

フィルタリング処理によるノンフォトリアリスティックレンダリングの動画への拡張

代表研究者

平岡透

長崎県立大学 情報システム学部 教授

1 はじめに

本研究では、次の三つのことを実施して明らかにした。

- ① 申請者がこれまでに開発してきた画像処理フィルタを用いた静止画の各種ノンフォトリアリスティックレンダリング (NPR) を、ちらつきとシャワードア効果を抑制して動画に拡張する方法の開発
- ② 新しいタイプの画像処理フィルタの開発と NPR への拡張
- ③ 画像処理フィルタの高速化の方法の開発

2 画像処理フィルタの NPR の動画への拡張

静止画の NPR を動画の各フレームに適用するとちらつきとシャワードア効果が問題となるため、動画の前後のフレームを用いることでちらつきとシャワードア効果を抑制した。具体的には、逆アイリスフィルタを用いたセル風動画、逆アイリスフィルタを用いたチェッカー模様動画、逆線集中度フィルタを用いた縞模様動画、逆ソーベルフィルタを用いた波紋模様風動画、ストライプアート動画を生成した。以下、逆アイリスフィルタを用いたチェッカー模様動画の詳細を示す。

2-1 アイリスフィルタを用いたチェッカー模様動画

(1) 方法

256 階調で $I \times J$ 画素のグレースケールの動画の m 番目のフレームの画素値を $g_{i,j,m}$ ($i = 1, 2, \dots, I; j = 1, 2, \dots, J; m = 1, 2, \dots, M$) とする。 m 番目のフレームから前後 N 枚のフレームを用いて $f_{i,j,m}$ を

$$f_{i,j,m} = \frac{\sum_{n=-N}^N \frac{1}{1+|n|} g_{j,j,m+n}}{\sum_{n=-N}^N \frac{1}{1+|n|}} \quad (1)$$

で計算する。

アイリスフィルタをかけた後の画素値を $IF(f_{i,j,m})$ とする。アイリスフィルタは、注目画素 (i, j) から半径 r 画素内の S の周辺画素 (k, l) を用いて実行される。周辺画素 (k, l) から注目画素 (i, j) へのベクトル $(i - k, j - l)$ と周辺画素 (k, l) の濃度変化量から求めたベクトル $((f_{k+2,l+2,m} + f_{k+2,l+1,m} + f_{k+2,l,m} + f_{k+2,l-1,m} + f_{k+2,l-2,m}) - (f_{k-2,l+2,m} + f_{k-2,l+1,m} + f_{k-2,l,m} + f_{k-2,l-1,m} + f_{k-2,l-2,m}), (f_{k+2,l+2,m} + f_{k+1,l+2,m} + f_{k,l+2,m} + f_{k-1,l+2,m} + f_{k-2,l+2,m}) - (f_{k+2,l-2,m} + f_{k+1,l-2,m} + f_{k,l-2,m} + f_{k-1,l-2,m} + f_{k-2,l-2,m}))$ のなす角を $\theta_{i,j,k,l,m}$ とすると、注目画素 (i, j) の集中度 $c_{i,j,m}$ は、

$$c_{i,j,m} = \frac{1}{S} \sum_{k=i-r}^{i+r} \sum_{l=j-r}^{j+r} \cos \theta_{i,j,k,l,m} \quad (2)$$

ただし、半径 r 以内の周辺画素のみを使用

で計算される。この集中度 $c_{i,j,m}$ のうちの最小値を $c_{min,m}$ 、最大値を $c_{max,m}$ として、集中度 $c_{i,j,m}$ を 256 階調の値を持つ集中度 $C_{i,j,m}$ に式(3)で変換する。

$$C_{i,j,m} = 255 \left(\frac{c_{i,j,m} - c_{min,m}}{c_{max,m} - c_{min,m}} \right) \quad (3)$$

$IF(f_{i,j,m}) = C_{i,j,m}$ となる。

逆アイリスフィルタを

$$f_{i,j,m}^{(t)} = a \left(f_{i,j,m}^{(t-1)} - IF(f_{i,j,m}^{(t-1)}) \right) + g_{i,j,m} \quad (4)$$

で計算する.

ここで, a は正定数, t は反復回数である. 初期値 $f_{i,j,m}^{(0)}$ を $f_{i,j,m}$ として, 式(5)から $f_{i,j,m}^{(1)}, f_{i,j,m}^{(2)} \dots$ を求める. $f_{i,j,m}^{(t)}$ が0より小さい値になった場合は0, $f_{i,j,m}^{(t)}$ が255より大きい値になった場合は255とする. また, $f_{i,j,m}^{(t)}$ を前後 N 枚のフレームを用いて,

$$f_{i,j,m}^{(t)} = \frac{\sum_{n=-N}^N \frac{1}{1+|n|} f_{i,j,m+n}^{(t)}}{\sum_{n=-N}^N \frac{1}{1+|n|}} \quad (5)$$

で更新する.

逆アイリスフィルタを T 回かけて式(5)で作成されたチェッカー模様画像から生成された動画が提案チェッカー模様動画である.

(2) 結果

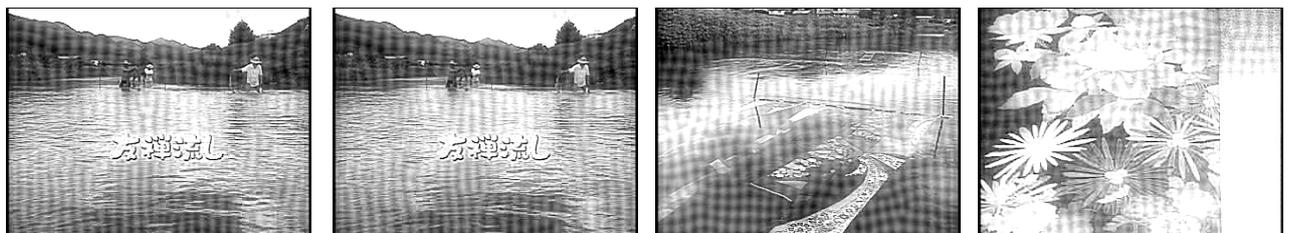
提案法の有効性を評価する実験を行う上で, 電子情報通信学会パターン認識・メディア理解研究専門委員会の評価用映像メディアデータベース検討部会の映像処理評価用映像データベースの友禅染めの動画(703フレーム(30フレーム/秒), 大きさ352×240画素, 256階調, グレースケール)を用いた. この動画の100フレーム目と101フレーム目, 387フレーム目, 388フレーム目を図1に示す. 387フレーム目と388フレーム目は, シーンが切り替わる前後のフレームである. この動画から生成された従来チェッカー模様動画と提案チェッカー模様動画を視覚的にかつ定量的に比較した. 以下の実験では, 文献[1]を参考に, 従来法と提案法のパラメータ r, a, T の値をそれぞれ5, 0.5, 20とした.



(a) 100 フレーム目 (b) 101 フレーム目 (c) 387 フレーム目 (d) 388 フレーム目

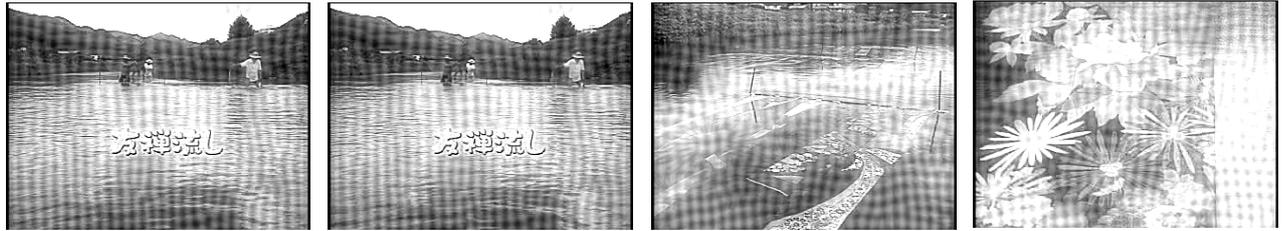
図1 友禅染めの動画のフレーム

まず, 従来チェッカー模様動画と提案チェッカー模様動画を視覚的に比較した. ここで, 従来チェッカー模様動画とは, 動画の各フレームに対してチェッカー模様画像を作成してチェッカー模様動画にしたものである. 提案法のパラメータ N の値を3とした. 参考のため, 従来チェッカー模様動画と提案チェッカー模様動画の100フレーム目と101フレーム目, 387フレーム目, 388フレーム目をそれぞれ図2と図3に示しておく. 従来チェッカー模様動画と提案チェッカー模様動画を見ると, 提案チェッカー模様動画は, 従来チェッカー模様動画のちらつきが抑制され, チェッカー模様が滑らかに変化することがわかった.



(a) 100 フレーム目 (b) 101 フレーム目 (c) 387 フレーム目 (d) 388 フレーム目

図2 従来チェッカー模様動画のフレーム



(a) 100 フレーム目 (b) 101 フレーム目 (c) 387 フレーム目 (d) 388 フレーム目

図3 提案チェッカー模様動画のフレーム

次に、従来チェッカー模様動画と提案チェッカー模様動画を定量的に比較した。従来チェッカー模様動画と提案チェッカー模様動画に対して、各フレームの画素値の差の絶対値の平均を求め、この平均の全フレームでの平均（以下、平均）を求めた。また、全フレームにおいて各フレームの画素値の差の絶対値が最大となる値（以下、最大値）も求めた。この平均と最大値が小さいほど、従来チェッカー模様動画および提案チェッカー模様動画のちらつきが少ないことを意味する。従来チェッカー模様動画の平均と最大値は、表1に示すように、それぞれ 23.871 と 83.096 であった。最大値は、シーンが切り替わる 387 と 388 フレーム目であった。一方、提案法のパラメータ N の値を 1, 2, ..., 10 と変えて得られた平均と最大値を表2に示す。最大値は、すべての場合でシーンが切り替わる 387 と 388 フレーム目であった。表2を見ると、 N の値が大きいくほど、平均と最大値が小さくなる傾向があることがわかった。表1と表2を見ると、提案チェッカー模様動画は従来チェッカー模様動画よりも平均と最大値がともに小さくなっていることがわかった。つまり、提案チェッカー模様動画は従来チェッカー模様動画よりもちらつきが抑制されているといえる。

表1 従来チェッカー模様動画の平均と最大値

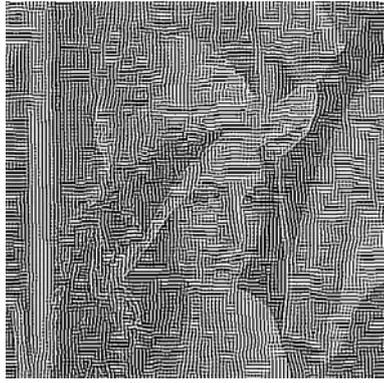
平均	最大値
23.871	83.096

表2 提案チェッカー模様動画の N による平均と最大値

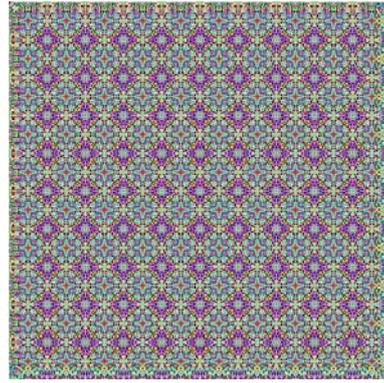
N	平均	最大値
1	15.336	64.004
2	14.151	59.869
3	13.697	57.957
4	13.446	57.050
5	13.312	56.423
6	13.252	55.845
7	13.197	55.425
8	13.136	55.115
9	13.109	54.916
10	13.111	54.657

3 新しいタイプの画像処理フィルタの開発と NPR への拡張

新しいタイプ画像処理フィルタを開発して NPR に応用した。具体的には、 k 平均法と逆フィルタを用いたストライプパッチワーク風ハーフトーニング、特異値分解と逆フィルタを用いた木炭画風画像、輪郭線とユークリッド距離を用いた文様群風画像、エントロピーと逆フィルタを用いたストライプパッチワーク風画像を生成した。これらの非写実的な画像を図4に示す。



(a) ストライプパッチワーク風ハーフトーン



(b) 文様模様群風画像



(c) 木炭画風画像



(d) ストライプパッチワーク風画像

図 4 非写実的な画像

4 画像処理フィルタの高速化の方法の開発

画像処理フィルタは計算量が大きく、NPR の画像処理フィルタを動画に拡張するとさらに計算量が大きくなるため、画像処理フィルタの高速化の方法を開発した。具体的には、バイラテラルフィルタ (BF) を用いてモアレ風画像を生成する処理において、BF の係数を二値化することで計算量を削減する方法を提案し、さらに生成されるモアレ風画像のクオリティを向上させた。

4-1 2 値係数バイラテラルフィルタによるモアレ風画像の高速化と高品質化

(1) 方法

モアレ風画像は、大きく二つの処理を実行することで生成される。一つ目の処理では、2 値係数 BF を用いて段階効果のある画像を作成する。二つ目の処理では、一つ目の処理で得られた画像に 2 値係数 BF とアンシャープマスクを組み合わせたフィルタ (反 2 値係数 BF) をかけて、段階効果を強調したモアレ風画像を生成する。

写真画像の座標 (i, j) での入力画素値を $f_{1,i,j}^{(0)}$ をとして、BF での出力画素値 $f_{1,i,j}^{(t)}$ (t は反復回数) を式(6)と式(7)で計算する。

$$f_{1,i,j}^{(t)} = \frac{\sum_{k=i-W}^{i+W} \sum_{l=j-W}^{j+W} w_{1,i,j,k,l}^{(t-1)} f_{1,k,l}^{(t-1)}}{\sum_{k=i-W}^{i+W} \sum_{l=j-W}^{j+W} w_{1,i,j,k,l}^{(t-1)}} \quad (6)$$

$$w_{1,i,j,k,l}^{(t)} = e^{-\alpha((i-k)^2 + (j-l)^2) - \beta(f_{1,i,j}^{(t)} - f_{1,k,l}^{(t)})^2} \quad (7)$$

ここで、 α と β は正定数、 W はウィンドウサイズである。

2 値係数 BF では、式(7)の代わりに式(8)を用いる。

$$w_{1,i,j,k,l}^{(t)} = \begin{cases} 1 (\alpha((i-k)^2 + (j-k)^2) + \beta(f_{1,i,j}^{(t)} - f_{1,k,l}^{(t)})^2 < C) \\ 0 \text{ (otherwise)} \end{cases} \quad (8)$$

ここで、 C は正定数である。式(8)は、BF の重み係数 $w_{1,i,j,k,l}^{(t)}$ を $\alpha((i-k)^2 + (j-k)^2) + \beta(f_{1,i,j}^{(t)} - f_{1,k,l}^{(t)})^2$ が C より小さい場合は 0 に、それ以外は 1 とするものである。

2 値係数 BF を T_1 回かけた後の画像 $f_{1,i,j}^{(T_1)}$ を $f_{2,i,j}^{(0)}$ として、反バイラテラルフィルタでの出力画素値 $f_{2,i,j}^{(t)}$ (t は反復回数) を式(9)と式(10)で計算する。

$$f_{2,i,j}^{(t)} = 2f_{2,i,j}^{(t-1)} - \frac{\sum_{k=i-W}^{i+W} \sum_{l=j-W}^{j+W} w_{i,j,k,l}^{(t-1)} f_{1,k,l}^{(t-1)}}{\sum_{k=i-W}^{i+W} \sum_{l=j-W}^{j+W} w_{i,j,k,l}^{(t-1)}} \quad (9)$$

$$w_{2,i,j,k,l}^{(t)} = \begin{cases} 1 (\alpha((i-k)^2 + (j-k)^2) + \beta(f_{1,i,j}^{(t)} - f_{1,k,l}^{(t)})^2 < C) \\ 0 \text{ (otherwise)} \end{cases} \quad (10)$$

ここで、 α と β 、 W 、 C の値は、2 値係数 BF と同じ値を用いる。

反 2 値係数 BF を T_2 回かけた後の画像が提案法によって生成されるモアレ風画像である。

(2) 結果

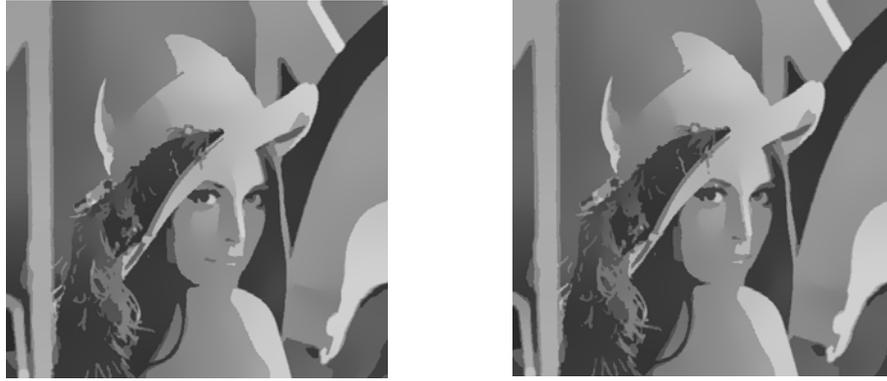
512×512 画素の大きさで 256 階調のレナの画像を用いて実験を行った。文献[2][3]を参考に、パラメータ α 、 β 、 W の値をそれぞれ 0.01、0.01、20 とした。また、従来法において T_1 と T_2 の値をそれぞれ 20 と 40 としたモアレ風画像 (図 5 参照) を生成した。計算環境は、OS が Windows 10 Enterprise, CPU が 3.20GHz, メモリが 8.00GB, グラフィックボードが Intel(R) HD Graphics 530, 使用言語は VC++ である。

まず、段階効果のある画像を作成する一つ目の処理を通して、提案法と従来法の計算時間について検証した。BF の反復回数 1 回の計算時間は 16.661 秒であった。一方、2 値係数 BF の反復回数 1 回の計算時間は 3.329 秒であった。つまり、提案法は従来法よりも処理を 5.005 倍高速化できることがわかる。



図 5 従来法のモアレ風画像

BF で $T_1 = 20$ として作成した画像 (BF 画像) を図 5 (a)に示す。2 値係数 BF で T_1 と C の値を変えた組合せによって作成された画像と BF 画像との画素値の差の絶対値の平均 (較差平均) を表 3 に示す。このとき、 T_1 の値を 1 から 20 で 1 刻み、 C の値を 2.0, 2.5, 3.0, 3.5, 4.0 と変えて実験を行った。表 1 より、 T_1 の値が大きくなるにつれて較差平均が小さくなり、ある値で極小値となり、それ以降は徐々に大きくなることわかる。また、 C の値が大きくなるにつれて較差平均の極小値が小さくなり、ある値で極小値となり、それ以降は徐々に大きくなることもわかる。よって、 T_1 と C の値はそれぞれ 12 と 3.0 前後がよいと考えられる。2 値係数 BF で T_1 と C がそれぞれ 12 と 3.0 の場合の画像を図 5 (b)に示す。図 5 を見ると、BF 画像と 2 値係数 BF の画像はほぼ同じである。ここで、 T_1 は提案法で 12 回、従来法で 20 回で、提案法の方が従来法よりも T_1 の値が小さいことから、提案法を用いることでさらなる処理の高速化が図れることがわかる。



(a) BF 画像

(b) 2 値係数 BF の画像

図 5 BF ($T_1 = 20$) と 2 値係数 BF ($T_1 = 12, C = 3.0$) の画像

最後に、提案法を用いて生成されるモアレ風画像を視覚的に確認した。提案法によって生成したモアレ風画像を図 6 に示す。このとき、前段落の結果を参考に T_1 と C の値をそれぞれ 12 と 3.0 とし、 T_2 の値を 10, 20, 30, 40 と変えて実験を行った。図 6 より、 T_2 の値を大きくするにつれてモアレ模様が発生していく様子がわかる。図 6 (d) ($T_2 = 40$ の場合) を見ると、従来法ではモアレ模様が発生しにくかった領域 (図 5 (b) 参照) にもモアレ模様が発生し、モアレ風画像全体に漏れなくモアレ模様が発生していることがわかる。

表 1 較差平均

		C				
		2.0	2.5	3.0	3.5	4.0
T_1	1	6.211	5.978	5.765	5.571	5.388
	2	5.398	4.990	4.628	4.312	4.039
	3	4.786	4.242	3.772	3.396	3.120
	4	4.331	3.675	3.124	2.728	2.491
	5	3.994	3.243	2.628	2.221	2.046
	6	3.723	2.903	2.232	1.842	1.767
	7	3.489	2.616	1.908	1.573	1.633
	8	3.287	2.361	1.641	1.391	1.627
	9	3.113	2.138	1.418	1.308	1.745
	10	2.956	1.933	1.244	1.332	1.935
	11	2.810	1.749	1.129	1.461	2.146
	12	2.677	1.590	1.121	1.631	2.350
	13	2.553	1.455	1.215	1.803	2.540
	14	2.440	1.352	1.358	1.962	2.714
	15	2.334	1.302	1.502	2.108	2.871
	16	2.238	1.314	1.643	2.246	3.012
	17	2.156	1.358	1.779	2.372	3.143
	18	2.094	1.424	1.910	2.491	3.260
	19	2.063	1.500	2.037	2.601	3.365
	20	2.064	1.584	2.154	2.706	3.461



(a) $T_2 = 10$



(b) $T_2 = 20$



(c) $T_2 = 30$



(d) $T_2 = 40$

図6 T_2 の値を変えた場合の提案法のモアレ風画像

5 まとめ

申請者がこれまでに開発してきた画像処理フィルタを用いた静止画の各種 NPR を、ちらつきとシャワードア効果を抑制して動画に拡張する方法を開発した。また、新しいタイプの画像処理フィルタを開発し、NPR へ拡張した。また、画像処理フィルタの高速化の方法を開発した。

今後の課題は、まだ本研究において適用していない静止画の NPR を動画に拡張することである。また、静止画の NPR を 3次元動画に拡張し、さらに本研究の動画の NPR と組み合わせた 3次元動画の NPR を開発することも今後の課題である。

【参考文献】

- [1] 平岡透, 熊野稔, 浦浜喜一: 逆アイリスフィルタによるチェッカー模様画像の生成, 芸術科学会論文誌, Vol.14, No.4, pp.165-169, 2015.
- [2] 平岡透, 浦浜喜一: バイラテラルフィルタによるモアレ風画像の生成, 映像情報メディア学会誌, Vol.67, No.2, pp.J74-J77, 2013.
- [3] 宮下啓, 平岡透, 浦浜喜一: モアレ風画像生成のためのバイラテラルフィルタのパラメータの検証, 映像情報メディア学会誌, Vol.67, No.9, pp.J356-J35, 2013.

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
Generating Stripe-Patchwork-Like Halftoning by k-Means Clustering and Inverse Filter	ICIC Express Letters	May. 2017
Generation of Charcoal-Drawing-Like Images by Singular Value Decomposition and Inverse Filter	ICIC Express Letters	Jul. 2017
輪郭線とユークリッド距離を用いた文様群画像の生成	産業応用工学会論文誌	Sep. 2017
Moire-Like Images Using Binarized-Weight Bilateral Filter for Higher Quality and Speed	ICIC Express Letters	Nov. 2017
Generation of Stripe-Patchwork Images by Entropy and Inverse Filter	ICIC Express Letters	Dec. 2017
Generation of Cell-Like Color Animation by Inverse Iris Filter	ICIC Express Letters	Jan. 2018
ストライプアート動画の生成	産業応用工学会論文誌	Mar. 2018