# IIEEJ 論文

# LIC 法を用いた鉛筆画風動画生成

畠 康高<sup>†</sup> 豊浦 正広<sup>†</sup> 茅 暁陽<sup>†</sup> (正会員)

†山梨大学

# LIC Pencil Drawing Filter for Video

Michitaka HATA<sup>†</sup>, Masahiro TOYOURA<sup>†</sup>, Xiaoyang MAO<sup>†</sup> (Member)

†University of Yamanashi

〈あらまし〉鉛筆画は、モノトーンを用いながら豊かな表現が可能であり、ラフなスケッチとしても完成度の高い芸術作品としても多くの人に親しまれている。我々はビデオを鉛筆画風動画に自動変換する手法を提案する。静止画を自動的に鉛筆画に変換できる LIC 法を用いた鉛筆画生成法を用いる。しかし、静止画の鉛筆画変換手法を動画の各フレームにそのまま適用すると、ちらつきやシャワードア効果が発生する。この問題を解決するために、フレーム間で相関を保ったストロークを生成する新しい技術を提案する。前フレームのストロークを基に、輝度の変化に合わせて、不足するあるいは過剰なストロークのみを、追加または削除し、オブジェクトの移動に合わせてストロークの位置を移流することで課題を解決する。

キーワード:絵画風動画,鉛筆画, LIC法,時間的エイリアシング

Summary> Despite of using monotone only, pencil drawing can present rich expressions, and is an attractive either as an expressive sketching or as finished fine art. We propose a new technique for processing an arbitrary video clip into a pencil drawing like animation. Our technique is an extension to the existing method that converts an image into a pencil drawing using LIC. When converting a video to a pencil drawing style movie, shower door effect and flickering occurs. In order to solve the problem, it is necessary to generate a stroke which maintains the coherence across frames. In this paper, we propose a new method for automatically generating a pencil drawing style movies from a videos using LIC. We reduce the flickering by reusing the strokes of the previous frame and add to or remove from it a minimum number of strokes for representing the tone of current frame. The shower door artifact is reduced through the advection of the strokes and paper texture along the optical flow.

Keywords: painterly filter, painterly video, pencil drawing, LIC, temporal aliasing

# 1. はじめに

コンピュータグラフィックス分野において、水彩画や油絵などの絵画風画像の生成をはじめとするノンフォトリアリスティックレンダリング(NPR)表現技法が多く提案され、近年アニメーションや映画をはじめ、様々な映像製作現場で利用されている。NPR表現を用いることで、美的で芸術的な画像・映像を創り出すことができ、また、抽象化された画像・映像によって実写よりも効果的に情報を伝えられるとされている。中でも本研究が取り扱う鉛筆画風の表現は、粗い描写も繊細な描写もできるため、高い芸術性と抽象性を持つ表現が可能である。その表現力によって手描きの温かみを想起させた例として、2013 年第53 回アヌシー国際アニメーション

フェスティバルで審査員特別賞を受賞した Anna Budanova 監督の鉛筆画と水彩画風の表現を用いた"Obida"がある.

本論文では鉛筆画風動画を生成する新しい方法を提案する. 提案手法では、3 次元モデルを必要とせず、ビデオやアニメ といった一般的な2次元動画を鉛筆画風に自動変換できるた め、専門家以外の利用者にも提供できる.

提案手法はもっとも代表的な鉛筆画生成法の一つである LIC 鉛筆画生成法を動画用に向けて拡張する. LIC (Line Integral Convolution) はテクスチャベース流れ場可視化技術であり、ホワイトノイズを流れの方向に沿ってぼかすことで流れ場を可視化する <sup>1)</sup>. LIC 鉛筆画生成法は入力画像にランダムディザ法を適用して生成するノイズとストロークの方向を表すベクトル場に LIC を適用することで入力画像を鉛筆画に

変換する. 自動生成が可能であるため, 動画への応用が期待 される. しかし,これまで様々な拡張<sup>2)-8)</sup>が行われたにも関 わらず、LIC を用いた効果的な鉛筆画動画生成法はまだ提案 されていない. LIC 鉛筆画生成法はノイズを入力とすること に加えて、高周波成分を含む紙テクスチャも利用するため、 時間的エイリアシングが特に生じやすい. また, 手前にある 模様の奥で物が動いているように感じるシャワードア効果を 抑えるためには、移動するオブジェクトに付随してストロー クを移動させる必要がある. しかし, LIC 鉛筆画法では, ス トロークを明示的にモデリングしていないため、従来のブラ シストロークの位置を3次元オブジェクトに固定する方法9 や前後のフレームに置いたブラシストロークの位置を対応づ けさせる方法 <sup>10,11)</sup>をこのまま利用することはできない. これ らの問題を解決するために、本論文では、オプティカルフロ ーに基づいてストロークの疑似表現のベースとなるノイズを 移流させ, 当該入力フレームに対する輝度の過不足に応じて, 移流後の前フレームのノイズを変更する方法を提案する.

以下,第2章では関連研究をレビューする.第3章では, 簡単に既存のLICベース鉛筆画生成技術を紹介する.第4章 では,フレーム間の相関に関する基本的な考え方と,新たに 提案された手法とその詳細を説明する.第5章では,いくつ かの結果を示した後,結論を述べる.

# 2. 関連研究

本章では、鉛筆画風画像の生成と、NPR 動画生成の関連研究を説明する.

# 2.1 鉛筆画風画像の生成

鉛筆画風画像の生成には、黒鉛や紙などの画材や、鉛筆画 を作成する画家の動作の観察によってモデルを獲得し、この 結果を対話型システムや,3次元モデルに適応しレンダリン グによって結果を得る手法が提案されている. Sousa と Buchanan は走査型電子顕微鏡で本物の鉛筆画を調べることに より,鉛筆画をシミュレーションしたモデルを提案した12,13). Takagi らは紙の微細構造と顔料分布を3次元ボリュームデー タでモデル化しボリュームレンダリングにより色鉛筆画画像 の生成に成功した <sup>14)</sup>. Semet らは Artificial Ant システムを利 用し、ユーザーにより良い画像を対話的に選択していくこと で鉛筆画を生成する方法を示した <sup>15)</sup>. Melikhov らは Disk B-Spline Curves に基づいて鉛筆画へ変換する方法を提示し た <sup>16</sup>. Lee らは鉛筆画風のトーンをレンダリングするための GPUベースのリアルタイムレンダリングする技術を開発した17. 最近では AlMera らが、ストロークの入力点と人間の腕の動 きの物理的な計算モデルから, 人間が描いた鉛筆のストロー クに似ている線を描画するアルゴリズムを設計した <sup>18)</sup>. これ らの手法によって実際の画材の特性やストロークの形状など を再現できるが、個別のモデル獲得のためには特殊な装置が 必要になり、対話的作業が必要なため利用者の技量や時間が

必要になる. また、3次元モデルを準備する必要がある.

より利用しやすい方法として、2次元の入力画像に何らかの画像処理を施し、鉛筆で描いたような画像に変換する絵画的フィルタリング手法が提案されている。本提案ではこのアプローチを採用し、動画を自動的に鉛筆画風に変換することを目指す。

最も代表的な鉛筆画フィルタとして, LIC (Line Integral Convolution) を利用する方法がある. LIC は本来, ベクトル 場に沿ってホワイトノイズをぼかすことによって流れの軌跡 を可視化する流場可視化法である<sup>1)</sup>. Mao らは LIC で可視化 される流れの軌跡が鉛筆のストロークと似ていることに着目 し,2次元画像を入力としたLIC 鉛筆画生成法を提案した<sup>2)</sup>. Mao らの方法に対してこれまで様々な拡張が行われた 3)~8). Yamamoto らによる色鉛筆画に拡張する研究では、いくつか の色のストロークを重ねて描き, 紙面上で光学的に混色する 効果を再現した<sup>3)</sup>. Chen らは各領域の特徴に適応したストロ ークを描画することに成功した<sup>4)</sup>. 最近では, Kwon らは明る さによって層を分けてストロークを描き、ストロークを重ね 合わせたトーンを再現した<sup>5)</sup>. 更に, Yang らは, ETF (Edge Tangent Flow)を用いた輪郭部分のストロークと、クロスハッ チングを用いたトーンで鉛筆画を再現した<sup>6</sup>. また, Yang ら は, SBL (Swing Bilateral LIC) filter を提案し, 境界やハイラ イトといった部分を明確に描いた色鉛筆を生成した<sup>7)</sup>. Hata らは、画像の顕著性を示す Saliency Map を用いて、人が描い たような, 注視点にアクセントがある鉛筆画を作成する方法 を提案した<sup>8</sup>. しかし、これらの方法はすべて1枚の静止画 像に対する変換手法であり,動画を対象としたものではなか った. 本研究では既存の LIC 鉛筆画フィルタを拡張し、鉛筆 画風動画を自動生成できる新しい技術を提案する.

## 2.2 NPR 動画の生成

鉛筆画以外の絵画風変換まで視野を広げると,動画に対する NPR 手法も提案されている.動画変換の際に新たに問題となるのは、生成結果に時間方向の相関を持たせなければ、ちらつきが起こることである.従来の絵画風動画生成の提案では、特にフレーム間のストロークに相関をもたせることが課題として捉えられてきた.

これらの課題にアプローチした絵画風動画の研究を挙げる. Barbara と Meier は 3 次元モデルを用いた油絵風の動画の生成法を提案した 9.3 次元モデル上にストロークを描くための位置を固定することで,ストロークに相関を得ている. Hertzmann らは油絵風の動画生成法を提案した 10. 前フレームの結果をベースに,前後のフレームで違いのある部分にブラシストロークを追加することでちらつきを防止した. Litwinowicz らは印象派風の動画生成法を提案した 11. オプティカルフローを用いてストローク位置を移動しつつ,ドロネー三角形分割を用いて新たにストローク位置を追加し,相関のあるストロークを実現した。これらの手法では筆の一途り

に対応するブラシストロークを1本ずつ明示的描画するため、フレーム間でストロークの位置の対応付けが可能である.ストロークを明示的に扱わず、画像に対するフィルタリング処理によってストロークを疑似的に表現するLIC鉛筆画生成法において、これらの手法を直接適用することはできない.

LIC 法は、相関のないホワイトノイズを入力とするため、原理的に時間的な相関を保証することが特に難しい。また、より鉛筆画らしさを表現するために高周波成分を含む紙のテクスチャも利用するため、時間的なエイリアシングが更に目立ちやすい。Yamamoto らはこれまでに動画に対する拡張を試みており<sup>19)</sup>、ストロークを2つのレイヤに分け、ベースレイヤにおいては方向性のあるストロークを表現せずホワイトノイズにガウシアンフィルタを用いたローパスフィルタを施し、トップレイヤにおいては、フレーム間でストロークに沿ってローパスフィルタを掛けることでちらつき抑える方法を提案した。しかし、ストロークの位置には相関がないために、効果は限定的であった。

#### 3. LIC 鉛筆画生成法

本研究では図1に示すMaoらのLIC法を用いた鉛筆画風画像生成法を基にする. Maoらの鉛筆画風画像生成法は、以下の手順によって鉛筆画風画像を生成する.

- Step 1. 入力画像からグレースケール画像を作成する.
- Step 2. グレースケール画像からエッジを作成する.
- Step 3. グレースケール画像をランダムディザ法で2値化し、 ノイズ画像を生成する.
- Step 4. グレースケール画像からテクスチャの方向を検出し、ベクトル場を生成する.
- Step 5. ベクトルとノイズ画像に LIC 法を適応し, LIC 画像 を作成する.
- Step 6. LIC 画像とエッジ画像を合成し、ストローク画像を生成する.
- Step 7. ストローク画像に紙テクスチャを合成し、結果の鉛 筆画風画像とする.

鉛筆画風動画生成は、原理的に動画の各フレームを**図 2(a)** に示すように入力画像として取り出し、フレーム毎に先述の鉛筆画生成法を適応することにより実現可能である.しかし、このままではフレーム間の相関に関わる 2 つの問題により不

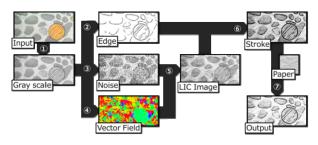


図1 LIC法を用いた鉛筆画生成法

Fig. 1 Pencil drawing generation using LIC

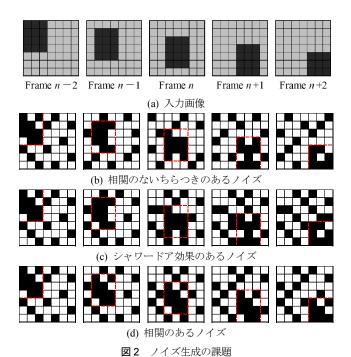


Fig. 2 The problem of noise generation

自然な動画が生成される.

一つ目の問題は"ちらつき"である。毎回ランダムディザ法でノイズを作ると、前後のフレームが全く同じだとしても、フレーム間の黒画素の位置に相関がないノイズ画像が生成される。その為、黒画素が基となって生成されるストロークの位置も毎回変わる。同じ位置の画素が各フレームでランダムに白に黒にと点滅し(図 2(b))、ちらつきとして知覚される。これを解決するためには前後のフレームにおいて同じ位置の画素値をできるだけ反転させない必要がある。

2つ目の問題は"シャワードア効果"の発生である。"シャワードア効果"とは、型板ガラスを通してシャワールームの向こう側の影を見るような、オブジェクトの移動と相関がないトーンを感じる現象である。フレーム間での黒画素の位置を保ったノイズ画像を使用すると、入力画像のオブジェクトの移動に伴ったストロークの基となる黒画素が移動しないことになる(図 2(c))。このため、オブジェクトのエッジとトーンが分離して見える。固定されたノイズが型板ガラスのように作用し、シャワードア効果が発生する。シャワードア効果はフレーム間のオブジェクトの位置に相関がないために発生するので、入力画像のオブジェクトの移動に伴い前後のフレームのノイズ画像での対応する画素を移動する必要がある。

従って、これらの課題を解決するためには、オブジェクトの移動に従ってノイズを移流し、前後のフレームで生成するノイズ画像の黒画素の位置を保ちつつ、輝度の変化に合わせた必要最低限の黒画素の追加と削除を行う必要がある(図 2(d)).

# 4. 提案手法

#### 4.1 概要

提案手法ではMaoらの鉛筆画生成法を各フレームに対して 実行する. その際、相関のある動画を作成するため、前フレ ームと当該フレーム間のオプティカルフローを算出し用いる (図3).

- Step 1. 当該フレームの入力画像からグレースケール画像を 作成する.
- Step 2. グレースケール画像からエッジを作成する.
- Step 3. グレースケール画像,前フレームのノイズ画像及び 前フレームと当該フレーム間のオプティカルフロー を用いてノイズ画像を生成する.
- Step 4. ベクトル場を生成する.
- Step 5.
   ベクトル場とノイズ画像に LIC 法を適応し, LIC 画像を作成する.
- Step 6. LIC 画像とエッジ画像を合成しストローク画像を生成する.
- Step 7. 前フレームと当該フレーム間のオプティカルフロー を考慮した紙テクスチャを生成する.
- Step 8. ストローク画像に紙テクスチャを合成し、当該フレームの鉛筆画画像となる.

第3章で述べた通り、相関のある鉛筆画風動画を生成する上で最も重要なのはフレーム間で相関のあるノイズ画像を生成することである。そのため、まずは、前フレームとオブジェクト間で相関がある、当該フレームのノイズ画像を生成する。入力画像の前フレームと当該フレームから当該フレームの各画素が前フレームのどの位置から移動したかを示すオプティカルフローを算出する。このオプティカルフローに従い前フレームのノイズ画像の各画素を移流する。オブジェクトの移動がオプティカルフローによって表されているので、前フレームとオブジェクト間に相関がある当該フレームのノイズ画像を得ることができる。

次に移流したノイズ画像に対して、黒画素をなるべく保存 しながら、前フレームと黒画素の位置に相関があるように修

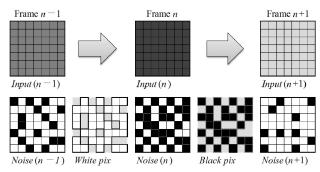


図4 ノイズの追加方法

Fig. 4 The additional method of a noise

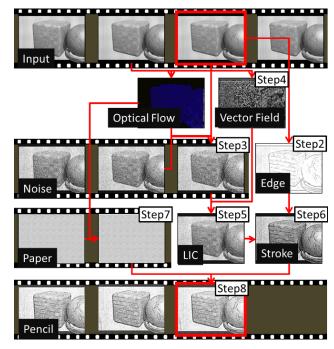


図3 LIC 法を用いた鉛筆画風動画生成法

Fig. 3 Pencil drawing movie generation using LIC

正する. LIC 鉛筆画生成法では、ランダムディザ法によって、各画素の輝度にふさわしい確率で黒画素を発生させており、言い換えると、入力画像の輝度を表すのにふさわしい数の黒画素を生成している。そこで、当該フレームの入力画像の輝度と移流したノイズ画像から、当該フレームの輝度を表現するのに、不足する、あるいは、過剰な黒画素の数を算出し、移流したノイズ画像に対して黒画素の追加・削除を行う。過不足分として追加・削除した黒画素以外は前フレームのノイズ画像と同じであるため、相関のあるノイズ画像を生成することができる.

図 4 に示すように、前フレームより暗くなる時 (Input(n-1)  $\rightarrow$  Input(n) は、(移流した後の)前フレームの白画素 (White pix) から必要な数の画素をランダムに選択し、黒画素に変更する. 逆に前フレームより明るくなる時 (Input(n) $\rightarrow$  Input(n+1)) は、(移流した後の)前フレームの黒画素 (Black pix) から必要な数の画素をランダムに選択し、白画素に変更することによって、入力画像の輝度毎の画素に対してふさわしい数量だけ黒画素を追加・削除する.

# 4.2 エッジ画像の生成(4.1 節 Step 2)

差分フィルタよりもノイズに強いため DoG (Difference of Gaussian) フィルタは NPR で良く用いられる. 本提案においても DoG フィルタによりエッジを検出する.

# ・エッジのちらつきの防止

フレームの前後でのエッジの検出の有無によっても、ちらつきが発生する。そこで、フレームnのエッジ画像 $Edge_n$ の

同位置にある前フレームのエッジ画像  $Edge_{n-1}$  または次のフレームのエッジ画像  $Edge_{n+1}$  の画素でエッジの有無が異なる場合,ちらつきと判断しエッジの有無を反転させる.この際,オブジェクトの移動を考慮するため, $Edge_{n-1}$  と  $Edge_{n+1}$  はオプティカルフローに従って移動してから反転処理を行う.

#### エッジのストローク化

ちらつきをさらに抑えるために、また、エッジの見た目を 鉛筆のストロークに近づけさせるために、エッジに沿ってLIC を用いてローパスフィルタを掛ける.

## 4.3 ノイズ画像生成アルゴリズム(4.1 節 Step 3)

相関のある鉛筆画風動画を作成する為に、当該フレームの入力画像 (Input(n)) の他に前フレームの入力画像 (Input(n-1)) と前フレームのノイズ画像 (Noise(n-1)) を基に相関のあるノイズ画像を生成する.以下にその手順を示す(図 5).

- Step 0. フレーム 0 のノイズ画像 Noise(0,p)はランダムディザ法を用いて生成する. ただし, p は画素とする.
- Step 1. フレーム n-1 とフレーム n の入力画像を用いてオプティカルフローを計算し、フレーム n の各画素がフレーム n-1 のどの画素から移動したかを求める.
- Step 2. Noise(n-1,p)を Step 1. で求めたオプティカルフローで移流しフレームnのノイズ画像 OptNoise(n,p)を計算する.
- Step 3. *OptNoise(n,p)*の黒画素をなるべく保存しながら,入 力画像の輝度に合わせ黒画素の追加・削除を行い, ノイズ画像 AddedNoise(n,p)を作成する.
- Step 4. 入力画像と AddedNoise(n,p)との誤差が多い個所を修正したノイズ画像 Noise(n,p) を生成する.

Step 1 と Step 2 によって前フレームと当該フレームとのオブジェクト間で、相関があるノイズ画像(OptNoise(n,p))を得て、Step 3 にて前フレームのノイズ画像の黒画素の位置と相関がある当該フレームのノイズ画像(AddedNoise(n,p))を得る。Step 4 にて、Step 3 で得たノイズ画像(AddedNoise(n,p))と入力画像を比べ大きい誤差を修正したノイズ画像

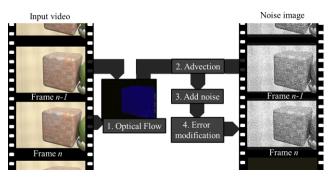


図5 相関のあるノイズ画像生成法

Fig. 5 A noise image generation method with coherence

(Noise(n,p)) を得る.

# AddedNoise(n,p)の生成について

入力画像 Input(n-1)の画素数を輝度 l 毎に集計し NumPix(l) を得る。また、オプティカルフローによって画素を移動した ノイズ画像 OptNoise(n)の黒画素の数を入力画像 Input(n-1) の画素の輝度 l 毎に集計し NumBlackPix(l) とする(図 6(a) 参照). 入力画像の輝度 l は 0~255(黒~白)とする.

入力画像の輝度 l から NumPix(l)に本来存在すべき黒画素の個数 NeedBlackPix(l)を求める。存在すべき黒画素と現在の黒画素の差分が,追加あるいは削除すべき黒画素の数 AddBlackPix(l)である(図 6(b)参照).

例えば、灰色 (l=127) の場合、灰色画素の数 NumPix(127) の半分がノイズ画像における黒画素であるべきであり、現在存在する黒画素の数 NumBlackPix(127)とあるべき黒画素 NeedBlackPix(127)との差分が、入力画像の灰色画素 (l=127) に対するノイズ画像に追加あるいは削除すべき黒画素の数 AddBlackPix(127)となる.

# $\textbf{Algorithm 1} \ NumBlackPix(l)$

```
1: for each pixel p do

2: l = input(n, p)

3: NumPix(l) = NumPix(l) + 1

4: if OptNoise(n, p) = 0 then

5: NumBlackPix(l) = NumBlackPix(l) + 1

6: end if

7: end for
```

# (a) Algorithm 1

# **Algorithm 2** AddBlackPix(l)

```
1: for each intensity level l do

2: NeedBlackPix(l) = (1 - l/255) * NumPix(l)

3: AddBlackPix(l) = NeedBlackPix(l) - NumBlackPix(l)

4: end for
```

# (b) Algorithm 2

#### **Algorithm 3** AddNoise(n, p)

```
1: for each pixel p do
     l = input(n, p)
     if AddBlackPix(l) > 0
     and OptNoise(n) = 255 and
     Random() < Abs(AddBlackPix(l))/(NumPix(l) -
     NumBlackPix(l)) then
4:
        AddedNoise(n, p) = 0
     else if AddBlackPix(l) < 0
     and OptNoise(n) = 0 and
     Random() < Abs(AddBlackPix(l))/NumBlackPix(l)
6:
        AddedNoise(n,p) = 255
7:
     else
8:
        AddedNoise(n, p) = OptNoise(n, p)
9:
     end if
10: end for
```

# (c) Algorithm 3

## 図6 提示手法のアルゴリズム

Fig. 6 Algorithms in the proposed method

追加すべき黒画素がある(AddBlackPix(I)>0)とは、対象とする画素が前フレームと比べ暗くなったということである. この場合、すでに存在する黒画素はそのまま残し、現在、白画素となっている画素(OptNoise(n,p)=255)のみに対して、AddBlackPix(I)の数分ランダムに黒画素を発生させる.

逆に、削除すべき黒画素がある(AddBlackPix(I)<0)とは、対象とする画素が前フレームと比べ明るくなったということである。この場合は、すでにある白画素をそのまま残し、現在、黒画素となっている画素(OptNoise(n,p)=0)に対してだけ、AddBlackPix(I)の数分ランダムに黒画素を消していく。

これによって、足りない黒画素を必要分だけ追加することができ、前フレームのノイズ画像 Noise(n-1)の黒画素や白画素の位置と、移流したノイズ画像 OptNoise(n)の黒画素や白画素の位置に相関が維持できる(図 6(c)参照).

図 6(c)では、乱数を用いて黒画素の追加と削除を行っているので、厳密には AddBlackPix(I)分の数だけ黒画素を作り出してはいない、フレーム 0 のランダムディザ法によるノイズ生成と同様に、輝度から求めた閾値と乱数による黒画素の生成で濃淡を十分に表すことができるので、処理速度を考慮し、厳密な AddBlackPix(I)分の黒画素を生成していない。

## ・入力画像とノイズ画像との誤差の修正

入力画像 *input(n,p)* と黒画素を追加・削除した *AddedNoise(n,p)*をそれぞれ平滑化し、これらの画像の各画素の差分を計算する。入力画像に対して誤差が大きい位置の画素に対して、対応する当該フレームの入力画像の輝度に従い、ランダムディザ法を用いてノイズを再生成する。これにより入力画像 *input(n,p)*に対して誤差の大きい部分を修正したノイズ画像 *Noise(n,p)*が作成される。

このノイズ画像Noise(n,p)を鉛筆画風動画生成においてLIC の入力ノイズ画像として使用し、当該フレームの鉛筆画を生成していく。

## 4.4 紙テクスチャの合成(4.1 節 Step 7)

最後に合成する紙画像に対しても、シャワードア効果を防止するために、移流を行う. 紙画像は紙の構造を表すテクスチャを、ある程度保ちながら移流していく必要がある.

テクスチャの移流は水彩画の動画生成のためにBousseauらが提案した方法 $^{20)}$ を用いる。フレーム0からテクスチャをオプティカルフローに従って移流していくと、だんだんとテクスチャの構造が崩れていく。そこで $\max$ 6 $^{21)}$ は一定フレーム毎に移流をリセットする方法を用いたが、リセット前後のフレームで相関がないためちらつきが発生する。そこで $\max$ 6 $^{20)}$ 0,一定フレーム毎に移流をリセットする方法と、その逆方向へ移流した結果を合成することによって、リセット前後の相関を保ちつつ、構造を保つことに成功した。

ここでは10フレーム毎に移流を行い、フレーム数に応じて 順方向のノイズ画像と逆方向のノイズ画像を線形に合成して いる.

# 5. 実験

#### 5.1 実装

C#と OpenCV のラッパーである Emgu CV を用いている. OpticalFlow にはブロックマッチング法を用いた. ベクトル場の生成にはHata らのガボールフィルタを用いたテクスチャ方向検出法を用いた<sup>8</sup>.

#### 5.2 結果

図7(a)は明るさのみ変化する動画を入力とした結果である. 図 7(b)はオブジェクトの移動を伴う動画に適用した結果である. 図 7(c)は雲が動く動画,図 7(d)は波の動画の結果である.

各図の右側は連続した3フレームの一部分を拡大して並べたものである. 紙画像の合成があると解り難いため、相関の有無を確かめやすいように LIC 画像で示した.

# 5.3 評価方法

見た目以外の評価を行うため、評価対象の当該フレームと前フレームとの差分を、式(1)を用いて計算した。 差が少ないほど フレーム 間での相関が高いといえる.

$$diff_n = \frac{1}{$$
対象画素数  $\sum_{x} \sum_{y} (img_n(x, y) - optimg_{n-1}(x, y))$  (1)

式(1)ではオブジェクトの移動を考慮するため、評価対象の前フレームの画像  $img_{n-1}$  をオプティカルフローで移流した  $optimg_{n-1}$  を用いる。 $optimg_{n-1}$  と評価対象の当該フレームの画像  $img_n$  との 1 画素当たりの差分の平均を計算する。オプティカルフローが検出された画素のみを対象画素とする。この  $diff_n$  の値が大きいほど(誤差が大きく)、すなわち、ちらつきがある画像であるといえる.

図8は式(1)によるフレーム毎のノイズ画像の評価結果のグラフである.表1にビデオの全フレームの平均値を示す.また,図9と表2に提案手法により生成された鉛筆画風動画の

表1 ノイズ画像のフレームあたりの diff,

**Table 1**  $diff_n$  per frame of Noise Image

$diff_n$	Light	Move	Cactus	Waves
入力画像	0.51%	1.65%	0.70%	1.52%
相関無	30.57%	31.26%	41.55%	40.27%
相関有	1.01%	1.83%	8.97%	7.39%

表2 鉛筆画のフレームあたりの  $diff_n$ 

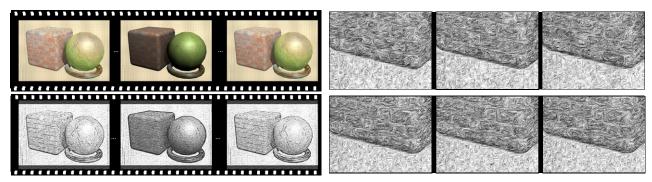
**Table 2** *diff*<sub>n</sub> per frame of Pencil Image

$diff_n$	Light	Move	Cactus	Waves
入力画像	0.51%	1.65%	0.70%	1.52%
相関無	10.72%	12.33%	17.29%	18.38%
相関有	3.39%	6.64%	12.63%	15.92%

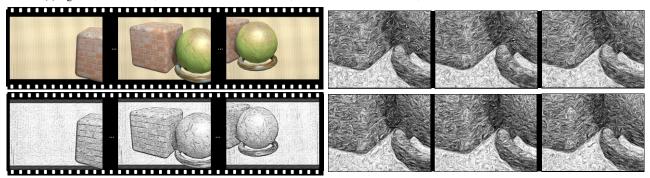
評価結果を示す. どちらも,図7の入力動画と入力動画から 得た本提案を用いた相関のある結果と,単純にフレーム毎に 鉛筆画を生成した相関のない結果を評価対象とした.

#### 5.4 考察

図7の右側の、連続した3フレームのLIC画像を拡大した結果を見ると、下の本提案でのLIC画像では、レンガやサボテンを描いているストロークの位置(トーンの濃淡)が保た



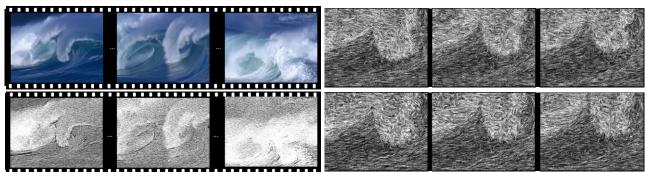
(a) Light (左上:入力画像,左下:相関のある本提案結果,右上:相関のないLIC 画像,右下:相関のある本提案でのLIC 画像)



(b) Move (左上:入力画像,左下:相関のある本提案結果,右上:相関のないLIC画像,右下:相関のある本提案でのLIC画像)



(c) Cactus (左上:入力画像,左下:相関のある本提案結果,右上:相関のないLIC 画像,右下:相関のある本提案でのLIC 画像)



(d) Waves (左上:入力画像, 左下:相関のある本提案結果, 右上:相関のない LIC 画像, 右下:相関のある本提案での LIC 画像) 図 7 結果画像

Fig. 7 The result images

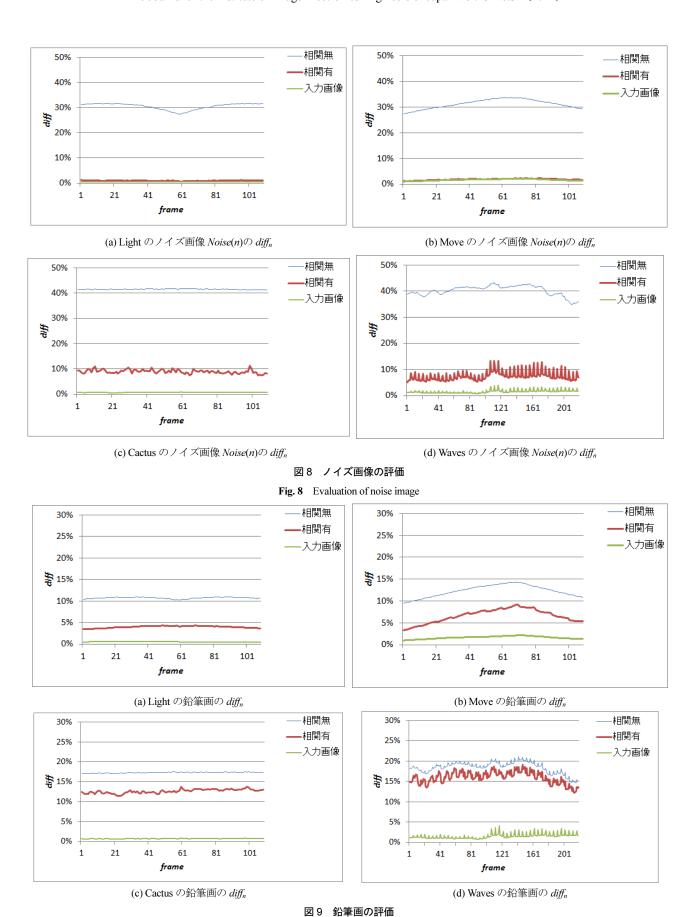


Fig. 9 Evaluation of pencil image

れているのに対し、上の相関のない結果ではフレーム毎にストロークの位置が変わっていることがわかる。図 7(d)は波の動きが激しいため、他の結果と比べてストローク位置の相関が小さい。

また、評価式(1)を用いた結果では、本提案である相関がある鉛筆画風動画の結果は、相関のない方法と比べてフレーム間の画素の差分 diff<sub>n</sub>が小さいことがわかる(図 8、表 1). ノイズ画像では相関がない場合に比べて 5 倍~30 倍の差が見られた. 鉛筆画での結果(図 9、表 2)でも本提案での相関のある結果の方が、より diff<sub>n</sub>が小さくちらつきが少ないといえる. ノイズ画像の評価(図 8、表 1)と比べると、相関がないものはノイズ画像と比べ改善されている. これは LIC によりぼかされたことによってちらつきが緩和されたためと考えられる. 一方、相関のある方法では、ノイズ画像の diff<sub>n</sub>よりも悪くなった. これはストロークの基点となる黒画素の位置では相関は持つが、(Cactus の山肌に顕著であるが)ストロークの方向に相関があまりないためと考えられる.

# 6. おわりに

2 次元動画からの鉛筆画風動画の自動生成法を提案した. その際,前後のフレーム間でストロークの位置の相関を保つことでちらつきとシャワードア効果を抑えた.

今後,ストロークの方向に相関を持たせるために安定したベクトル場を検出する方法を検討する予定である。例えばYang らの方法  $^{n}$ は輪郭線という強い方向性からオブジェクトの方向を得ているためより安定した方向検出が可能であると考えられる。また,人の注目する部分を強調して描く鉛筆画生成法  $^{8}$ の適用も現在検討している。鉛筆画や線画のアニメーションでは動きのある被写体をわざとちらつかせて目立たせる手法をとる場合がある。今後対象物や位置によって相関の強さを制御する方法も検討していく予定である。また Xie らのGPU を用いた鉛筆画の生成法の技術を一部利用し,高速化を行う予定である  $^{20}$ .

# 謝辞

本研究は JSPS 科研費 26240015, 26560006 の助成を受けた ものである.

## 参考文献

- B.Cabral, L.C.Leedom: "Imaging Vector Field Using Line Integral Convolution", Proc. of the Annual Conference on Computer Graphics and Interactive Techniques, pp.263–270 (1993).
- X.Mao, Y.Nagasaka, A.Imamiya: "Automatic Generation of Pencil Drawing from 2D Images Using Line Integral Convolution", Proc. of the International Conference on Computer Aided Design and Computer Graphics, pp.240–248 (2001).
- S.Yamamoto, X.Mao, A.Imamiya: "Colored Pencil Filter with Custom Colors", Computer Graphics and Applications (Proc. of Pacific

- Graphics), pp.329-338 (2004).
- Z.Chen, J.Zhou, X.Gao, L.Li, J.Liu: "A Novel Method for Pencil Drawing Generation in Non-Photo-Realistic Rendering", Advances in Multimedia Information Processing - PCM 2008 Lecture Notes in Computer Science, Vol.5353, pp.931–934 (2008).
- Y.Kwon, K.Min: "Texture-Based Pencil Drawings from Pictures", Communications in Computer and Information Science, Vol.206, pp.70–77 (2011).
- H.Yang, K.Min: "Feature-Guided Convolution for Pencil Rendering", KSII Trans. on Internet and Information Systems, Vol.5, No.7, pp.1311–1328 (2011).
- H.Yang, Y.Kwon, K.Min: "A Stylized Approach for Pencil Drawing from Photographs", Computer Graphics Forum, Vol.31, No.4, pp.1471–1480 (2012).
- M.Hata, M.Toyoura, X.Mao: "Automatic Generation of Accentuated Pencil Drawing with Saliency Map and LIC", The Visual Computer (Proc. of the Computer Graphics International), Vol.28, No.6–8, pp.657–668 (2012).
- B.J.Meier: "Painterly Rendering for Animation", Proc. of the Annual Conference on Computer Graphics and Interactive Techniques, pp.477–484 (1996).
- A.Hertzmann, K.Perlin: "Painterly Rendering for Video and Interaction", Non-Photorealistic Animation and Rendering, pp.7–12 (2000).
- 11) P.Litwinowicz: "Processing Images and Video for an Impressionist Effect", Proc. of the Annual Conference on Computer Graphics and Interactive Techniques, pp.407–414 (1997).
- 12) M.C.Sousa, J.W.Buchanan: "Observational Model of Blenders and Erasers in Computer-Generated Pencil Rendering", Proc. of the Conference on Graphics Interface, Vol.1999, pp.157–166 (1999).
- 13) M.C.Sousa, J.W.Buchanan: "Computer-Generated Graphite Pencil Rendering of 3D Polygonal Models", Computer Graphics Forum, Vol.18 No.3, pp.195–208 (1999).
- 14) S.Takagi, M.Nakajima, I.Fujishiro: "Volumetric Modeling of Colored Pencil Drawing", Proc. of Pacific Conference on Computer Graphics and Applications, pp.250–258 (1999).
- 15) Y.Semet, U.O'Reilly, F.Durand: "An Interactive Artificial ant Approach to Non-Photorealistic Rendering", Genetic and Evolutionary Computation – Lecture Notes in Computer Science, Vol.3102, pp.188–200, (2004).
- 16) K.Melikhov, F.Tian, X.Xie, Hock Soon Seah: "DBSC-Based Pencil Style Simulation for Line Drawings", Proc. of the International Conference on Game Research and Development, pp.17–24 (2006).
- 17) H.Lee, S.Kwon, S.Lee: "Real-Time Pencil Rendering", Proc. of the International Symposium on Non-Photorealistic Animation and Rrendering, pp.37–45 (2006).
- 18) Z.AlMeraj, B. Wyvill, T.Isenberg, A.A.Gooch, R.Guy: "Automatically Mimicking Unique Hand-Drawn Pencil Lines", Computers & Graphics. Vol.33, No.4, pp.496–508 (2009).

- 19) S. Yamamoto, X. Mao, K. Tanii, A. Imamiya: "Enhanced LIC Pencil Filter", Proc. of the International Conference on Computer Graphics, Imaging and Visualization, pp.251–256 (2004).
- 20) A.Bousseau, F.Neyret, J.Thollot, D.Salesin: "Video Watercolorization using Bidirectional Texture Advection", ACM Trans. on Graphics – Proc. of ACM SIGGRAPH, Vol.26, No.3.(2007).
- 21) N.Max, B.Becker: "Flow Visualization Using Moving Textures", Data Visualization Techniques (Editor: Bajaj, C.), John Wiley and Sons Ltd., pp.99–105 (1999).

22) D.Xie, Y.Zhao, D.Xu, X.Yang: "Convolution Filter Based Pencil Drawing and Its Implementation on GPU", Advanced Parallel Processing Technologies - Lecture Notes in Computer Science, Vol.4847, pp.723–732 (2007).

> (2014年2月14日 受付) (2014年4月21日 再受付)



# 畠 康高

2005 年 山梨大学工学部 卒業. 2007 年 同 大学大学院修士課程 修了. 2010 年 同大学 大学院博士課程入学. 現在, コンピュータ グラフィックス, 可視化の研究に従事.



# 豊浦 正広

2003 年 京都大学工学部情報学科 卒業. 2007 年 日本学術振興会特別研究員DC2. 2008 年 京都大学大学院情報学研究科博士後期課程 修了. 2008 年 日本学術振興会特別研究員PD. カリフォルニア大学サンタバーバラ校訪問研究員. 2009 年山メ大学大学院医学工学総合研究部 助教. 拡張現実感,コンピュータビジョンの研究に従事.電子情報通信学会,情報処理学会,IEEE 各会員.博士(情報学).



## **茅 暁陽** (正会員)

1983 年 中国復旦大学 計算機学科卒業. 1987年 東京大学大学院情報科学研究科修士課程 修了. 1990 年 同大学大学院理学研究科博士課程 修了. 1990 年 株式会社クボタコンピュータ応用エンジニア. 1994 年 New York 州立大学 Stony Brook 校 客員研究員. 1995 年 科学技術振興事業団特別研究員. 1996 年 山梨大学工学部 講師. 1997 年同准教授. 1997 年 同大学院医学工学総合研究部 准教授. 2008 年 同教授. 画像処理,コンピュータグラフィックス,可視化の研究に従事. 情報処理学会,画像電子学会,ACM 各会員. 理学博士.