領域分割スムージングフィルタを用いた波線画像の生成*

平岡 透†

Generation of Ripple Images by Smoothing Filters
Using Different Regions*

Toru HIRAOKA†

We propose a non-photorealistic rendering method for generating ripple images from gray-scale photographic images. Ripple images represent photographic images with wavy lines in a certain direction. Our method is executed by an iterative process with two smoothing filters using different regions in the window. Our method has two features. The first feature is that ripple patterns can be automatically generated according to the density and contour of photographic images. The second feature is that the interval and direction of ripple patterns can be changed by changing the values of the parameters of our method. In order to verify the effectiveness of our method, we conducted experiments using various photographic images. Experimental results show that our method can realize the two features.

1. はじめに

写真画像や3次元データを鉛筆画や水彩画、油絵などの非写実的な画像に変換するノンフォトリアリスティックレンダリングという技術がある[1–3]。近年では、幾何学的な模様で表現された非写実的な画像を生成するノンフォトリアリスティックレンダリングの方法も数多く提案されている。たとえば、幾何学的な模様で表現された非写実的な画像として、迷路画像[4]やオブアート画像[5,6]、反応拡散模様画像[7]、ストライプアート画像[8]、ストライプパッチワーク画像[9]、波線画像[10]などがある。このような非写実的な画像は、ウェブサイトやSNS (Social Networking Service) で活用することができる。たとえば、面白味のある画像として使用したり、個人情報の観点から個人や場所を特定しにくくしたりすることができる。

本論文では、幾何学的な模様で表現された非写実的な

画像として波線画像を取り上げ、写真画像から波線画像を生成するノンフォトリアリスティックレンダリングの方法を提案する。波線画像は、写真画像をある一定方向の波線で表現したものである。波線画像を生成する従来法[10]では、逆ソーベルフィルタを用いた処理を実行することで波線画像を生成しているが、波線模様を0, $(1/4)\pi$, $(1/2)\pi$, $(3/4)\pi$ ラジアン方向でしか生成できない。任意の方向に波線模様を生成できる方法を開発すれば、利活用の幅が広がると考えられる。そこで、領域分割スムージングフィルタを用いた繰り返し処理を実行し、パラメータの値を変えることによって波線模様の方向を任意に変えることができる方法を提案する。また、提案法は、パラメータの値を変えることによって波線模様の間隔も変えることができ、写真画像の画素値の変化に応じて自動的に波線模様を生成できるという特徴がある。さらに、提案法は、従来法と比較して、より滑らかな波線模様を生成でき、異なる印象を与える波線画像を生成できる。提案法の有効性を評価するために、さまざまな写真画像を用いた実験を行った。実験の結果、提案法は上述の特徴を実現できることが明らかになった。

2. 提案法

提案法は、グレースケールの写真画像から波線画像を生成する。提案法は、大きく三つのステップで実行される。一つ目のステップでは、ウィンドウ内の異なる二つ

* 原稿受付 2019年6月14日

* 第63回システム制御情報学会研究発表講演会にて発表 (2019年5月)

† 長崎県立大学 情報システム学部 情報システム学科
Department of Information Systems, Faculty of Information Systems, University of Nagasaki; 1-1-1, Manabino, Nagayo-chou, Nishisonogi-gun, Nagasaki 851-2195, JAPAN

Key Words: non-photorealistic rendering, ripple image, smoothing filter, direction control.

の領域に対して処理を行うスムージングフィルタ（領域分割スムージングフィルタ）を用いて二枚の画像を作成する．二つ目のステップでは，スムージングされた二枚の画像の差を計算する．三つ目のステップでは，二枚の画像の差を用いて写真画像を変換する．これら三つのステップを繰り返すことで波線画像が生成される．提案法のフローチャートを Fig. 1 に示し，提案法の詳細を以下に示す．

Step 0 グレースケールの写真画像の空間座標 (i, j) の画素値を $f_{i,j}$ とする．繰り返し回数 t 番目の画像の画素値を $f_{i,j}^{(t)}$ とする．ここで， $f_{i,j}^{(1)} = f_{i,j}$ とする．また，画素値 $f_{i,j}^{(t)}$ は 0 から 255 の 256 階調の値をもつものとする．

Step 1 画素値 $f_{i,j}^{(t)}$ をサイズ W のウィンドウ内の異なる二つの領域でそれぞれスムージングを行う．スムージングは，各領域内の画素値の平均を計算する．二つの領域は，Fig. 2 に示すようにウィンドウの中心を通る傾きが θ ラジアン直線によって分割される．傾き θ の直線上の画素は，二つの領域に含まれるものとする．二つの領域でスムージングされた画素値をそれぞれ $s_{1,i,j}^{(t)}$ と $s_{2,i,j}^{(t)}$ とする．

Step 2 二つの画素値 $s_{1,i,j}^{(t)}$ と $s_{2,i,j}^{(t)}$ の差を計算して， $d_{i,j}^{(t)}$

とする．

$$d_{i,j}^{(t)} = s_{2,i,j}^{(t)} - s_{1,i,j}^{(t)} \quad (1)$$

Step 3 画素値の差 $d_{i,j}^{(t)}$ に写真画像の画素値 $f_{i,j}$ を加えて， $f_{i,j}^{(t+1)}$ とする．

$$f_{i,j}^{(t+1)} = f_{i,j} + d_{i,j}^{(t)} \quad (2)$$

ここで，画素値 $f_{i,j}^{(t+1)}$ が 0 より小さければ，画素値 $f_{i,j}^{(t+1)}$ を 0 とする．また，画素値 $f_{i,j}^{(t+1)}$ が 255 より大きければ，画素値 $f_{i,j}^{(t+1)}$ を 255 とする．

Step 1 から Step 3 の処理を T 回繰り返すことで，波線画像が生成される．

3. 実験

本論文では，三つの実験を行った．一つ目の実験では，Fig. 3 に示すレナの画像を用いて，提案法のパラメータの値を変えて生成される波線画像を視覚的に確認した．二つ目の実験では，レナの画像以外のさまざまな写真画像を用いて生成される波線画像を視覚的に確認した．三つ目の実験では，提案法と従来法 [10] の波線画像を視覚的に比較した．以下の実験で使用した写真画像は，画像処理の研究でよく用いられる標準画像データベース SIDBA (Standard Image Database)[11] から取り上げたものである．また，以下の実験で使用した写真画像の大きさは， 512×512 画素である．

Fig. 3 Lenna image

3.1 パラメータの値を変えた実験

繰り返し回数 T の値を変えてレナの画像から生成される波線画像を視覚的に確認した．繰り返し回数 T の値は，10, 20, 50, 100 と変えて実験を行った．このとき，パラメータ W と θ の値は，それぞれ 5 と 0 とした．この実験で生成された波線画像を Fig. 4 に示す．Fig. 4 より，繰り返し回数 T の値が大きくなるにつれて，波線模様が鮮明になる様子がわかる．繰り返し回数 T は，100 前後の値を設定すればよいと考える．

ウィンドウサイズ W の値を変えてレナの画像から生成される波線画像を視覚的に確認した．ウィンドウサイズ W の値は，5, 7, 9, 11 と変えて実験を行った．このとき，パラメータ T と θ の値は，それぞれ 100 と 0 とした．この実験で生成された波線画像を Fig. 5 に示す．Fig. 5 より，ウィンドウサイズ W の値が大きいくほど，波線模様の間隔が大きくなることがわかった．ウィンドウサイズ W の値は，ユーザの利用用途に応じて設定す

Fig. 1 Flow chart of our method

Fig. 2 Conceptual diagram of two regions in the window

Fig. 4 Arbitrarily-oriented ripple images in the case of the iteration number $T = 10, 20, 50$, and 100

Fig. 5 Arbitrarily-oriented ripple images in the case of the window size $W = 5, 7, 9$, and 11

ばよいと考える。

傾き θ の値を変えてレナの画像から生成される波線画像を視覚的に確認した。傾き θ の値は、 $(1/6)\pi$, $(1/3)\pi$, $(1/2)\pi$, $(2/3)\pi$ と変えて実験を行った。このとき、パラメータ T と W の値は、それぞれ 100 と 5 とした。この実験で生成された波線画像を Fig. 6 に示す。Fig. 6 よ

Fig. 6 Arbitrarily-oriented ripple images in the case of the slope $\theta = (1/6)\pi, (1/3)\pi, (1/2)\pi$, and $(2/3)\pi$

Fig. 7 Various photographic images

り、傾き θ の値を変えることによって、波線模様の方角を変えることができることがわかった。このとき、波線模様は傾き θ に沿って生成されることを確認した。傾き θ の値は、ユーザの利用用途に応じて設定すればよいと考える。

3.2 さまざまな写真画像を用いた実験

提案法を Fig. 7 に示す 4 枚の写真画像に適用した。4 枚の写真画像について、ウェブサイトや SNS では人や乗り物が多くアップロードされていることから、SIDBA の中から人や乗り物を取り上げた。このとき、傾き θ の値を 0 , $(1/6)\pi$, $(1/3)\pi$, $(1/2)\pi$ と変えて実験を行った。また、パラメータ T と W の値は、それぞれ 100 と 5 とした。この実験で生成された波線画像を Fig. 8 から Fig. 11 に示す。Fig. 8 から Fig. 11 より、すべての写真画像において、波線模様を自動的に生成できることがわかった。また、傾き θ の値を変えることで任意の方向に波線模様を自動的に生成できることもわかった。

3.3 従来法の波線画像との比較実験

提案法の波線画像 (Fig. 5, Fig. 6) と従来法 [10] の波線画像 (Fig. 12) を視覚的に比較した。Fig. 12 は、レナの画像から従来法によって生成された 0 , $(1/4)\pi$,

Fig. 8 Arbitrarily-oriented ripple images in the case of the slope $\theta = 0$

Fig. 10 Arbitrarily-oriented ripple images in the case of the slope $\theta = (1/3)\pi$

Fig. 9 Arbitrarily-oriented ripple images in the case of the slope $\theta = (1/6)\pi$

Fig. 11 Arbitrarily-oriented ripple images in the case of the slope $\theta = (1/2)\pi$

$(1/2)\pi$, $(3/4)\pi$ ラジアンの方角に波線模様を発生させた波線画像である。Fig. 5, Fig. 6, Fig. 12 より、提案法は従来法よりも滑らかな波線模様を生成でき、提案法と従来法によって生成される波線画像は異なる印象を与えることができることがわかった。また、従来法は 0 , $(1/4)\pi$, $(1/2)\pi$, $(3/4)\pi$ ラジアンの方角にしか波線模様を生成できないが、提案法は任意の方角に波線模様を生成できることもわかった。

4. おわりに

本論文では、幾何学的な模様で表現された非写実的な画像として波線画像を取り上げ、写真画像から波線画像を生成するノンフォトリアリスティックレンダリングの方法を提案した。提案法は、領域分割スムージングフィルタを用いた繰り返し処理によって実行した。提案法の有効性を検証するために、さまざまな写真画像を用いた実験を行った。実験の結果、写真画像の画素値の変化に応じて自動的に波線模様を生成でき、また提案法のパラメータの値を変えることによって波線模様の間隔や方向

Fig. 12 Ripple images generated by the conventional method

を任意に変えることができることがわかった。

今後の課題は、提案法を拡張してカラーの写真画像や3次元データに適用することである。

謝 辞

本研究は、JSPS 科研費 JP19K12664 の助成を受けたものです。

参 考 文 献

- [1] P. Haeberli: Paint by Numbers: Abstract image representations; *ACM SIGGRAPH Computer Graphics*, Vol. 24, No. 4, pp. 207–214 (1990)
- [2] J. Lansdown and S. Schofield: Expressive rendering: a review of nonphotorealistic techniques; *IEEE Computer Graphics and Applications*, Vol. 15, No. 3, pp. 29–37 (1995)
- [3] D. Martin, G. Arroyo, A. Rodriguez and T. Isenberg:

A survey of digital stippling; *Computers & Graphics*, Vol. 67, pp. 24–44 (2017)

- [4] K. Inoue and K. Urahama: Halftoning with minimum spanning trees and its application to maze-like images; *ELSEVIER Computers & Graphics*, Vol. 33, No. 5, pp. 638–647 (2009)
- [5] T. C. Inglis, S. Inglis and C. S. Kaplan: Op art rendering with lines and curves; *ELSEVIER Computers & Graphics*, Vol. 36, No. 6, pp. 607–621 (2012)
- [6] A. G. M. Ahmed: Line-based rendering with Truchet-like tiles; *CAe'14 Proceedings of the Workshop on Computational Aesthetics*, pp. 41–51 (2014)
- [7] M. T. Chi, W. C. Liu and S. H. Hsu: Image stylization using anisotropic reaction diffusion; *The Visual Computer*, Vol. 32, No. 12, pp. 1549–1561 (2016)
- [8] 平岡, 浦浜: ストライプアート画像生成の一手法; 映像情報メディア学会誌, Vol. 70, No. 12, pp. J281–J283 (2016)
- [9] T. Hiraoka, H. Nonaka and Y. Tsurunari: A method for controlling patterns of stripe-patchwork images; *ICIC Express Letters*, Vol. 13, No. 3, pp. 175–180 (2019)
- [10] 平岡, 浦浜: 逆ソーベルフィルタによる波紋風画像の生成; 画像電子学会誌, Vol. 44, No. 2, pp. 349–352 (2015)
- [11] University of Southern California: The USC-SIPI Image Database; <http://sipi.usc.edu/database/> (2019)

著 者 略 歴

ひら
岡

とある
透

(正会員)

1997年3月九州芸術工科大学大学院芸術工学研究科情報伝達専攻博士前期課程修了, 2005年3月九州芸術工科大学大学院芸術工学研究科情報伝達専攻博士後期課程修了。2011年11月大分工業高等専門学校情報システム学科准教授, 2016年4月長崎県立大学情報システム学部情報システム学科教授となり現在に至る。画像処理, 地理情報処理, 地域活性化の研究に従事。映像情報メディア学会, 日本写真測量学会, 日本建築学会などの会員。