値が適切	[画素値と同単位]	自動 ( $\approx 1.2\sigma_n$ )

計算量

1ボクセルあたりの演算数は  $(2r_s+1)^3 \times (2r_p+1)^3$  に比例する。デフォルト設定では  $7^3 \times 3^3 = 343 \times 27 = 9261$  回の差分計算。

パラメータ調整指針

h の調整（最重要）

h の値	効果
$h < 0.5\sigma_n$	ノイズが残る（過小平滑化）
$h \approx \sigma_n$	シャープさとノイズ除去のバランス
$h \approx 1.2\sigma_n$	<b>推奨範囲</b> （本ツールのデフォルト）
$h > 2\sigma_n$	ぼやけが生じる（過大平滑化）

patch\_radius と search\_radius の関係

- `patch_radius` が大きいほど：類似度評価が安定するがテクスチャが失われやすい
- `search_radius` が大きいほど：遠くの類似構造も参照できるが計算時間が急増
- 典型的には `search_radius = 2 × patch_radius + 1` 程度が目安

## 適した試料

骨梁、炭素繊維、多孔質セラミクスなど、繰り返し構造を持つ試料。最もノイズ除去性能と構造保存性のバランスが優れる。

## 2.5 Total Variation Denoising（全変分デノイジング）

### 概要

Rudin, Osher & Fatemi (1992) により提案された変分法ベースのフィルタ（ROFモデル）。画像の「全変分」（勾配のL1ノルム）を最小化しながら元画像への忠実度も保つ最適化問題として定式化される。エッジを非常にシャープに保ちながら均質領域を強力に平滑化する。

### 数式（等方性TV、3D版）

#### 最適化問題

$$\min_u; \frac{\lambda}{2} \|u - f\|^2 + \|\nabla u\|_1$$

#### 等方性TV（Isotropic TV）

$$\|\nabla u\|_1 = \sum_{\mathbf{x}} \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_z u)^2}$$

第1項はデータ忠実度項（フィデリティ項）、第2項は等方性TV正則化項。

### 数値解法：Chambolle-Pock法（Primal-Dual法）

#### 鞍点問題

$$\min_u \max_{\mathbf{p}}; \frac{\lambda}{2} \|u - f\|^2 + \langle \mathbf{p}, \nabla u \rangle - \sum_{\mathbf{x}} \|\mathbf{p}_{\mathbf{x}}\| \leq 1$$

の反復解法。 $\mathbf{p} = (p_x, p_y, p_z)$  は双対変数（勾配場に対応）。

#### 各反復ステップ

##### 1. 双対更新（勾配計算+射影）

$$\mathbf{p}^{n+1} = \Pi_{\|\cdot\| \leq 1} (\mathbf{p}^n + \sum \nabla \bar{u}^n)$$

射影（等方性TV）：

$$\Pi_{\|\cdot\| \leq 1} \mathbf{q} \mapsto \frac{\mathbf{q}}{\max(1, \|\mathbf{q}\|)}$$

##### 2. 主変数更新（収縮）

$$u^{n+1} = \frac{u^n + \tau \operatorname{div}(\mathbf{p}^{n+1}) + \tau \lambda f}{1 + \tau \lambda}$$

##### 3. 外挿（加速）

$$\|\bar{u}^{n+1} - u^{n+1}\| + \theta(u^{n+1} - u^n)$$

収束条件を満たすステップサイズ（3Dの場合）

$$L = \sqrt{12}, \quad \tau = \sigma = \frac{1}{L}, \quad \theta = 1$$

パラメータ

パラメータ	記号	定義	単位	デフォルト
<code>lambda</code>	$\lambda$	正則化パラメータ。データ忠実度と平滑化のバランスを制御	無次元（相対値）	自動（10.0）
<code>iterations</code>	$n_{\max}$	Chambolle-Pock反復回数。多いほど最適解に近づく	回	100

lambda の効果

$\lambda \rightarrow \infty \Rightarrow u \rightarrow f$ （元画像）  
 $\lambda \rightarrow 0 \Rightarrow u \rightarrow \text{定数}$ （完全平坦化）

lambda	効果
1～5	強い平滑化
5～20	標準的デノイジング
20～100	弱い平滑化

iterations の収束

100回でほぼ収束するが、強い平滑化（小さいlambda）では200～300回が必要な場合がある。

適した試料

金属、セラミクス、無機材料など、平坦な領域とシャープなエッジからなる試料。テクスチャが少ない試料で最も効果的。ただし、カートゥーン化（階調のつぶれ）が生じやすいため、テクスチャが重要な生物試料には不向き。

## 2.6 Wavelet Denoising（ウェーブレットデノイジング）

概要

画像をウェーブレット変換して周波数・空間の両域で分析し、ノイズに対応する係数を閾値処理で除去してから逆変換する。BayesShrink（Chang et al., 2000）による適応的閾値選択を採用。

3D Haar ウェーブレット変換

分析フィルタ（Forward DWT）

$$a[i] = \frac{s[2i] + s[2i+1]}{\sqrt{2}} \quad \text{（低域：平均）}$$

$$d[i] = \frac{s[2i] - s[2i+1]}{\sqrt{2}} \quad \text{(高域：差分)}$$

3次元の場合、X→Y→Z方向に順に1D変換を適用する分離可能変換として実装される。

1レベルの3D DWTにより生成されるサブバンド（8個）

サブバンド	内容	処理
LLL	低域×低域×低域（縮小版近似）	次のレベルの入力として使用
LLH	低域×低域×高域（Z方向エッジ）	閾値処理
LHL	低域×高域×低域（Y方向エッジ）	閾値処理
HLL	高域×低域×低域（X方向エッジ）	閾値処理
LHH	低域×高域×高域	閾値処理
HLH	高域×低域×高域	閾値処理
HHL	高域×高域×低域	閾値処理
HHH	高域×高域×高域（点状ノイズ）	閾値処理・ノイズ推定にも使用

Lレベルの分解では LLL サブバンドを再帰的に分解し、合計  $7L + 1$  個のサブバンドが得られる。

閾値処理：ソフト閾値処理（Soft Thresholding）

係数  $c$  に対するソフト閾値処理：

$$\eta_T(c) = \text{sgn}(c) \cdot \max(|c| - T, 0)$$

ハード閾値処理（ $T$  以下をゼロにする）より滑らかな結果が得られる。

閾値決定：BayesShrink

各サブバンドのノイズ標準偏差  $\sigma_n$  をHHHサブバンドから推定し、各サブバンドに最適な閾値を計算する。

ノイズ推定（MAD推定器の近似）

HHHサブバンドの係数の絶対値の平均から推定：

$$\hat{\sigma}_n = \overline{|c_{\text{HHH}}|} \cdot \sqrt{\frac{\pi}{2}}$$

ガウス分布の場合  $E[|X|] = \sigma \sqrt{2/\pi}$  を逆用している。

BayesShrink閾値

各サブバンドの信号標準偏差  $\hat{\sigma}_x$  を推定し：

$$\hat{\sigma}_y^2 = \frac{1}{N} \sum_i c_i^2, \quad \hat{\sigma}_x = \sqrt{\max(\hat{\sigma}_y^2 - \hat{\sigma}_n^2, 0)}$$

$$T_{\text{Bayes}} = \frac{\hat{\sigma}_n^2}{\hat{\sigma}_x}$$



$\hat{\sigma}_x \rightarrow 0$ （サブバンドが純ノイズ）の場合は全係数をゼロにする。

### パラメータ

パラメータ	記号	定義	単位	デフォルト
levels	$L$	DWT分解レベル数	回	3
threshold_scale	$s$	BayesShrink閾値への乗数 $T' = s \cdot T_{\text{Bayes}}$	無次元	1.0

#### levels の効果

levels	対応するノイズスケール	特徴
1	最細ノイズのみ	微細構造を最大限保存
2	細～中スケール	バランス型
3	中スケールまで（推奨）	多くのCT画像に適する
4～5	大スケールも含む	均質領域が多い試料向け

分解可能なレベル数は画像サイズに依存：画像の最小辺を  $\min(W,H,D)$  として  $L_{\max} = \lfloor \log_2(\min(W,H,D)) \rfloor$ 。

#### threshold\_scale の効果

- $s < 1.0$ ：弱い平滑化（ノイズが残る、細かい構造を保存）
- $s = 1.0$ ：BayesShrink理論値（自動最適）
- $s > 1.0$ ：強い平滑化（ノイズが減る、細かい構造も消える）

### 適した試料

CT全般。特に計算速度が重要な大容量データの前処理に適する。BayesShrinkにより自動パラメータ設定が機能するため、パラメータ調整の手間が少ない。

## 2.7 Anisotropic Diffusion（異方性拡散）

### 概要

Perona & Malik (1990) による偏微分方程式（PDE）ベースのフィルタ。熱拡散方程式を拡張し、エッジ強度に応じて拡散係数を変化させることで、均質領域内は平滑化しつつエッジ部は拡散を抑制する。

### 数式（3D版）

#### 拡散方程式

$$\frac{\partial u}{\partial t} = \text{div}(\mathbf{g}(\|\nabla u\|) \cdot \nabla u)$$

#### 3D 離散化（6近傍差分）

$$u^{n+1}(\mathbf{x}) = u^n(\mathbf{x}) + \Delta t \sum_{d \in \{E,W,N,S,U,D\}} g(|\nabla_d u^n|) \cdot \nabla_d u^n$$

$$\nabla_d u = u(\mathbf{x} + \mathbf{e}_d) - u(\mathbf{x})$$
（各方向の前進差分）。

エッジ停止関数（Edge-Stopping Function）

Mode 1（Perona-Malik type 1）：

$$g_1(s) = \frac{1}{1 + (s/K)^2}$$

Mode 2（Perona-Malik type 2）：

$$g_2(s) = \exp!(-\frac{s^2}{K^2})$$

$s = |\nabla u|$  は局所勾配の大きさ。 $K$  はエッジ強度の閾値（エッジ感度パラメータ）。

Mode 1 vs Mode 2 の違い

	Mode 1	Mode 2
関数の形	ゆっくりゼロに近づく	急速にゼロに近づく
エッジ保存	広い勾配範囲で拡散あり	閾値以上で急激に拡散抑制
推奨用途	一般的なCT画像	明瞭なエッジを持つ試料

安定性条件

3次元離散化の安定性には以下が必要：

$$\Delta t < \frac{1}{6} \approx 0.1667$$

デフォルト  $dt = 0.14$  はこれを満たす安全な値。

パラメータ

パラメータ	記号	定義	単位	デフォルト
iterations	$n_{\max}$	時間積分のステップ数（多いほど平滑化が進む）	回	20
K	$K$	エッジ強度の閾値。 $ \nabla u  \leq K$ なら拡散あり、 $ \nabla u  > K$ なら拡散抑制	[画素値/pixel]	自動 ( $\approx 2\sigma_n$ )
dt	$\Delta t$	時間ステップ幅（3Dでは $< 1/6$ が必須）	無次元	0.14
mode	—	エッジ停止関数の選択（1 または 2）	—	1

K の自動設定

$$K_{\text{auto}} = 2 \cdot \hat{\sigma}_n$$

K の調整指針

- $K$  が小さい → エッジが細かく保存される（ノイズとエッジの分離が厳しい）
- $K$  が大きい → 強い平滑化、細かいエッジも失われる
- $K$  の適正值は画像のコントラストに強く依存する

iterations の目安

iterations	効果
5～10	軽い平滑化
15～30	標準的なノイズ除去（推奨）
50～100	強い平滑化（ステアケース効果に注意）
> 100	過剰平滑化、アーティファクト発生の可能性

ステアケース効果

反復が多すぎると、本来滑らかな領域が階段状の輝度パターン（ステアケース）になる。これはPerona-Malikモデルの既知の問題であり、反復回数で制御する。

適した試料

境界が比較的明確な試料全般。均質領域が広く、エッジが明確な無機材料（金属、ガラス、岩石）に特に有効。繊維や骨梁のような複雑な3Dネットワーク構造にも対応できる。

## 2.8 BM4D（Block-Matching 4D Filter）

概要

Maggioni et al. (2013) により提案された、3Dボリューム画像に対するブロックマッチングフィルタ。BM3D（2D画像用）を3次元ボリュームに拡張したもの。類似した3Dブロックを探索・グループ化して4次元配列を構成し、協調フィルタリングを行う。理論的には最高水準のノイズ除去性能を持つ。

本実装では計算コストを考慮した実用的な近似版を採用しており、ブロックマッチングに基づく類似度重み付き平均（協調フィルタリング）を実装している。

数式

類似ブロックの探索

参照ブロック  $B(\mathbf{x})$ （中心  $\mathbf{x}$ 、半径  $r_b$ ）と候補ブロック  $B(\mathbf{y})$ （中心  $\mathbf{y}$ ）の正規化SSD距離：

$$d(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{p} \in \mathcal{B}} \left[ f(\mathbf{x} + \mathbf{p}) - f(\mathbf{y} + \mathbf{p}) \right]^2$$

$$|\mathcal{B}| = (2r_b + 1)^3 \text{ は有効比較画素数。}$$

ブロックの採用基準

$$d(\mathbf{x}, \mathbf{y}) \leq T_{\text{match}}, \quad T_{\text{match}} = 2.7 \sigma^2$$

この閾値は、ノイズのみの場合に  $d \approx 2\sigma^2$ （期待値）となることから設定される。

協調フィルタリング（重み付き平均）

採用した  $M$  個のブロック群から出力を計算。各出力画素  $\mathbf{z}$ （参照ブロック内）に対して：

$$u(\mathbf{z}) = \frac{\sum_{m=1}^M w_m \cdot f(\mathbf{z}_m)}{\sum_{m=1}^M w_m}$$

ブロック類似度重み

$$w_m = \exp\left(-\frac{d(\mathbf{x}, \mathbf{y}_m)}{2\sigma^2}\right)$$

距離がゼロ（自分自身）で重みが最大となり、距離が大きいほど寄与が小さくなる。

集約（Aggregation）

複数の参照ブロックが同じ出力画素に寄与するため、`atomicAdd` による累積加算と最終的な除算で結果を得る。

$$u(\mathbf{z}) = \frac{\text{numerator}(\mathbf{z})}{\text{denominator}(\mathbf{z})}$$

パラメータ

パラメータ	記号	定義	単位	デフォルト
<code>block_radius</code>	$r_b$	参照ブロックの半径。ブロックサイズ $= (2r_b+1)^3$	pixel	2（→ 5×5×5）
<code>search_radius</code>	$r_s$	類似ブロックの探索半径。探索範囲 $= (2r_s+1)^3$	pixel	3（→ 7×7×7）
<code>sigma</code>	$\sigma$	ノイズ標準偏差（採用閾値と重みに使用）	[画素値と同 単位]	自動

計算量

1ボクセルあたり  $(2r_s+1)^3 \times (2r_b+1)^3$  回の差分計算が必要。デフォルトでは  $343 \times 125 = 42875$  回。NLMより約4.6倍重い。

最大グループ数（`MAX_MATCHED_BLOCKS`）

1参照ブロックあたり最大16個の類似ブロックを保持（ソート済みリスト）。これにより、近傍全探索を行いつつも計算量を制限している。

パラメータ調整指針

パラメータ	小さい	大きい
<code>block_radius</code>	局所的な類似度評価	広域の構造パターンを比較
<code>search_radius</code>	計算速度優先	より多くの類似ブロックを参照
<code>sigma</code>	少ないブロックを採用	多くのブロックを採用（ぼやけ）

sigmaの設定と効果

- sigma が小さすぎる → 類似ブロックが見つからず、ノイズが残る
- sigma が大きすぎる → 非類似ブロックも採用され、ぼやける
- 真のノイズ標準偏差  $\sigma_n$  に近い値が最適

適した試料

高ノイズ環境でのCT画像全般。NLMと比べてブロック（局所構造）の類似度を評価するため、単純な繰り返しパターンより複雑な3D構造に対して優れた性能を発揮する可能性がある。ただし計算コストが高いため、まずパラメータを小さめに設定してテストすることを推奨する。

第3章 フィルタの選択ガイド

3.1 試料タイプ別推奨フィルタ

試料タイプ	第1推奨	第2推奨	注意点
骨・骨梁構造	NLM	Bilateral	繰り返し構造に強いNLMが最適
炭素繊維複合材	NLM	Aniso. Diff.	繊維方向のエッジ保存が重要
金属・合金	TV	Bilateral	シャープなエッジが重要
セラミクス・ガラス	TV	Aniso. Diff.	平坦領域が多い
多孔質材料	NLM	BM4D	細孔境界の正確な保存
生物軟組織	Bilateral	Wavelet	コントラストが低い
岩石・地質試料	Aniso. Diff.	TV	様々なスケールのエッジ
均質な高分子	Gaussian	Wavelet	単純な平滑化で十分
スパイクノイズ含む	Median	—	外れ値除去に特化

3.2 ノイズレベル別推奨フィルタ

S/N比	状況	推奨フィルタ
高 (S/N > 30)	軽いノイズ	Gaussian, Wavelet
中 (S/N 10~30)	標準的なCT	Bilateral, NLM, Aniso. Diff.
低 (S/N < 10)	ノイズが多い	NLM, BM4D, TV

3.3 計算速度の比較（相対値）

同一パラメータ規模での相対的な処理速度の目安（GPU版）：

フィルタ	相対速度	備考
------	------	----

フィルタ	相対速度	備考
Gaussian	◎◎◎ 最速	畳み込みのみ
Median	◎◎	ソート処理あり
Wavelet	◎◎	FFTより軽量
Aniso. Diff.	◎	反復回数依存
TV	○	反復最適化
Bilateral	○	カーネルサイズ依存
NLM	△	探索範囲依存（最重要）
BM4D	△～×	search_radius に敏感

3.4 パラメータ調整のワークフロー

1. まず 2D プラグイン（プレビュー付き）でパラメータを探索  
↓（代表的なスライス 1 枚で確認）

2. デフォルトパラメータで実行してみる  
↓（結果を確認）

3. ぼやけすぎ →  $h/\sigma/K$  を小さく、 $\lambda$  を大きく  
ノイズが残る →  $h/\sigma/K$  を大きく、 $\lambda$  を小さく  
↓（最適値が見つかったら）

4. 3D 外部プログラム（GPU版）で全スタックに適用  
↓

5. cmd-hst.log でパラメータを記録・確認

第4章 チャンク処理とオーバーラップ

4.1 チャンク処理の必要性

大容量CTデータ（数GB～数十GB）はGPUメモリに一度に読み込めない。本ツールはZ方向にデータをチャンク（塊）に分割して順次処理する。

4.2 オーバーラップの設計

チャンク境界でのアーティファクトを防ぐため、各チャンクは隣接チャンクとオーバーラップして読み込まれる。保存するのは有効領域のみ。

各フィルタのオーバーラップサイズ

フィルタ	オーバーラップ $\delta$	計算式
Bilateral	$\max(r_k, \lceil 3\sigma_s \rceil) + 1$	$r_k$ : カーネル半径
Gaussian	$\max(r_k, \lceil 3\sigma \rceil) + 1$	3 $\sigma$ ルール

フィルタ	オーバーラップ $\delta$	計算式
Median	$r_k + 1$	$r_k = \lfloor k/2 \rfloor$
NLM	$r_s + r_p + 1$	探索+パッチ半径
TV	3 (固定)	境界差分の影響範囲
Wavelet	$2^L + 1$	$L$ : 分解レベル数
Aniso. Diff.	3 (固定)	差分の伝播範囲
BM4D	$r_s + r_b + 1$	探索+ブロック半径

### 4.3 コマンドログ (cmd-hst.log)

実行完了後、実行コマンドとパラメータがカレントディレクトリの `cmd-hst.log` に追記される。ImageJプラグインから呼び出した場合は元画像ディレクトリの1階層上に記録される。

```
tif_nlm_g /data/ct/input /data/ct/output 1 3 200.0
tif_tvd_g /data/ct/input /data/ct/output 10.0 100
tif_blf_g /data/ct/input /data/ct/output 5 2.0 21845.0
```

## 第5章 コマンドリファレンス

### 5.1 コマンド一覧

```
# Bilateral Filter
tif_blf <in_dir> <out_dir> [kernel_size] [spatial_sigma] [intensity_sigma]
tif_blf_g <in_dir> <out_dir> [kernel_size] [spatial_sigma] [intensity_sigma]

# Gaussian Filter
tif_gsf <in_dir> <out_dir> [sigma]
tif_gsf_g <in_dir> <out_dir> [sigma]

# Median Filter
tif_mdf <in_dir> <out_dir> [kernel_size]
tif_mdf_g <in_dir> <out_dir> [kernel_size]

# Non-Local Means
tif_nlm_g <in_dir> <out_dir> [patch_radius] [search_radius] [h]

# Total Variation Denoising
tif_tvd_g <in_dir> <out_dir> [lambda] [iterations]

# Wavelet Denoising
tif_wvd_g <in_dir> <out_dir> [levels] [threshold_scale]
```

```
# Anisotropic Diffusion
tif_adf_g <in_dir> <out_dir> [iterations] [K] [dt] [mode]

# BM4D
tif_bm4d_g <in_dir> <out_dir> [block_radius] [search_radius] [sigma]
```

5.2 パラメータデフォルト値一覧

コマンド	パラメータ1	パラメータ2	パラメータ3	パラメータ4
tif_blf(_g)	kernel=5	$\sigma_s=2.0$	$\sigma_r$ =自動	—
tif_gsf(_g)	$\sigma=2.0$	—	—	—
tif_mdf(_g)	kernel=3	—	—	—
tif_nlm_g	r_p=1	r_s=3	h=自動	—
tif_tvd_g	$\lambda$ =自動	iter=100	—	—
tif_wvd_g	L=3	scale=1.0	—	—
tif_adf_g	iter=20	K=自動	dt=0.14	mode=1
tif_bm4d_g	r_b=2	r_s=3	$\sigma$ =自動	—

5.3 対応ファイル形式

ビット深度	形式	備考
8-bit	符号なし整数	値域 0~255
16-bit	符号なし整数	値域 0~65535
32-bit	IEEE 754 単精度浮動小数点	CT再構成像の標準形式

入力・出力ともに個別TIFFファイル形式（1スライス = 1ファイル）。ファイル名はアルファベット順にソートされてZ方向スタックとして読み込まれる。

参考文献

1. Tomasi, C. & Manduchi, R. (1998). Bilateral filtering for gray and color images. *ICCV 1998*, pp. 839–846.
2. Buades, A., Coll, B. & Morel, J.-M. (2005). A non-local algorithm for image denoising. *CVPR 2005*, Vol. 2, pp. 60–65.
3. Rudin, L., Osher, S. & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1–4), pp. 259–268.
4. Chambolle, A. & Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), pp. 120–145.



5. Chang, S. G., Yu, B. & Vetterli, M. (2000). Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9), pp. 1532–1546.
6. Perona, P. & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), pp. 629–639.
7. Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2007). Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8), pp. 2080–2095.
8. Maggioni, M., Katkovnik, V., Egiazarian, K. & Foi, A. (2013). Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1), pp. 119–133.