

符号付変化量最大化反復バイラテラルフィルタによる塗り絵の下絵線画の生成 Generating Line Art for Coloring by Signed Variation Maximizing Iterative Bilateral Filter

井上 光平[†]
Kohei Inoue

浦浜 喜一[†]
Kiichi Urahama

あらまし

反復バイラテラルフィルタ (IBF) における反復回数
決定の問題を解決するための方法として、画素値の符号
付変化量を最大化する IBF (S-IBF) を提案し、塗り絵の
下絵線画の生成に応用する。下絵線画の生成手順を、画
像の簡略化とエッジの抽出の2段階に分けて考え、S-IBF
で画像を簡略化した後、アンシャープマスキングの考え
方に基づくエッジ抽出を行い、エッジの強度を線の太さ
に反映させた下絵線画を生成する。既存のソフトウェア
で生成した下絵と比較して、線の途切れが少ない下絵線
画が得られることを実験で確認する。

1. まえがき

近年、エッジ保存平滑化フィルタの一つであるバイラ
テラルフィルタ (bilateral filter [1]: BF) が画像処理や
コンピュータグラフィックスなどの幅広い分野で利用さ
れている。例えば、Durand ら [2] はダイナミックレン
ジ圧縮を行い、Eisemann ら [3] はフラッシュ写真強調
を行い、Fleishman ら [4] はメッシュノイズ除去を行っ
た。BF にはエッジを保存するだけでなく、主要なエッ
ジを強調する性質もあることが知られている [5]。この
エッジ強調性を更に引き出すために、BF を繰り返しか
ける反復 BF (iterative BF: IBF) のアイデアも、既に
Tomasi ら [1] によって提案されている。しかし、その反
復回数を合理的に決定する方法は未だ開発されておらず、
適用例に応じて、経験的に良い結果が得られる値に設定
されることが多い [6]。そこで本論文では、IBF におけ
る反復回数の決定の問題を解決するための方法として、画
素値の符号付変化量を最大化する IBF (signed variation
maximizing IBF: S-IBF) を提案し、その収束性を示す。
また、S-IBF の応用例として、与えられた写真から塗り
絵の下絵線画を生成する方法を提案する。本論文の主眼
点は、IBF における反復回数決定問題の一解決策とし
ての S-IBF の提案にあるが、その応用例である塗り絵
の下絵線画生成についても、これまで研究例が少ないこと
から、いくらか意味があると考えている。

塗り絵は、輪郭だけが描かれた図形や模様の中に色を
塗り分けて楽しむ知育玩具であり、従来は子供向けのも
のとされていたが、最近では大人向けのものも販売され
ている [7]。通常は、第三者から提供される題材の中か
らユーザが気に入ったものを選んで色を塗るので、題材
の選択は完全に能動的ではない。そこで、ユーザが用意
した写真などの画像から塗り絵の下絵を作成するソフト
ウェアが開発されている [8, 9]。しかし、その数はあまり
多くないようである。このようにユーザが自分で下絵を
作ることができれば、題材選びの幅が広がり、作品の希

少価値も高まると考えられる。そこで本論文でも、与え
られた画像から下絵線画を生成する方法を提案し、生成
した下絵を既存のソフトウェアによるものと比較する。

『ぬり絵ツクール』[8] は、そのようなソフトウェア
の一つであり、一つの領域を1色で塗ったようなイラスト
画などを塗り絵の下絵に変換するのに適しており、下
絵に色を付けて塗り絵を完成させることもできる。しか
し、連続的な色の変化を含む写真などでは、2値化した
ような画像になってしまい、実際の写真を塗り絵の下絵
線画にするのには、あまり適していないようである。『塗
り絵の下絵を作ろう』[9] は、与えられた画像から対象
物の輪郭線などを抽出して、塗り絵の下絵を生成するソ
フトウェアであり、アルゴリズムの詳細は不明であるが、
結果の画像から、Sobel フィルタなどのエッジ抽出フィ
ルタを用いているのではないかと推測される。このよう
に、塗り絵の下絵生成では、エッジ抽出フィルタを用い
るのが最も単純な方法であると考えられるが、ノイズも
エッジと一緒に抽出されて、あまりよい結果は得られな
い。そこで本論文では、ノイズを抑えながらエッジを抽
出するための一手段として、エッジ保存平滑化フィルタ
を利用する。

人手による実際の塗り絵の下絵では、元となる絵画や
写真に含まれる主要なエッジが線で表現され、細かなテ
クスチャやノイズは省略されることが多い。このような
下絵の生成手順は、元の画像の簡略化と、簡略化された
画像からのエッジ抽出という2段階の処理に分けて考え
ることができる。画像簡略化の処理には、エッジ保存平滑
化が適しているので、この部分に S-IBF を用いる。エッ
ジ抽出の処理では、アンシャープマスキングによるエッ
ジ強調成分を抽出し、明暗の差が大きいエッジを太い線
で描き、明暗の差が小さいエッジを細い線で描くよう
にする。アンシャープマスキングは古くからよく知られて
いる鮮鋭化フィルタの一つであるが、最近でもノンフォ
トリアリスティックレンダリングなどに用いられており
[10]、その有効性が示されている。そこで本論文では、ア
ンシャープマスキングに基づいたエッジ抽出法を提案す
る。この方法によって、エッジの強弱を線の太さに反映
させた下絵線画が得られる。更に、下絵線画を Potrace
[11] というソフトウェアを用いてベクターデータに変換
し、拡大縮小が可能な下絵線画を作成する。すなわち、
本論文で提案する塗り絵の下絵線画の生成手順は次の三
つの処理、すなわち、a) S-IBF によるエッジ保存平滑化、
b) エッジの強弱に応じて幅の異なる線の抽出、c) 下絵
線画のベクターデータ化、からなる。最後の手順 c) に
ついては、既存のソフトウェアを使用するので独自性は
ないが、2値画像の有用性を示す一例にはなっていると
考える。

[†]九州大学大学院芸術工学研究院

以下、2章で IBF を概説し、3章で S-IBF を提案し、4

章でS-IBFによる塗り絵の下絵線画の生成法を提案し、5章で実験例を示し、6章でむすびとする。

2. 反復バイラテラルフィルタ

n 個の画素からなるモノクロ画像が与えられるとし、第 i 画素の空間座標と画素値を、それぞれ $x_i, f_i \in \{0, 1, \dots, 255\}$ ($i = 1, \dots, n$) とすると、第 i 画素におけるバイラテラルフィルタ (bilateral filter [1]: BF) の出力は

$$b_i = \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(f_i, f_j) f_j}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(f_i, f_j)} \quad (1)$$

と表される。ここで $w_\alpha(x_i, x_j), w_\beta(f_i, f_j)$ はそれぞれ空間方向、画素値方向の重みであり、本論文では、

$$w_\alpha(x_i, x_j) = \exp(-\alpha \|x_i - x_j\|^2) \quad (2)$$

$$w_\beta(f_i, f_j) = \exp(-\beta \|f_i - f_j\|^2) \quad (3)$$

とする。 α, β は正定数である。 N_i は第 i 画素の近傍画素の集合である。本論文では、第 i 画素を中心とする正方形のウィンドウに含まれる画素の集合を W_i とし、与えられた画像に含まれる全画素の集合を Ω とすると、 $N_i = W_i \cap \Omega$ とする。すなわち、画像の端の部分で、ウィンドウが画像からはみ出す場合には、ウィンドウに含まれ、かつ画像の枠内にある画素だけをを用いて式(1)を計算する。従って、ここでは周期的境界条件は用いない。

BFを繰り返しかける反復BF(iterative BF: IBF)の第 $(t+1)$ 反復における出力は

$$b_i^{(t+1)} = \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(b_i^{(t)}, b_j^{(t)}) b_j^{(t)}}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(b_i^{(t)}, b_j^{(t)})} \quad (4)$$

と表される。 $b_i^{(t)}$ の初期値は $b_i^{(0)} = f_i$ とする。

定理 1 $t \rightarrow \infty$ のとき、 $b_i^{(t)}$ は一定値 c に収束する。すなわち、 $b_1^{(\infty)} = \dots = b_n^{(\infty)} = c$ が成り立つ。

(証明) $b_{\max}^{(t)} = \max_{i \in \Omega} \{b_i^{(t)}\}$, $b_{\min}^{(t)} = \min_{i \in \Omega} \{b_i^{(t)}\}$ とし、 $b_i^{(t+1)} = b_{\max}^{(t)}$ を満たす i の集合を $S_{\max}^{(t+1)}$ とし、 $b_i^{(t+1)} = b_{\min}^{(t)}$ を満たす i の集合を $S_{\min}^{(t+1)}$ とする。また、 $T^{(t+1)} = \Omega - S_{\max}^{(t+1)} - S_{\min}^{(t+1)}$ とする。このとき、式(4)より

$$b_i^{(t+1)} \leq \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(b_i^{(t)}, b_j^{(t)}) b_{\max}^{(t)}}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(b_i^{(t)}, b_j^{(t)})} \quad (5)$$

$$= b_{\max}^{(t)} \quad (6)$$

$$b_i^{(t+1)} \geq \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(b_i^{(t)}, b_j^{(t)}) b_{\min}^{(t)}}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(b_i^{(t)}, b_j^{(t)})} \quad (7)$$

$$= b_{\min}^{(t)} \quad (8)$$

が成り立ち、 $T^{(t+1)}$ には $b_i^{(t+1)} = b_{\min}^{(t)}$ または $b_i^{(t+1)} = b_{\max}^{(t)}$ となるような i は含まれないから、 $i \in T^{(t+1)}$ に対して $b_{\min}^{(t)} < b_i^{(t+1)} < b_{\max}^{(t)}$ が成り立つ。次に、

$N_i \cap T^{(t+1)} \neq \emptyset$ となるような $i \in S_{\max}^{(t+1)}$ が存在し、そのような $i \in S_{\max}^{(t+1)}$ に対して $b_i^{(t+2)} < b_{\max}^{(t+1)}$ が成り立つことから、 $S_{\max}^{(t+2)} \subset S_{\max}^{(t+1)}$ かつ $S_{\max}^{(t+2)} \neq S_{\max}^{(t+1)}$ が成り立つ。同様に、 $S_{\min}^{(t+2)} \subset S_{\min}^{(t+1)}$ かつ $S_{\min}^{(t+2)} \neq S_{\min}^{(t+1)}$ も成り立つ。よって、 $T^{(t+2)} \supset T^{(t+1)}$ かつ $T^{(t+2)} \neq T^{(t+1)}$ が成り立つ。すなわち、 $T^{(t+1)}$ の要素数 $|T^{(t+1)}|$ は t の増加と共に単調に増加していき、いずれ $T^{(t+1)} = \Omega$ となる。このとき、 $i \in \Omega$ に対して $b_{\min}^{(t)} < b_i^{(t+1)} < b_{\max}^{(t)}$ が成り立つ。その後、 $|T^{(t+1)}|$ は増減を繰り返すが、 $T^{(t+1)} = \Omega$ となるときに t に、改めて $t_0, t_1, \dots, t_k, \dots$ と番号を付けると、 $\lim_{k \rightarrow \infty} (b_{\max}^{(t_k)} - b_{\min}^{(t_k)}) = 0$, $[b_{\min}^{(t_{k+1})}, b_{\max}^{(t_{k+1})}] \subset [b_{\min}^{(t_k)}, b_{\max}^{(t_k)}]$ ($k = 0, 1, \dots$) が成り立ち、閉区間 $[b_{\min}^{(t_k)}, b_{\max}^{(t_k)}]$ に関する区間縮小法により、 $b_{\min}^{(t_k)} \leq c \leq b_{\max}^{(t_k)}$ ($k = 0, 1, \dots$) なる c が定まり、 $\lim_{k \rightarrow \infty} b_i^{(t_k)} = c$ となる。(証明終)

定理1は、IBFを収束するまで繰り返すと、画像に含まれるノイズ成分だけでなく、信号成分も平滑化されてしまうことを表しており、エッジ保存平滑化などを行う際には、IBFの反復を途中で打ち切る必要があることを意味している。しかし、例えば、ノイズ除去でピーク信号対雑音比を最大化するような最適な反復回数を求めるのは困難であり、適用例に応じて経験的に良い結果が得られる反復回数を選択されることが多い。そこで次章では、反復回数の設定が不要なIBFを提案する。

3. 符号付変化量の最大化

式(1)のBFによる画素値の変化量は $b_i - f_i$ であり、その符号は絶対値を1に規格化して $(b_i - f_i)/|b_i - f_i|$ ($|b_i - f_i| > 0$)と表される。この符号を含めた変化量の総和を最大化するようなIBFを考える:

$$\max_{\{\tilde{b}_i\}_{i=1}^n} \sum_{i=1}^n \frac{b_i - f_i}{|b_i - f_i|} (\tilde{b}_i^{(t)} - f_i) \quad (9)$$

ここでは、IBFの出力 $b_i^{(t)}$ との違いを明示するために、 $\tilde{b}_i^{(t)}$ と表記している。式(9)の目的関数を $E^{(t)}$ とおくと、IBFによって $E^{(t)}$ を増加させるには、BFによる変化量が正の画素については、IBFの出力が増加する間IBFをかけ続け、BFによる変化量が負の画素については、IBFの出力が減少する間IBFをかけ続ければよい。すなわち、初期値を $\tilde{b}_i^{(0)} = f_i$ とし、第1反復の出力を $\tilde{b}_i^{(1)} = b_i^{(1)} = b_i$ とし、第2反復以降の出力を

$$\tilde{b}_i^{(t)} = \begin{cases} \hat{b}_i^{(t)}, & (f_i - \tilde{b}_i^{(t-1)}) (\hat{b}_i^{(t)} - \tilde{b}_i^{(t-1)}) \geq 0 \\ \tilde{b}_i^{(t-1)}, & \text{otherwise} \end{cases} \quad (10)$$

とする。ここで $t \geq 2$ であり、

$$\hat{b}_i^{(t)} = \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(\tilde{b}_i^{(t-1)}, \tilde{b}_j^{(t-1)}) \tilde{b}_i^{(t-1)}}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(\tilde{b}_i^{(t-1)}, \tilde{b}_j^{(t-1)})} \quad (11)$$

である。式(10)の不等式は、

$$f_i \leq \tilde{b}_i^{(t-1)} \leq \hat{b}_i^{(t)} \quad (12)$$

または

$$\hat{b}_i^{(t)} \leq \tilde{b}_i^{(t-1)} \leq f_i \quad (13)$$

のときに成り立つ．式 (10) で表されるフィルタを S-IBF と略記することにする．

定理 2 式 (10) の反復によって, $E^{(t)}$ は単調に増加する．

(証明) $b_i - f_i \geq 0$ となる i の集合を S^+ とし, その他の i の集合を S^- とすると,

$$\begin{aligned} D^{(t)} &= E^{(t)} - E^{(t-1)} \\ &= \sum_{i=1}^n \frac{b_i - f_i}{|b_i - f_i|} (\tilde{b}_i^{(t)} - \tilde{b}_i^{(t-1)}) \quad (14) \end{aligned}$$

$$= \sum_{i \in S^+} (\tilde{b}_i^{(t)} - \tilde{b}_i^{(t-1)}) - \sum_{i \in S^-} (\tilde{b}_i^{(t)} - \tilde{b}_i^{(t-1)}) \quad (15)$$

と書ける．更に S^+ を, 式 (10) の不等式を満たす集合 S_A^+ と満たさない集合 S_B^+ とに分け, S^- も同様に S_A^- と S_B^- とに分けると,

$$D^{(t)} = \sum_{i \in S_A^+} (\tilde{b}_i^{(t)} - \tilde{b}_i^{(t-1)}) - \sum_{i \in S_A^-} (\tilde{b}_i^{(t)} - \tilde{b}_i^{(t-1)}) \quad (16)$$

と書ける． $i \in S_A^+$ については式 (12) が成り立ち, $i \in S_A^-$ については式 (13) が成り立つことから, $D^{(t)} \geq 0$ すなわち $E^{(t)} \geq E^{(t-1)}$ が得られる． (証明終)

定理 3 式 (10) の反復によって得られる数列 $\{\tilde{b}_i^{(2)}, \tilde{b}_i^{(3)}, \dots\}$ は収束する．

(証明) $f_{\max} = \max_{i \in \Omega} \{f_i\}$, $f_{\min} = \min_{i \in \Omega} \{f_i\}$ とすると, 式 (10), (11) より

$$\tilde{b}_i^{(t)} \leq \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(\tilde{b}_i^{(t-1)}, \tilde{b}_j^{(t-1)}) f_{\max}}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(\tilde{b}_i^{(t-1)}, \tilde{b}_j^{(t-1)})} \quad (17)$$

$$= f_{\max} \quad (18)$$

$$\tilde{b}_i^{(t)} \geq \frac{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(\tilde{b}_i^{(t-1)}, \tilde{b}_j^{(t-1)}) f_{\min}}{\sum_{j \in N_i} w_\alpha(x_i, x_j) w_\beta(\tilde{b}_i^{(t-1)}, \tilde{b}_j^{(t-1)})} \quad (19)$$

$$= f_{\min} \quad (20)$$

が成り立つ．すなわち, $\{\tilde{b}_i^{(2)}, \tilde{b}_i^{(3)}, \dots\}$ は有界である．また定理 2 より, $\tilde{b}_i^{(t)}$ の関数 $E^{(t)}$ は単調増加である．よって, $\{\tilde{b}_i^{(2)}, \tilde{b}_i^{(3)}, \dots\}$ は収束する． (証明終)

一般に, S-IBF の収束値は, IBF のそれとは異なり, IBF における定理 1 に対応して, S-IBF については, 次のことがいえる．

定理 4 $b_i \geq f_i$, $b_j \leq f_j$, $b_i > b_j$ を満たすような $i \in \Omega$, $j \in \Omega$ が存在するならば, $t \rightarrow \infty$ のとき, $\tilde{b}_i^{(t)}$ は一定値には収束しない．

(証明) $b_i \geq f_i$ を満たす i については, $\tilde{b}_i^{(t)}$ は t に関して単調非減少であり, $b_j \leq f_j$ を満たす j については, $\tilde{b}_j^{(t)}$ は t に関して単調非増加である．従って, $\tilde{b}_{\max}^{(t)} =$

$\max_{i \in \Omega} \{\tilde{b}_i^{(t)} | b_i \geq f_i\}$, $\tilde{b}_{\min}^{(t)} = \min_{j \in \Omega} \{\tilde{b}_j^{(t)} | b_j \leq f_j\}$ とおくと, $b_i > b_j$ を満たすような $i \in \Omega$, $j \in \Omega$ が存在することより, $\tilde{b}_{\min}^{(t)} \leq b_j < b_i \leq \tilde{b}_{\max}^{(t)}$ ($t = 2, 3, \dots$) , すなわち, $\lim_{t \rightarrow \infty} (\tilde{b}_{\max}^{(t)} - \tilde{b}_{\min}^{(t)}) > 0$ が成り立つ．よって, $t \rightarrow \infty$ のとき, 少なくとも二つの異なる画素値 $\tilde{b}_{\max}^{(\infty)}$, $\tilde{b}_{\min}^{(\infty)}$ をもつ画素が存在するので, $\tilde{b}_i^{(t)}$ は一定値には収束しない． (証明終)

このように, IBF と S-IBF の収束先は一般に異なる．このことは, IBF による画素値の変化が一方向的でないことの傍証になっている．すなわち IBF では, 画素値が増加した後に減少に転じたり, 減少した後に増加に転じたりすることがあり得る．そのため, IBF の反復の初期の段階で BF のエッジ強調性 [5] によって強調されるエッジも, 反復を繰り返していくと, いずれ平滑化されてしまう．それに対して S-IBF では, そのような画素値の増減の反転を禁じており, それによって信号成分が平滑化されてしまうのを抑制している．そのため S-IBF では, IBF のように反復回数を設定する必要はなく, 画素値が収束するまで反復を繰り返し, 収束値を出力すればよい．

4. 塗り絵の下絵線画の生成への応用

本章では, S-IBF を塗り絵の下絵線画の生成に応用する方法を示す．実際の塗り絵の下絵線画では, 元の絵画や写真の主要なエッジが線で表現され, 細かなテクスチャは省略されることが多い．このような塗り絵の下絵線画の生成手順は, 元の画像の簡略化と, 簡略化された画像からのエッジ抽出という 2 段階の処理に分けて考えることができる．最初の画像の簡略化の処理は, 細かなテクスチャやノイズを平滑化するのが目的であるが, その後のエッジ抽出を良好に行うために, 主要なエッジは平滑化しないほうがよい．ここでは, このようなエッジ保存平滑化に S-IBF を利用する．

次に, S-IBF によって簡略化された画像からエッジを抽出する手順を示す．S-IBF による出力画像を $\tilde{b} = [\tilde{b}_1, \dots, \tilde{b}_n]$ とし, \tilde{b} を入力画像とするガウスフィルタ (Gaussian filter: GF) の出力画像を $g = [g_1, \dots, g_n]$ とする． \tilde{b} と g から得られる塗り絵の下絵を $s = [s_1, \dots, s_n]$ とし, その第 i 画素値を

$$s_i = 255 \left(1 - \frac{h_i}{h_{\max}} \right) \quad (21)$$

とする．ここで

$$h_i = \begin{cases} g_i - \tilde{b}_i & \text{if } g_i > \tilde{b}_i, \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

であり, $h_{\max} = \max_i \{h_i\}$ である．また, 下絵線画を $d = [d_1, \dots, d_n]$ とし,

$$d_i = \begin{cases} 0 & \text{if } s_i < \theta, \\ 255 & \text{otherwise} \end{cases} \quad (23)$$

とする．すなわち, 下絵線画 d は下絵 s を閾値 θ で 2 値化したものである．式 (22) の $g_i - \tilde{b}_i$ は, \tilde{b} をアンシャープマスキングで強調するときの強調成分であり, 明暗が接するエッジの部分では, 暗い領域で正になり, 明るい



図 1: 画像例: (a) 入力画像 (256 × 256 画素), (b) IBF の出力画像, (c) S-IBF の出力画像

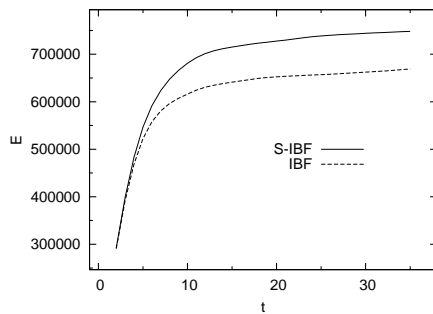


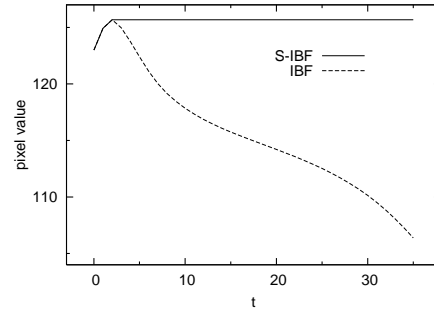
図 2: 目的関数 $E^{(t)}$ の変化

領域で負になる．従って，式 (23) のように 2 値化すると，暗い領域に黒色の輪郭線が描かれることになる．また，明暗の差が大きいエッジには太い線が描かれ，明暗の差が小さいエッジには細い線が描かれる．実際の塗り絵の下絵線画でも，このように太さの異なる線が使い分けられることが多い．

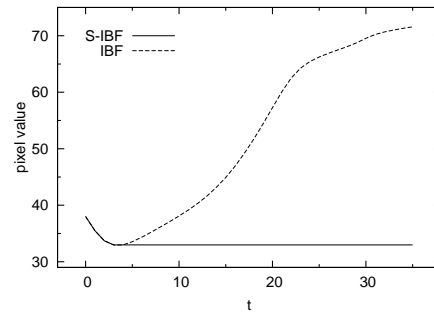
上記の方法では，2 値化を行う前の下絵 s を出力としてもよいが，2 値化までしておくとして，Potrace [11] などのソフトウェアを用いて，拡大縮小が可能なベクターデータに変換できるという利点がある．そこで本論文では，グレースケール画像 s を下絵と呼び，それを 2 値化した d を下絵線画と呼んで，両者を区別することにする．

5. 実験例

最初に，図 1(a) に示す画像を入力として実験した結果を示す．図 2 に目的関数 $E^{(t)}$ の変化の様子を示す．横軸は反復回数 t である．実線は S-IBF を示し，破線は IBF を示す．S-IBF のほうが IBF よりも $E^{(t)}$ の値が大きく，定理 2 に示した通り， t の増加に伴って単調に増加している．S-IBF の収束の条件は， $D^{(t)}/E^{(t-1)} < \theta$ とし， $\theta = 10^{-3}$ とした．この例では $t = 35$ で収束した．計算時間は 636.5 秒であった．計算環境は Pentium 4(R) CPU 3.40GHz, 2.00GB RAM であり，プログラミング言語は Microsoft Visual C++ である． $t = 35$ での IBF と S-IBF の出力画像をそれぞれ図 1(b), (c) に示す．全体的にはよく似た画像になっているが，右側の背景や女性の首の下の部分に違いが見られる． $\alpha = (2\sigma_\alpha^2)^{-1}$ ， $\beta = (2\sigma_\beta^2)^{-1}$ とし， $\sigma_\alpha = \sigma_\beta = 10$ とした．また，正方形ウィンドウ W_i は， $W_i = \{(i+k, j+l)\}$ ， $-z \leq k \leq z$ ， $-z \leq l \leq z$ とし， $z = 4\sigma_\alpha = 40$ とした．この画像上の位置 (119,



(a)



(b)

図 3: 画素値の変化: (a) 位置 (119, 183), (b) 位置 (191, 143)

183), (191, 143) における画素値の変化の様子をそれぞれ図 3(a), (b) に示す．横軸は t である．実線は S-IBF を示し，破線は IBF を示す．IBF による画素値 (破線) は，同図 (a) では増加した後に減少し，(b) では減少した後に増加している．それに対して S-IBF による画素値 (実線) は，最初に増加または減少して極値に達した後はその値が保たれている．IBF において，BF のエッジ強調性が現れるのは，入力画像から BF の出力画像 (あるいは IBF の第 1 反復における出力) への変化量の符号が変わらない反復の初期の段階である．従って，IBF によって BF のエッジ強調性を最大限引き出すには，画素値の符号付変化量が極大あるいは極小となる時点で反復を終了するのがよいと考えられるが，その反復回数は一般に画素によって異なり (例えば，図 3(a) では $t = 2$ で極大になり，同図 (b) では $t = 3$ で極小になっている)，全画素について符号付変化量が最適となる反復回数を定めるのは困難である．そこで S-IBF では，極値に達した画素値を保持することによって，IBF における反復回数決定の問題を避けながら，一度強調されたエッジが平滑化されるのを防いでいる．

図 1(b), (c) から生成した塗り絵の下絵と下絵線画を図 4 に示す．GF のガウス関数の標準偏差は 1 とし，式 (23) の 2 値化の閾値は $\theta = 245$ とした．図 4(c), (d) を，Potrace [11] によってベクターデータに変換した結果を，それぞれ図 5(a), (b) に示す．下絵 (図 4(a), (b)) を 2 値化して下絵線画 (同図 4(c), (d)) にすることで，このようなベクターデータへの変換が可能になる．

図 6 に塗り絵の下絵線画の生成例を示す．図 6(a), (g),

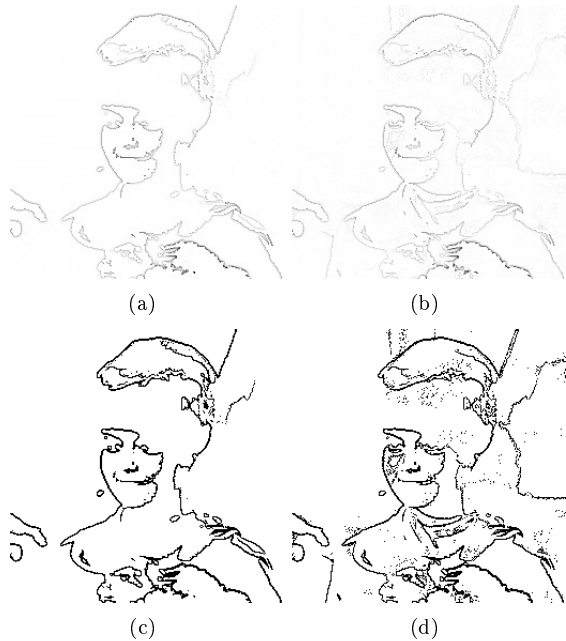


図4: 塗り絵の下絵と下絵線画: 図1(b), (c) から生成した下絵 ((a), (b)) と下絵線画 ((c), (d))

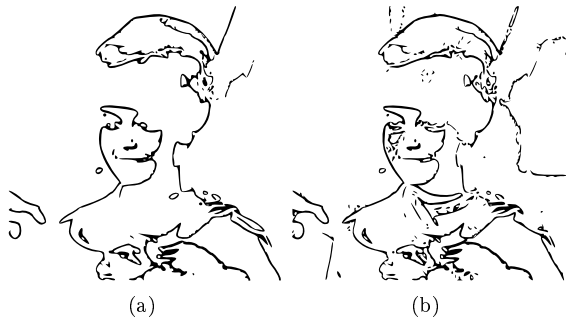


図5: Potrace [11] によるベクターデータへの変換例: (a), (b) は, それぞれ図4(c), (d) から生成したもの

(m) は入力画像 f であり, 各画像から生成された画像をそれらの横に並べている. 図6(a), (g), (m) の画素数は, それぞれ 480×640 , 364×400 , 480×640 である. 図6(b), (h), (n) は S-IBF の出力画像 \tilde{f} であり, 同図 (c), (i), (o) は下絵 s であり, 同図 (d), (j), (p) は下絵線画 d である. 図6(e), (k), (q) はソフトウェア「塗り絵の下絵を作るう」[9] による下絵の作成例である. このソフトウェアには, 作成オプションとして「線の強さ」と「線を除去」を11段階(ここではそれを $0, 1, \dots, 10$ と表すことにする)で調節できるようになっており, 更に「ノイズ除去」を行うか否かを選択できるようになっている. 図6(e), (k), (q) は「線の強さ」と「線を除去」を5とし「ノイズ除去」を行うにした結果である. 図6(f), (l), (r) はそれぞれ, Potrace [11] によって図6(d), (j), (p) をベクターデータに変換した結果である. 図6(b), (h), (n) を生成するのに要した S-IBF の反復回数は, それぞれ39回, 42回, 36回である. 図6(b), (h), (n) では, ノ

イズや細かなテクスチャが平滑化され, 主要なエッジが強調されて, 画像が簡略化されている. 図6(c), (i), (o) の下絵では, 図6(b), (h), (n) におけるエッジの強弱が画素の濃淡で表されている. 図6(d), (j), (p) の下絵線画は, それぞれ同図 (c), (i), (o) を2値化したものであり, 同図 (b), (h), (n) におけるエッジの強弱が線の太さに反映されている. 図6(e), (k), (q) の既存のソフトウェア [9] による下絵はグレースケール画像であり, 本手法では, 2値化する前の同図 (c), (i), (o) に対応する. これらと比較すると, 図6(c), (i), (o) のほうが対象物の細かな形状を正確に抽出できており, エッジの途切れも少ない. 図 (d), (j), (p) では, 2値化によって図 (c), (i), (o) で抽出されたエッジがより明確になっている. 図 (f), (l), (r) は, 入力画像と同じ解像度で表示すると図 (d), (j), (p) とほぼ同じであるが, 広い紙に拡大して印刷する場合などには, 画素の形状を目立たせることなく, 滑らかな線で下絵線画を表示することができる. 例として, 拡大図を図7に示す. 図7(a), (c), (e) は, それぞれ図6(d), (j), (p) の一部であり, 図7(b), (d), (f) は, それぞれ図6(f), (l), (r) の一部である. 図7(a), (c), (e) ではジャギーが目立つが, 図7(b), (d), (f) では, 拡大しても滑らかな線が保たれている.

6. むすび

反復バイラテラルフィルタ (IBF) における反復回数の決定の問題を解決するための方法として, 画素値の符号付変化量を最大化する IBF (S-IBF) を提案し, 塗り絵の下絵線画の生成に応用した. 下絵線画の生成手順を, 画像の簡略化とエッジの抽出の2段階に分けて考え, S-IBF で画像を簡略化した後, アンシャープマスキングの考え方に基づくエッジ抽出を行い, エッジの強度を線の太さに反映させた下絵線画を生成した. また, Potrace によって下絵線画をベクターデータに変換し, 拡大縮小印刷にも対応できるようにした. 既存のソフトウェアで生成した下絵と比較して, 線の途切れが少ない下絵線画が得られることを実験で確認した. S-IBF をカラー画像やビデオ用に拡張するのが今後の課題である.

参考文献

- [1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Proc. ICCV, 1998.
- [2] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," Proc. SIGGRAPH '02, pp. 257-266, 2002.
- [3] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," ACM Trans. Graph., vol. 23, no. 3, pp. 673-678, Aug. 2004.
- [4] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," ACM Trans. Graphics, vol. 22, no. 3, pp. 950-953, Jul. 2003.
- [5] 浦浜喜一, 井上光平, "バイラテラルフィルタのエッジ強調性," 信学論, vol. J86-A, no. 3, pp. 326-328, Mar. 2003.
- [6] 常セン, 井上光平, 浦浜喜一, "反復バイラテラルフィルタによるコントラスト強調," 映情学誌, vol. 63, no. 7, pp. 999-1001, Jul. 2009.
- [7] P・J・ルドウーテ原画, 河出書房新社編集部編, "大人の塗り絵 美しい花編," 河出書房新社, 2005.
- [8] 仁きよし, "ぬり絵ツクール"
URL: <http://www15.ocn.ne.jp/ugoku/>

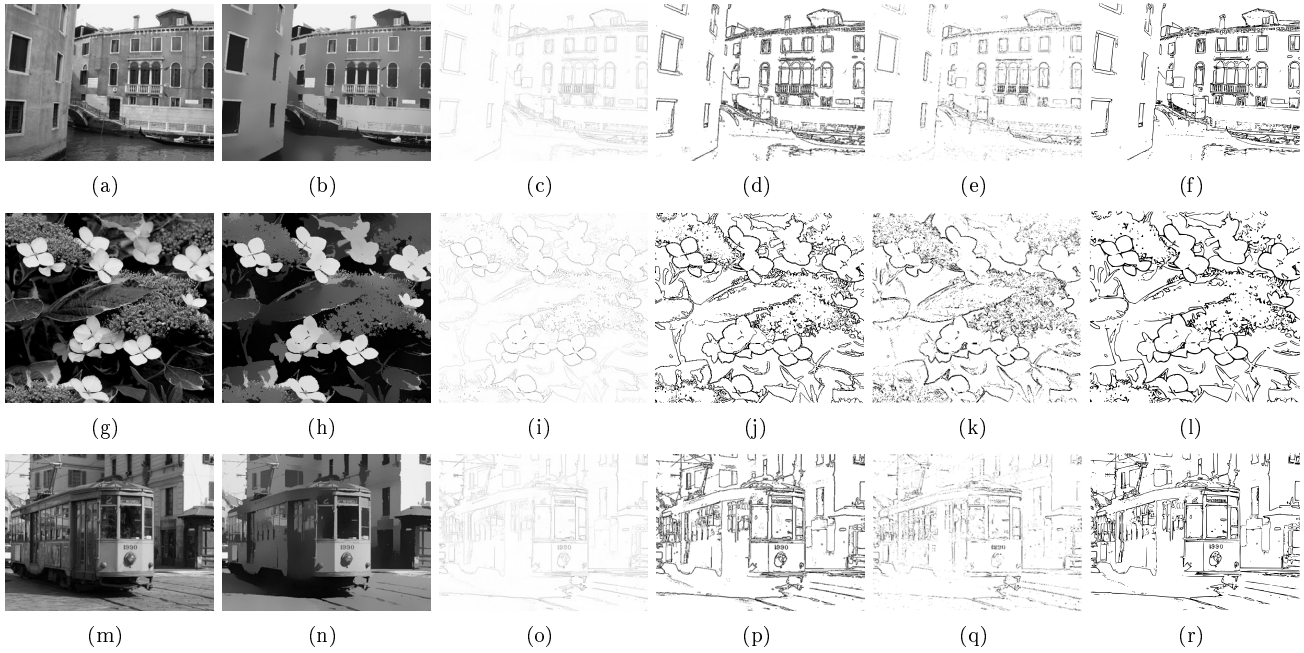


図 6: 塗り絵の下絵線画の生成例: (a), (g), (m) 入力画像 f , (b), (h), (n) S-IBF の出力画像 \tilde{b} , (c), (i), (o) 下絵 s , (d), (j), (p) 下絵線画 d , (e), (k), (q) 既存のソフトウェア [9] による作成例, (f), (l), (r) Potrace [11] による (d), (j), (p) の変換結果

- [9] Aquariansoft, “塗り絵の下絵をつくろう”
URL: <http://www.aqsoft.jp/nurie/index.html>
- [10] D. Mould and K. Grant, “Stylized black and white images from photographs,” Proc. NPAR, pp. 49–58, 2008.
- [11] P. Selinger, “Potrace: a polygon-based tracing algorithm,” Sept. 2003.
URL: <http://potrace.sourceforge.net/>

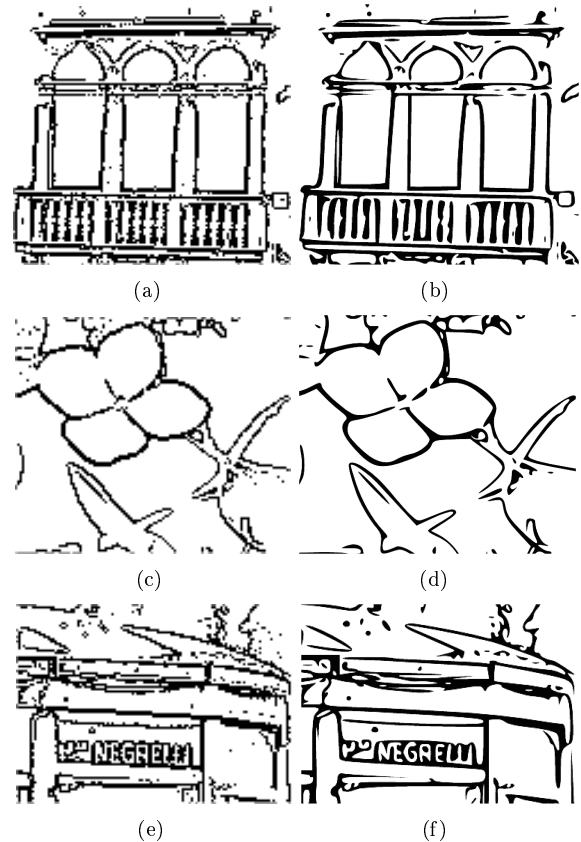


図 7: 拡大図: (a), (c), (e) は, それぞれ図 6(d), (j), (p) の一部, (b), (d), (f) は, それぞれ図 6(f), (l), (r) の一部