

[特別講演] 多チャンネルバイラテラルフィルタの高速化

福嶋 慶繁^{1,a)} 山下 頌太^{1,b)}

概要: バイラテラルフィルタは、代表的なエッジ保存フィルタである。その重みは畳み込みカーネルの相対位置や参照画素値に応じて変化し、これは linear time-variant(LTV) なフィルタとなるため、直接的には FFT や再帰型フィルタで効率的に表現することは難しい。近年、バイラテラルフィルタを分解し、linear time-invariant(LTI) なフィルタの重ね合わせで表現できることが示され、カーネルサイズに非依存な高速実装が可能となっている。しかし、この表現では、チャンネル数に応じて指数的に分解すべきフィルタ回数が増加するため、3チャンネルといったカラー情報にするだけでも計算時間が増大する。一方で、分解するフィルタをセパラブルな LTV とすることで、必要な分解数を抑えられることが示されている。ただし、一回のフィルタは LTI のフィルタよりも計算時間が大きい。本稿では、一見簡単に見える、高速なバイラテラルフィルタのカラー画像処理への拡張においての問題を明らかにし、どのようなアプローチがあるのかを示す。

キーワード: バイラテラルフィルタ, 近似高速化, エッジ保存平滑化フィルタ, ガウシアンフィルタ

1. はじめに

バイラテラルフィルタ (Bilateral Filter: BF) [1] は、画像の平滑化とエッジ保持を両立したフィルタであり、その優れたフィルタ特性から、ノイズ除去 [2]、ハイダイナミックレンジ画像 [3]、アップサンプリング [4]、霞除去 [5]、テクスチャ強調 [6]、スタイライゼーション [7] といった、様々な画像処理アプリケーションに応用されている。また、ジョイントバイラテラルフィルタ [8]、[9] と呼ばれる、フィルタ対象画像に加えて参照画像を用い、参照画像を用いてバイラテラルフィルタの重みを決定する方法は、参照画像のエッジ情報を活用しながら畳み込みが可能のため、コストや結果を画像の輪郭に応じて平滑化可能になり、さらに広い範囲に応用されている。例えば、ステレオマッチング [10] やオプティカルフロー推定 [11] といった推定問題から、推定結果のマップ (例えばデプスマップや動きベクトル) を参照画像の輪郭に合わせるように高精度化 [12] することといったことに使われている。

このフィルタは、注目・参照画素間の空間距離と輝度差に基づいた適応的な重みを用いて畳み込みを行う。しかし、ナイーブな畳み込み処理には各重みに指数関数の計算が必要であるため、その計算コストは高い。そこで、様々

な高速化手法が提案されている。

この高速化にはいくつかの方法があり、現在の主流となっている高速化アルゴリズムでは、BF を複数のガウシアンフィルタ処理に分解することで、高速化する方法 [3]、[13]、[14]、[15]、[16]、[17]、[18]、[19] や、セパラブルフィルタ [20]、[21]、[22]、[23] として近似する方法、カーネルや画素を間引くことで高速化する方法 [24]、[25] がある。

まず、バイラテラルフィルタを複数のガウシアンフィルタに分解する手法では、バイラテラルフィルタよりもガウシアンフィルタが非常に高速に計算可能なことから、複数回実行したとしてもトータルで高速になる。例えば、FFT やサブサンプリングを用いた高速化を用いる方法 [3]、[15]、[16] では通常のガウシアンフィルタよりも高速に動作可能であり、加えて、再帰フィルタで実現したものは、定数時間で高効率に計算可能である [26]、[27]、[28]。もし、定数時間アルゴリズムを用いれば、バイラテラルフィルタが複数回のガウシアンフィルタの総和に変換されるため計算オーダーが $O(1)$ となる。これらの手法をグレースケール画像に対しては適用すれば、概ね 10 回未満のガウシアンフィルタに分解可能であり、効率的に動作する。しかしながら、分解されるガウシアンフィルタの個数は色のチャンネル数の累乗で増加するため次元の呪いを受ける。例えば、1チャンネルで 8 回のガウシアンフィルタで十分だったものが $8^3 = 512$ 回ものフィルタが必要となる。カラー画像処理の効率化を目的とした定数時間アルゴリズム [15]、[16] も提案されて

¹ 名古屋工業大学
Nagoya Institute of Technology
a) fukushima@nitech.ac.jp
b) s.yamashita.171@stn.nitech.ac.jp

いるが、これらは処理に用いるデータの構造が複雑であり、また、データのダウンサンプリングを前提としているため、高精度な近似はむづかしい。

一方で、セパラルフィルタによる高速化は、バイラテラルフィルタを $O(r)$ (r はカーネル半径) のオーダーに落とすことができる。このセパラルバイラテラルフィルタ (Separable Bilateral Filter: SBF) [20], [21], [22] は、バイラテラルフィルタを複数のセパラルフィルタに分解するものであり、計算オーダーは $O(r)$ ながら、カーネルサイズが中規模程度までであれば十分高速に動作する。しかし、この手法も同様に次元の呪いをうける。ただし、ガウシアンフィルタに分解する場合と比べて必要な畳み込み回数が非常に少なく、1チャンネルで2, 3回のフィルタ、つまり $2^3 = 8$ 回程度の畳み込みでよく、カラーの場合でも十分対応可能である。

ほかに、画素やカーネルを間引く方法では、単純に畳み込みのカーネル数や画素数を間引くため減らした分だけ演算量が削減される。この方法は次元の呪いもなく、カラー画像の場合は有効である。

本稿では、これらのバイラテラルフィルタの近似高速化手法について、特にカラー画像におけるまとめる。

2. バイラテラルフィルタの近似高速化

2.1 バイラテラルフィルタの定義

f を入力画像、 \tilde{f} を出力画像とするとバイラテラルフィルタは以下の定義となる。

$$\tilde{f}_i = \frac{\sum_{j \in \mathcal{N}_i} w_{i,j} f_j}{\sum_{j \in \mathcal{N}_i} w_{i,j}}, \quad (1)$$

ここで、 i と j は画素位置であり、 \mathcal{N}_i は画素 i の周辺画素集合である。 w はフィルタ重みであり以下で表される。

$$w_{i,j} = \exp\left(\frac{-\|i-j\|_2^2}{2\sigma_s^2}\right) \exp\left(\frac{-\|f_i - f_j\|_2^2}{2\sigma_r^2}\right), \quad (2)$$

ここで、 $\|\cdot\|_2$ は l_2 -ノルムであり、 σ_s と σ_r は空間、レンジ方向の平滑化パラメータである。この実装では、カーネル半径が大きくなるほど $O(r^2)$ で計算量が増加する。

2.2 定数時間バイラテラルフィルタ

グレイスケール画像の場合のバイラテラルフィルタのガウシアンフィルタへの分解は以下のように定義される。

$$\tilde{f} = \frac{\sum_{n \in \mathcal{N}} g * w_n f}{\sum_{n \in \mathcal{N}} g * w_n} \quad (3)$$

ここで、 g はガウシアンカーネル、 w_n は重み画像を表している。また、 N は分解数、 n はその n 番目の要素を表している。つまり、バイラテラルフィルタは、画像とある重みのアダマール積をガウシアンフィルタで畳み込んだものの

総和へと分解可能である。このガウシアンフィルタの計算は高速化可能であり、代表的なものとしてはFFTでの高速化がある。さらに、再帰フィルタによる実装を用いれば、より効率的に計算可能であり、カーネル半径によらない任意のサイズのガウシアンフィルタを定数時間で計算可能である。このガウシアンフィルタの畳み込みの回数を削減することで高速化可能であるが、その計算は近似となるため、より高い近似精度を持つ畳み込みが求められている。

この分解の方法には、さまざまな手法が提案されており重みの定義の仕方によって分類できる。早期の方法は輝度値を離散的にサンプルする方法が提案されていた。それをより高精度にするために、多項式や三角関数、その最適展開などが提案されている。離散的にサンプルする手法もハット関数による畳み込みとみなせることから、つまり、任意の関数で分解することでその効率が異なることが言える。現在では、三角関数の最適展開で行う手法が最も少ない分解数で高精度な畳み込みを実現可能である。

次に、この分解をカラー画像へと展開する。カラーの重みは $\exp(\frac{B^2+G^2+B^2}{2\sigma^2})$ であり、これは指数法則により3つの指数関数 $\exp(R^2/2\sigma^2) \exp(G^2/2\sigma^2) \exp(B^2/2\sigma^2)$ に分解可能である。つまり、グレイスケールと同様の重みが3回出てきており、分解したものを分解してさらに分解する必要がある。グレイスケールの場合、分解されるガウシアンフィルタの数は8個程度であるが、もしカラーであればその3乗となり次元の呪いが避けられない。

カラー画像に対応するために、グレイスケールの処理を各チャンネルへ3回行うアプローチもあるが、その場合は重みが異なり、出力値が全く異なる。特に、画像の色合いの関係上で特定のチャンネルだけ周囲とのエッジが曖昧な場合は、そのチャンネルだけボケがつよくなり、RGBすべての違いを考慮したフィルタとは出力が大きく変わってくる。

2.3 セパラルバイラテラルフィルタ

セパラル実装は、2次元畳み込みフィルタを1次元の垂直・水平畳み込みフィルタという2つのフィルタに分解する古典的なアプローチの一つである。その結果、計算オーダーは $O(r)$ へと削減することができる。カーネルが可分性を持つフィルタでは、この手法によって近似なしに高速化が可能である。例えば、ガウシアンフィルタや、ラプラシアンフィルタなどが相当する。しかし、バイラテラルフィルタはレンジの重みが可分性を持たないため、出力が近似値となり、近似精度についての議論が必要である。近似値となる場合のバイラテラルフィルタのセパラル実装は、フィルタリング方向の順序によって出力が異なるが、ここでは、1パス目の処理が垂直方向、2パス目の処理が水平方向であると仮定して説明する。

セパラルバイラテラルフィルタは、バイラテラルフィルタのセパラル実装であり、最初に Pham らによって提

案された [20]. その実装では、まず、入力画像に対して垂直方向にバイラテラルフィルタを適用し、中間画像を生成する。その後、中間画像に対して水平方向にバイラテラルフィルタを適用し、最終的な出力を得る。なお、水平方向の BF の重みは中間画像の輝度値を参照して計算される。

Pham らの実装方法は、高速かつ省メモリだが、近似精度が低いことが問題として挙げられる。この問題に対して、水平方向フィルタに用いる重みを、中間画像からではなく、入力画像を参照して計算することで精度を向上させる手法が提案されている [22]. 中間画像と比較して、入力画像は正しいテクスチャ情報を持っていることから、この手法を適用することで、近似精度が有意に向上する。また、これに加えて文献 [22] では、SBF の出力に平滑化効果が過剰にかかりやすいという傾向に着目し、平滑化効果を抑えるためのパラメータを水平方向フィルタの計算式に追加することで近似誤差を減らしている。この手法はスイッチングデュアルカーネル (Switching Dual Kernels: SDK) と呼ばれ、初期の SBF に比べると、上記の手法により大幅な高精度化がされている。

このフィルタを一般化し、定数時間バイラテラルフィルタのように複数回のセパラブルバイラテラルフィルタに分解することを考えたものが前報告である [23]. この場合、以下のように定式化可能である。

$$\tilde{f} \approx \sum_{k \in K} \frac{sbf_h * sbf_v * w_k f}{sbf_h * sbf_v * w_k} \quad (4)$$

ここで、 sbf_h, sbf_v は、横縦のセパラブルバイラテラルフィルタカーネル、 $*$ は畳み込み、 w_k は重みを表しており、定数時間バイラテラルフィルタのガウシアンフィルタに代わってセパラブルバイラテラルフィルタを畳み込む形になっている。

この定式化をカラー画像に適用する場合、複数の展開の仕方がある。定数時間バイラテラルフィルタのように複数の指数関数として展開する場合、指数関数を 1 つのままにテイラー展開する場合、カラー画像の重みを平均値で代用することより、チャンネル削減する方法である。 [23]. また 0 次の場合は次元展開をする必要がなく、この場合は早期の手法である文献 [20] や、その改善版である SBF-SDK [22] をこの定式化を単純化したもので表すことができる。

2.4 乱択バイラテラルフィルタ

乱択バイラテラルフィルタでは、ナイーブなバイラテラルフィルタの畳み込みカーネルをランダムに間引いて計算することで高速化する。

$$\tilde{f}_i \approx \frac{\sum_{j \in \mathcal{N}_i} w_{i,j} I_j f_j}{\sum_{j \in \mathcal{N}_i} w_{i,j} I_j}, \quad (5)$$

ここで \mathcal{N}_i は、画素 i における周辺画素集合であり、乱択フィルタの場合は、画素ごとに変化する。つまり通常の畳

み込みに対してサンプルするか否かの確率 p で変動させる。

$$P[I_j] = \begin{cases} p_j & (I_j = 1) \\ 1 - p_j & (I_j = 0), \end{cases} \quad (6)$$

乱択バイラテラルフィルタでは間引いた分だけフィルタが高速化される。しかしカーネルを間引く場合、単純に等間隔に間引くとエイリアシングを避けられない。そこで乱択バイラテラルフィルタでは、サンプル位置をランダムにすることでエイリアシングを抑制する。

サンプルするとき重要な情報として、バイラテラルフィルタの重みは、注目画素から近く類似した色であるほど重みが高いということがある。言い換えれば、低い重みの場所は一つサンプルを加えても最終的な出力への影響力が低い。つまり、影響力の高い画素だけ選択して畳み込みば高精度に近似可能となる。しかしながら、カーネルごとに最適な重みを探索しては、その探索自体のコストが大きい。そのため、統計的に最適なサンプリングパターンを次に作成することで乱択バイラテラルフィルタを高速に動作させる。このパターン生成の詳細は、文献 [25] を参照されたい。

この手法は、直接的な畳み込みで実現されるため、色情報の次元の呪いを受けない。しかしながら、カーネル半径が大きくなると必要な画素数が増加するというナイーブな実装の問題はそのまま引き継がれている。そのため、カーネル半径が大きくなると定数時間アルゴリズムよりも効率が低下しやすい。また、ランダムにサンプルするため、画像の平滑化度合いに若干ざらつきが存在する。そのため、画像周辺からの変動も考慮する画質評価指標である SSIM [29] を用いるとその近似精度は低下しやすい。

3. 比較実験

様々なバイラテラルフィルタの近似高速化を、計算効率と近似精度においてを比較した。近似精度は Kodak 社より提供されているカラー画像 24 枚における平均精度で評価し、精度の客観評価尺度として、PSNR と SSIM を用いた。比較手法として、定数時間バイラテラルフィルタとしてコンプレッシブバイラテラルフィルタ (Compressive bilateral filter: BCF)、セパラブルバイラテラルフィルタとして、早期の手法である文献 [20](Pham-SBF) や、その改善版である SBF-SDK [22] とその高精度版 (SBF-SDK-QF)、テイラー展開によるカラーの 3 階展開したもの、まとめて展開したもの、色削減したもののセパラブルバイラテラルフィルタ (SBF-ME,IE,CR) を用いた。また、乱択バイラテラルフィルタ (Randomized Bilateral Filter: RBF) [24] も比較した。実験で扱う各手法は C++ で実装されており、SIMD 命令、OpenMP を用いてベクトル化、並列化されている。また、CPU として Intel Core i7-6700 3.40GHz を

用いた。なお、畳み込みカーネルの半径は、バイラテラルフィルタの空間平滑化パラメータ σ_s の3倍 $3r$ とした。

まず図1, 2に、コンプレッシブバイラテラルフィルタとセパラブルバイラテラルフィルタの計算時間とPSNR, SSIMによる近似精度の比較結果を示す。1次以上のセパラブルバイラテラルフィルタ(k-SBF)とコンプレッシブバイラテラルフィルタは次数を増やすことで精度と速度のトレードオフを取っている。結果を見ると、次元の呪いの効果が弱いSBFが高効率であることがわかる。特にSDK-SBFやSDK-CRの効果が大きい。なお、計算時間の横軸は対数軸であり、定数時間バイラテラルフィルタに比べてセパラブルバイラテラルフィルタの効果が非常に大きいことがわかる。

図3, 4に乱択バイラテラルフィルタとセパラブルバイラテラルフィルタの比較を示す。なお、横軸は対数軸から通常の軸に戻している。PSNRで比較すると、SBFはカーネルサイズが大きいくほど近似の効率が落ちやすいため、 σ_s が大きくなるほどRBFに比べて処理効率が悪くなる。しかし σ_s が小さい時は、RBFよりも高効率な処理ができています。また、SSIMでみると、SBFのほうがより広い σ_s の範囲で効率的に動作している。これは、SSIMが周辺画素からの変動、ざらつきも評価対象としているためであり、単純に単一の画素値だけで評価しない場合にはRBFよりもより滑らかに平滑化できるSBFのほうが効果があることを示している。なお、実験では、乱択バイラテラルフィルタでは間引き率を変えることでトレードオフを調整している。また、1次以上の近似では乱択バイラテラルフィルタよりも効率が悪かったため、0次と1次の近似のみを示している。

4. まとめ

本稿では、バイラテラルフィルタを3種類の近似高速化手法(定数時間ガウシアンフィルタへの分解, セパラブルバイラテラルフィルタへの分解, 乱択化によるカーネル間引き)を比較した。特にカラー画像の場合は、 $O(1)$ の定数時間フィルタによる分解よりも、 $O(r)$ や (r^2) となるセパラブルバイラテラルフィルタや乱択バイラテラルフィルタが効果的であることを示した。更なる高速化のためには、ドメイン固有言語であるHalide [30], [31]やDarkroom [32]といったものを用いて、複雑化する画像処理パイプラインを最適化するなど、アルゴリズム以外にもハードウェアを考慮したプログラミングなどを行い、場面に応じたアルゴリズムの選択をする必要があるだろう。

謝辞 本研究は科研費JP17H01764の助成を受けた。

参考文献

[1] Tomasi, C. and Manduchi, R.: Bilateral Filtering for Gray and Color Images, *IEEE International Conference*

on Computer Vision (ICCV), pp. 839–846 (1998).

[2] Buades, A., Coll, B. and Morel, J. M.: A Non-Local Algorithm for Image Denoising, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 60–65 (2005).

[3] Durand, F. and Dorsey, J.: Fast Bilateral Filtering for the Display of High-Dynamic-Range Images, *ACM Trans. on Graphics*, Vol. 21, No. 3, pp. 257–266 (2002).

[4] Kopf, J., Cohen, M. F., Lischinski, D. and Uyttendaele, M.: Joint Bilateral Upsampling, *ACM Trans. on Graphics*, Vol. 26, No. 3, pp. 96:1–96:5 (2007).

[5] He, K., Sun, J. and Tang, X.: Single Image Haze Removal Using Dark Channel Prior, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 12, pp. 2341–2353 (2011).

[6] Bae, S., Paris, S. and Durand, F.: Two-scale Tone Management for Photographic Look, *ACM Trans. on Graphics*, Vol. 25, No. 3, pp. 637–645 (2006).

[7] DeCarlo, D. and Santella, A.: Stylization and Abstraction of Photographs, *ACM Trans. on Graphics*, Vol. 21, No. 3, pp. 769–776 (2002).

[8] Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M. and Toyama, K.: Digital Photography with Flash and No-flash Image Pairs, *ACM Trans. on Graphics*, Vol. 23, No. 3, pp. 664–672 (2004).

[9] Eisemann, E. and Durand, F.: Flash Photography Enhancement via Intrinsic Relighting, *ACM Trans. on Graphics*, Vol. 23, No. 3, pp. 673–678 (2004).

[10] Scharstein, D. and Szeliski, R.: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, *International Journal of Computer Vision*, Vol. 47, No. 1, pp. 7–42 (2002).

[11] Hosni, A., Rhemann, C., Bleyer, M., Rother, C. and Gelautz, M.: Fast Cost-Volume Filtering for Visual Correspondence and Beyond, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 2, pp. 504–511 (2013).

[12] Matsuo, T., Fukushima, N. and Ishibashi, Y.: Weighted Joint Bilateral Filter with Slope Depth Compensation Filter for Depth Map Refinement, *International Conference on Computer Vision Theory and Applications (VISAPP)*, pp. 300–309 (2013).

[13] Porikli, F.: Constant Time $O(1)$ Bilateral Filtering, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8 (2008).

[14] Yang, Q., Tan, K. H. and Ahuja, N.: Real-Time $O(1)$ Bilateral Filtering, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 557–564 (2009).

[15] Adams, A., Gelfand, N., Dolson, J. and Levoy, M.: Gaussian KD-trees for Fast High-dimensional Filtering, *ACM Trans. on Graphics*, Vol. 28, No. 3, pp. 21:1–21:12 (2009).

[16] Adams, A., Baek, J. and Davis, M. A.: Fast High-Dimensional Filtering Using the Permutohedral Lattice, *Computer Graphics Forum*, Vol. 29, No. 2, pp. 753–762 (2010).

[17] Chaudhury, K.: Constant-Time Filtering Using Shiftable Kernels, *IEEE Signal Processing Letters*, Vol. 18, No. 11, pp. 651–654 (2011).

[18] Sugimoto, K. and Kamata, S.-I.: Compressive Bilateral Filtering, *IEEE Trans. on Image Processing*, Vol. 24, No. 11, pp. 3357–3369 (2015).

[19] Sugimoto, K., Fukushima, N. and Kamata, S.: Fast bilateral filter for multichannel images via soft-assignment coding, *Asia-Pacific Signal and Information Processing*

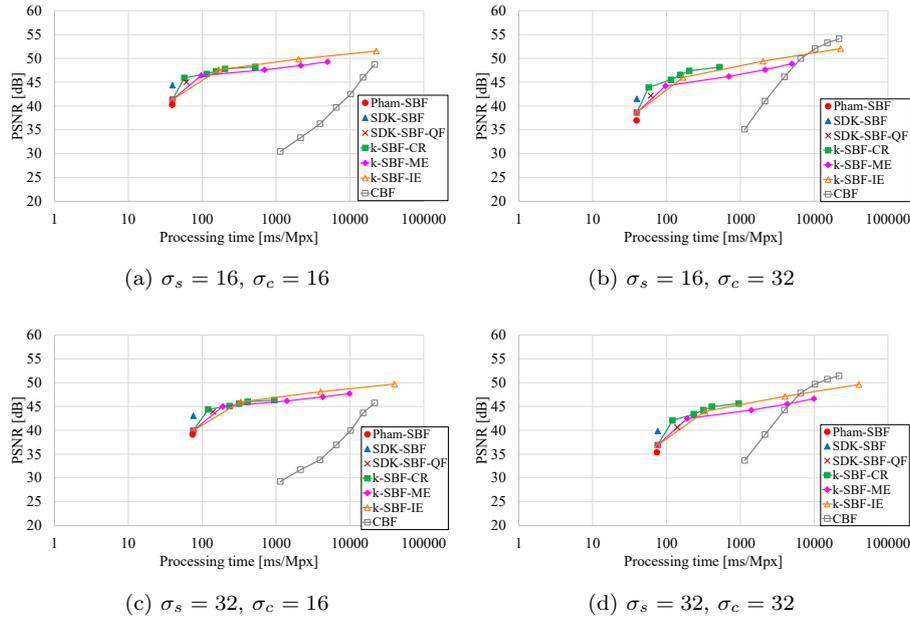


図 1: コンプレッシブバイラテラルフィルタとセパラルバイラテラルフィルタの比較 (PSNR).

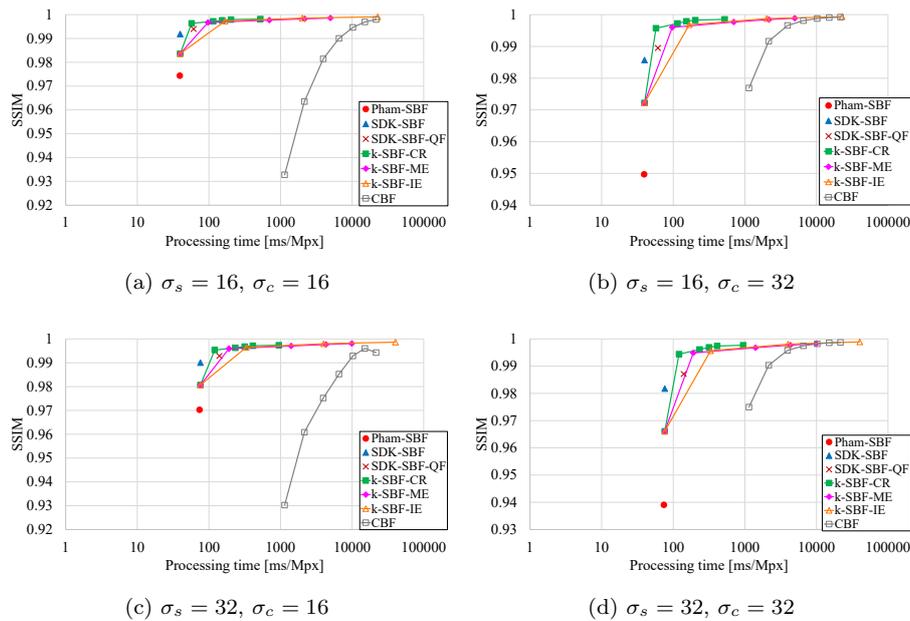


図 2: コンプレッシブバイラテラルフィルタとセパラルバイラテラルフィルタの比較 (SSIM).

Association Annual Summit and Conference (APSIPA) (2016).

- [20] Pham, T. Q. and Vliet, L. J. V.: Separable Bilateral Filtering for Fast Video Preprocessing, *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–4 (2005).
- [21] Kim, Y. S., L., H., Choi, O., Lee, K., Kim, J. and Kim, J.: Separable bilateral nonlocal means, *IEEE International Conference on Image Processing (ICIP)*, pp. 1513–1516 (2011).
- [22] Fukushima, N., Fujita, S. and Ishibashi, Y.: Switching Dual Kernels for Separable Edge-Preserving Filtering, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1588–1592 (2015).

- [23] 山下頌太 and 福嶋慶繁: 加重係数の多項式近似によるセパラルバイラテラルフィルタリングの高精度化, 情報処理学会研究報告オーディオビジュアル複合情報処理 (AVM) (2017).
- [24] Banterle, F., Corsini, M., Cignoni, P. and Scopigno, R.: A Low-Memory, Straightforward and Fast Bilateral Filter Through Subsampling in Spatial Domain, *Computer Graphics Forum*, Vol. 31, No. 1, pp. 19–32 (2012).
- [25] 藤田秀, 木村誠 and 福嶋慶繁: 乱択化バイラテラルフィルタによる高速エッジ保持平滑化, 情報処理学会研究報告コンピュータビジョンとイメージメディア研究会 (CVIM) (2016).
- [26] Sugimoto, K. and Kamata, S.: Fast Gaussian Filter with Second-Order Shift Property of DCT-5, *IEEE Interna-*

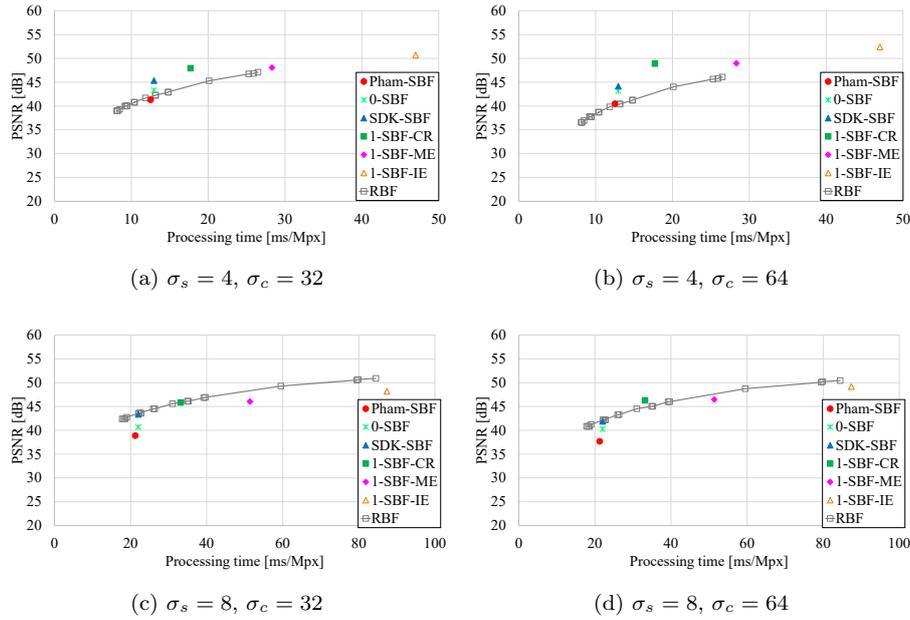


図 3: 乱択バイラテラルフィルタとセパラルバイラテラルフィルタの比較 (PSNR).

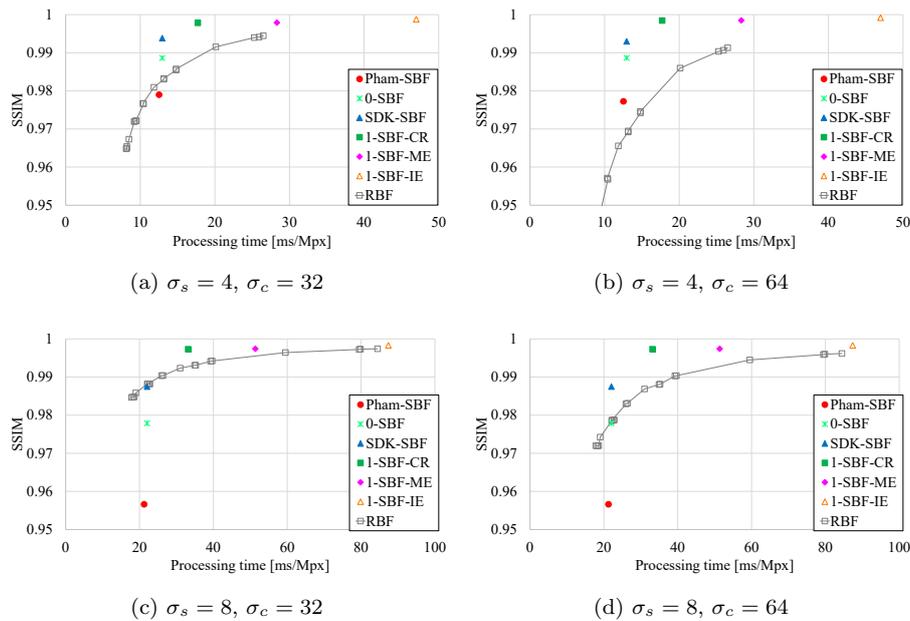


図 4: 乱択バイラテラルフィルタとセパラルバイラテラルフィルタの比較 (SSIM).

tional Conference on Image Processing (ICIP), pp. 514–518 (2013).

- [27] Sugimoto, K., Kyochi, S. and Kamata, S.: Universal approach for DCT-based constant-time Gaussian filter with moment preservation, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018).
- [28] Kober, V.: Fast algorithms for the computation of sliding discrete sinusoidal transforms, *IEEE Transactions on Signal Processing*, Vol. 52, No. 6, pp. 1704–1710 (2004).
- [29] Wang, Z., Bovik, A., Sheikh, H. and Simoncelli, E.: Image Quality Assessment: from Error Visibility to Structural Similarity, *IEEE Trans. on Image Processing*, Vol. 13, No. 4, pp. 600–612 (2004).

- [30] Ragan-Kelley, J., Adams, A., Paris, S., Levoy, M., Amarasinghe, S. and Durand, F.: Decoupling Algorithms from Schedules for Easy Optimization of Image Processing Pipelines, *ACM Transactions on Graphics*, Vol. 31, No. 4 (2012).
- [31] Ragan-Kelley, J., Barnes, C., Adams, A., Paris, S., Durand, F. and Amarasinghe, S.: Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines, *ACM SIGPLAN Notices*, Vol. 48, No. 6, pp. 519–530 (2013).
- [32] Hegarty, J., Brunhaver, J., DeVito, Z., Ragan-Kelley, J., Cohen, N., Bell, S., Vasilyev, A., Horowitz, M. and Hanrahan, P.: Darkroom: compiling high-level image processing code into hardware pipelines., *ACM Trans-*

actions on Graphics, Vol. 33, No. 4, pp. 144-1 (2014).