

## LIC 法を利用した鉛筆画の自動生成法

Automatic Generation of Pencil Drawings Using Line Integral Convolution

茅 暁陽, 長坂 好恭, 山本 茂文, 今宮 淳美  
Xiaoyang Mao, Yoshiyasu Nagasaka, Shigefumi Yamamoto, Atsumi Imamiya

山梨大学工学部コンピュータメディア工学科  
Department of Computer and Media Engineering, Yamanashi University

### 概要

本論文は、流れ場の可視化手法として近年注目されている LIC (Line Integral Convolution) 法を利用し、2次元画像を自動的に鉛筆画に変換する手法を提案する。LIC 法では、流線上に定義されているローパスフィルタを用いてホワイトノイズを畳み込む処理を行うため、出力画像には明るさにばらつきをもった線状の軌跡が見える。このような軌跡で鉛筆のストロークを近似し、入力画像のトーンにマッチしたホワイトノイズとストロークの方向を表すベクトル場に LIC 法を適用することで鉛筆画を生成する。従来の CG による鉛筆画の表現法は、個々のストロークをポリラインという幾何学的プリミティブを用いて表現するため、複雑なデータ構造が必要な上に、トーンのマッチングやクリッピング処理などにも多くの時間を要していた。それに比べて提案手法は、簡単なフィルタリング処理を施すだけで入力画像にマッチしたトーンをもった鉛筆画を生成することができる。本論文ではさらに、入力画像のテクスチャをより自然に表すためのストローク方向決定法も併せて提案する。

### Abstract

In this paper, we propose a new technique for automatically generating pencil drawing from 2D images using line integral convolution. Our idea is inspired by the similarity between the stroke textures of pencil drawings and the flow textures generated with line integral convolution. We succeeded in creating pencil drawings with their tone matching that of the original images simply by taking the vector field defining stroke directions and the white noise images generated by referring to the intensity of the original images as the input to the line integral convolution algorithm. By employing the texture analysis techniques, we also succeeded in automatically orienting strokes in the directions that convey the textures of objects appropriately.

### キーワード

ノンフォトリアリスティック, 絵画風画像生成, 鉛筆画, LIC

### Keyword

non-photo-realistic rendering, pencil drawing, artistic media, line integral convolution

## 1. はじめに

近年，非写実的レンダリング (Non-Photorealistic Rendering: NPR) は，コンピュータグラフィックス研究における最重要課題のひとつとして注目されている [1-2]。油彩 [3]，印象派絵画 [4]，水彩画 [5]，ペンとインクなどによるイラスト画作成 [6-12]，(色)鉛筆画 [13-17] など，伝統的な芸術メディアやスタイルをモデル化する多くの技法が提案されている。本論文では，与えられた写真や2次元画像から鉛筆画風の画像を自動的に生成する新しい手法を提案する。鉛筆画は下書き用のスケッチだけでなく，完成された作品にも利用される。鉛筆画がもつ無彩色のトーンや未完成の風合いは見る者の好奇心や想像力を掻き立てる。鉛筆画は芸術に限らず，服飾や建築設計，科学技術用イラスト画作成にも幅広く利用されている。

鉛筆画を扱っている既存の NPR 技法を振り返ってみる。PencilSketch とよばれる初期の2次元システム [17] では，マウスをベースとした仮想タブレット上で，ユーザは対話的に鉛筆の硬度，筆圧，ストロークの方向等の描画パラメータ値を指定する。最近では，Sousa と Buchanan が実際の鉛筆画の顕微鏡写真を観察した結果に基づいて，鉛筆，紙，そして両者間の相互作用に関するモデルを開発している [15-16]。モデルパラメータ値とストロークが決まると，与えられた2次元画像が対応する鉛筆画に変換される [15]。また3次元のポリゴンモデルも，モデルのパラメータ値からトーン値を決める参照テーブルを用いて，自動的に鉛筆画に高速変換される [16]。一方で高木と藤代 [13-14] は，3次元ボリュームを用いてモデル化された紙の微細構造と顔料分布を，ボリュームレイトレーシングでレンダリングする方式を提案した。これ以外にも，ペンとインクによるイラスト画作成の一連の成果 [6-12] は，本論文の手法に関連する NPR 技法である。

既存の鉛筆画生成技法とペンとインクによるイラスト画生成技法はすべて，幾何学ベースか画像ベースか，あるいはユーザによる介入が必要かどうかという，主要な2つの観点から分類できる。幾何学ベースの技法は3次元シーンの記述を入力とする。

通常3次元幾何学データとビューイング情報があれば，サーフェイスのトーンやオブジェクトの3次元形状を的確に伝えるストロークをもった画像を自動的に生成することができる。一方，画像ベースの技法は2次元画像を直接的に処理して，芸術的な表情付けやイラストの効果を創り出す。この場合は，基礎となる3次元幾何学情報やビューイングパラメータが与えられていないため，自動生成は比較的困難であり，ほとんどの手法では，ストロークテクスチャを決定する上で少なからずユーザの介入を必要とする。2次元濃淡画像を自動的にペン画像に変換する手法として中川 [12] らの手法がある。この手法では線描画風ペン画と点描画風ペン画を自動生成できる。線描画風ペン画の場合は単一方向に描かれているストロークテクスチャか，濃淡の勾配を求めることにより検出した入力画像のテクスチャの流れにそったストロークテクスチャを使用する。またあらかじめ用意した単一方向のストロークテクスチャをユーザが指定したガイドラインにそってワーピングすることにより任意の方向に流れるストロークを生成することも可能である。

本論文で提案する手法も画像ベースに属し，2次元濃淡画像を鉛筆画風画像に自動変換する。本手法の最大な特徴は近年流れの可視化に広く利用され始めた線積分畳み込み法 (Line Integral Convolution: LIC) [18] の利用である。ポリラインのような幾何学的プリミティブを用いてストロークを生成する代わり，画素ごとにローパスフィルタをかけるだけで鉛筆画を生成することができ，ストロークのスキャン変換，アンチエイリアシング，クリッピング等の煩雑な処理が必要なく，実装も容易である。また，事前に諧調テクスチャを用意する手法 [12, 15, 16] や差分画像を使ってストローク密度を制御する手法 [7, 8] に比ベトーンのマッチングが容易に行えることも特長の一つである。

本論文は，次章以降以下のように構成されている。まず次章では，鉛筆画と LIC に関して簡単な導入を行った後，LIC を利用して鉛筆画を生成する基本的な考え方を説明する。第3章は提案手法の詳細である。LIC をベースとする鉛筆画を生成するプロセス

を概観した後、自動システムを実装するための各要素技術を詳しく述べる。第4章では提案手法を用いて作成した画像例を示し、提案手法の効果を検証する。最後に第5章で本論文をまとめ、今後の研究課題に言及する。

## 2. LICによる鉛筆画生成

### 2.1 鉛筆画の基本要素

鉛筆画の本質は、主題となる要素の描画に、グレイスケールのトーンだけを利用することである。異なるレベルからなるトーンはストロークを描くことによって得られる。個々のストロークの外見は、描かれる紙の地合い、鉛筆の硬度、鉛筆の芯先が紙にどのように触れるかということに依存している [20, 22]。

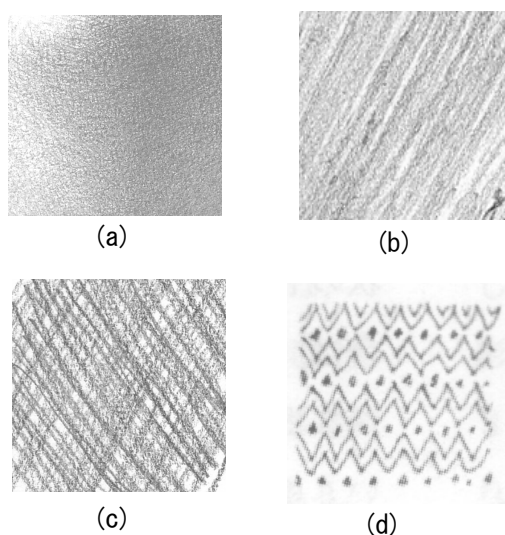


図1: 実際の鉛筆画の基本トーン。(a) 写実的トーン, (b) 平行ストロークによる模倣トーン, (c) クロスハッチングによる模倣トーン, (d) 装飾用トーン。

鉛筆画では、主に2種類の基本トーンが使われている [20]。ひとつは写実的トーン (*realistic tone*) である。図1(a)にその例を示す。写実的トーンは、対象となる要素の陰影をなるべく忠実に再現する場合に利用され、通常は異なる方向のストロークを個々が区別できなくなるまで重ねて描くことで得られる。もうひとつは、模倣トーン (*imitative tone*) とよばれる。平行ストロークとクロスハッチングに

よって描かれた模倣トーンの例を、それぞれ図1(b)と1(c)に示す。模倣トーンの場合は個々のストロークがはっきりと残っているが、見る側の人間がストロークを混ぜ合わせ、滑らかな陰影を心理的に形成する。この2種類の基本トーン以外にも、図1(d)に示したような装飾用 (*decorative*) トーンがしばしばオブジェクトの模様を表現するために利用される。

### 2.2 LIC法

LIC法は、1993年に Cabral と Leedom によって提案された、2次元ベクトル場を可視化するテクスチャベース技法である。2次元ベクトル場が規則的な直交格子によって与えられたとする。LICアルゴリズムは、そのベクトル場と同じサイズのホワイトノイズを入力し、ベクトル場の方向にそって局所的に滲んだようなテクスチャを出力する。ベクトル場の格子セルと入出力画素との間には1対1の対応が成り立っている。出力画像の各画素値を決定する際に、対応する位置の格子セルを通過するような局所流線を生成する。そして、その流線にそって定義された1次元のローパスフィルタカーネルを用いて、入力画像の流線上の画素を畳み込む。図2(c)は、図2(a)のベクトル場と図2(b)のホワイトノイズから生成したLIC画像の例である。LIC法は現在、流れ場の大局的な性質とともに、局所的な特徴の詳細も表現できる手法として注目されている。

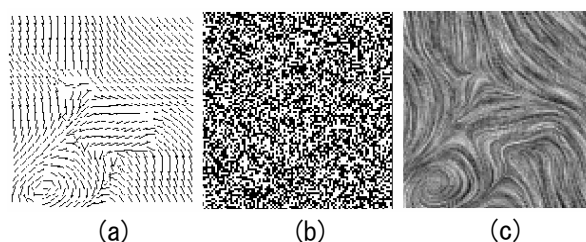


図2: LIC画像の例。(a) ベクトル場, (b) 入力ホワイトノイズ画像, (c) 出力画像。

### 2.3 LIC画像を用いた基本トーンの表現

鉛筆画を生成するためにLIC法を利用するという基本アイデアを説明する。まず図3を見てほしい。図3(a)は、実際の鉛筆画に使われている典型的な模倣トーンをデジタルサンプリングしたものであ

る。その画像全体を眺めてみると、平行なストロークによって、グレイスケールのトーンができあがっていることがわかる。一方、その画像の任意の部分領域に目を凝らすと、ストロークの方向や画素の輝度がランダムに変化していることがわかる。

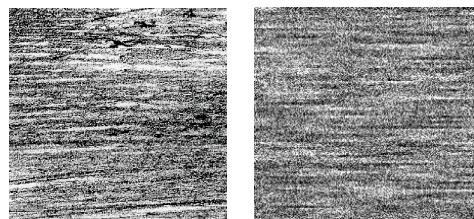


図 3: 実際の鉛筆画 (a) と LIC テクスチャ (b) との比較.

この輝度変化は、芯先と画用紙との相互作用の結果である。実際芯先から剥がれて画用紙に付着する顔料の量は、紙の地合いの程度によってランダムに変化する。この性質こそ、ペンとインクによるイラスト画など他の画法がもちえない、鉛筆画独特のやわらかくやさしい印象につながる最も重要な要因のひとつである。ところが、同様のトーンの特徴を LIC 画像も創り出していることがわかる (図 3(b))。LIC 画像は、元画像をベクトル場から導出される流線にそってローパスフィルタをかけることによって得られるので、流線にそったストローク調の軌跡が残る。一方、任意の局所領域内の画素の輝度は、LIC に入力されてきたホワイトノイズ画像の性質を継承してランダムに変化する。このような、LIC テクスチャと実際の鉛筆画との相似性を考慮して、我々は複雑で時間のかかる物理シミュレーションを実行することなしに、LIC テクスチャを用いて鉛筆画のトーンを模倣することにする。すなわち、主題の要素のトーンにマッチしたホワイトノイズとストロークの方向を与えるベクトル場を用意するだけで、LIC アルゴリズムによって鉛筆画風の画像を生成することができるのである。図 4 は、LIC 法を利用して生成した 3 種類の基本トーンである。図 4(a) は、各格子点で方向をランダムにとったベクトル場を用いて生成した写実的トーンである。平行なベクトル場を用いることによって、図 4(b) のような平行ストロークの模倣トーンを生成することもでき

る。クロスハッチングの模倣トーンは、異なる方向をもつ平行ストロークのトーンを何枚か重ねることによって得ることができる (図 4(c))。さらに、LIC 計算に用いられるパラメータ値を調整することで、特定の視覚効果を簡単に追加することもできる。例えば、荒い粒度のノイズを利用すれば、太いストロークを使った効果が出せる。また、畳み込みカーネルの長さを変化させれば、ストロークの平均長を変えることもできる。

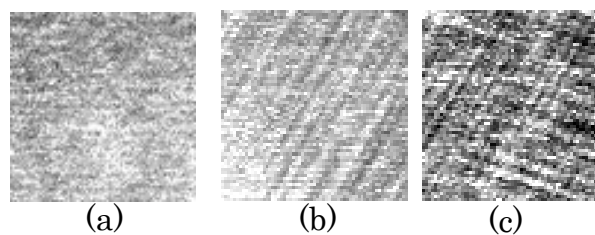


図 4: LIC 法で生成した基本トーン. (a) 写実的トーン, (b) 平行ストロークの模倣トーン, (c) クロスハッチの模倣トーン.

絵画風画像生成における LIC の利用はほかにもいくつか存在する。高木らは LIC 法を利用して色鉛筆画生成における加水効果をシミュレートしている [13-14]。ブラシストロークの速度と水の性質をフィルタカーネルの長さに、各ボクセルが貢献する顔料の量をフィルタカーネルの重みにマッピングし、ハイトフィールドに対して LIC 計算を行うことで、加水効果による顔料の再分布を擬似的に表現した。

### 3. LIC 法を用いた鉛筆画生成

#### 3.1 アルゴリズム

本節では、LIC 法を用いて自動的に濃淡画像から鉛筆画を生成する過程を説明する。入力画像がカラー画像の場合はまずそれを濃淡画像に変換する。図 5 に示すように、生成過程は以下の 7 ステップから構成されている：

1. 入力画像 (図 5(a)) からホワイトノイズ画像 (図 5(b)) を生成する；
2. 入力画像 (図 5(a)) を異なる領域 (図 5(c)) に分解する；
3. 領域の境界を抽出する (図 5(d))；

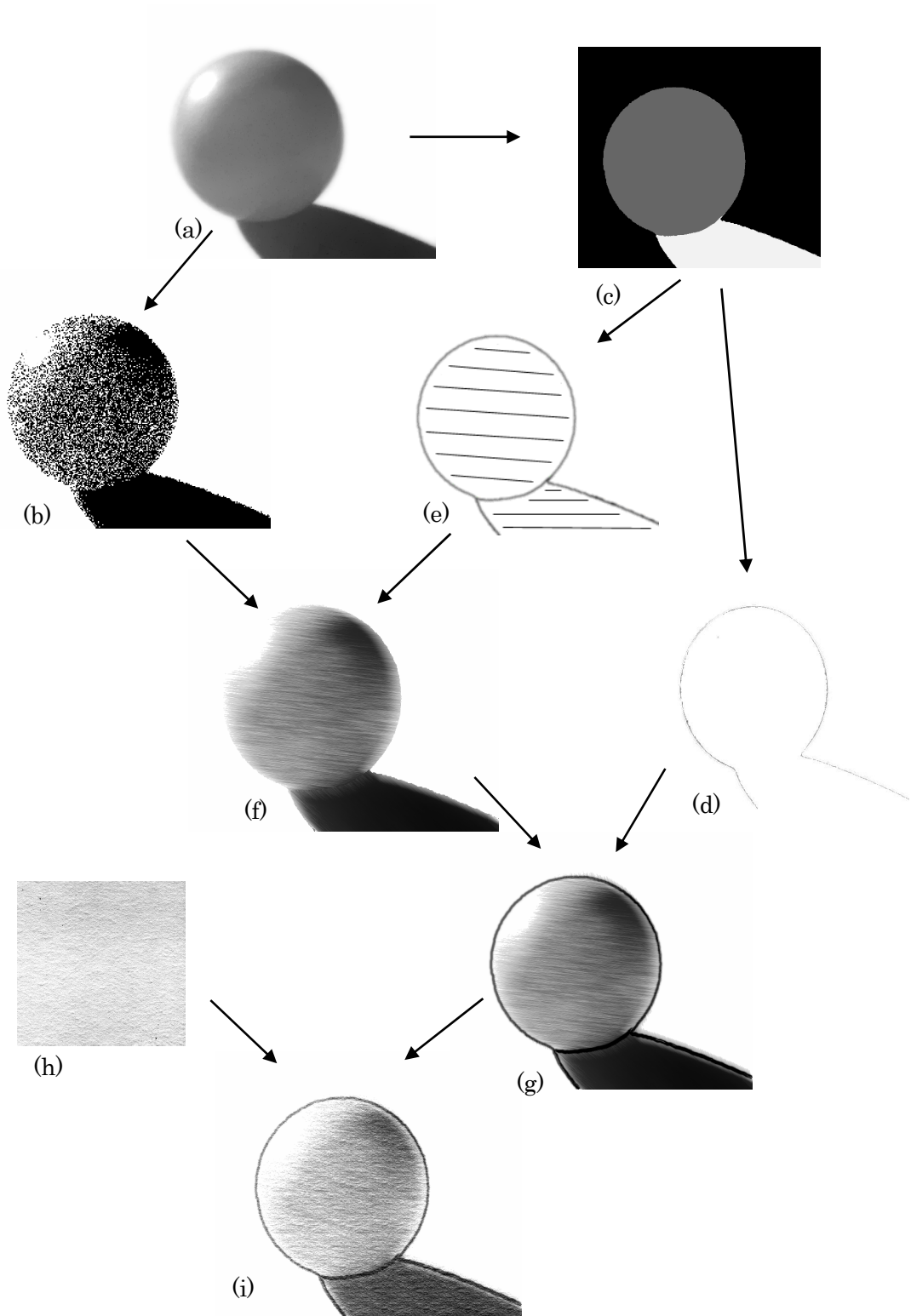


図5：LIC ベースの鉛筆画生成手続き.

4. ストロークの方向を示すベクトル場 (図 5(e)) を生成する；
5. ホワイトノイズ画像 (図 5(b)) とベクトル場 (図 5(e)) から LIC 法によって鉛筆画 (図 5(f)) を生成する；
6. 必要に応じて、境界 (図 5(d)) を結果画像 (図 5(f)) に加え、輪郭を強調した画像 (図 5(g)) を生成する；
7. 結果画像 (図 5(g)) に画用紙のサンプル (図 5(h)) を合成し、最終画像 (図 5(i)) を得る。

ステップごとに後続の節で各々詳述する。

### 3.2 ホワイトノイズ

鉛筆画の最終結果のグレイスケールトーンは、入力されるホワイトノイズ画像によってほとんど決まってしまう。入力画像と結果の鉛筆画のトーンを一致させるために、入力画像のトーンを用いてノイズの分布を決めることにする。ホワイトノイズ画像は、白の画素が現れる確率を入力画像の対応画素の輝度レベルに比例するようにとることによって得られる。 $I_{input}$  を入力画像のある画素の輝度とする。 $P$  を擬似乱数関数によって得られた  $[0.0, 1.0]$  の範囲の浮動小数点数とする。ノイズ画像の対応画素の輝度  $I_{noise}$  は、以下の式にしたがって計算される：

$$I_{noise} = \begin{cases} 255 & \text{if } P \geq T \\ 0 & \text{otherwise} \end{cases} \quad P \in [0.0, 1.0]$$

$$T = k \left( 1 - \frac{I_{input}}{255} \right) \quad k \in (0.0, 1.0]$$

ここで、 $k$  は結果の鉛筆画の全体的明るさを決定する係数である。カーネル形状の積分値を用いて LIC 計算の結果を正規化することで出力画像の明るさに対するカーネルの影響を一定にしているが、 $k$  の最適値を決めるには、カーネルの形状や長さも考慮する必要がある。結果の鉛筆画にローパスフィルタリングを施し入力画像との差を取り、その差が最小となるような  $k$  が入力画像のトーンをもっとも正確に表せるものと考えられる。デフォルトでは、

長さ 20 ピクセルのボックスフィルタがカーネル関数として使用されるが、この場合経験的に  $k$  を 0.7 前後に設定することでほとんどの画像に対して良好な結果が得られる。ユーザは対話的に  $k$  の値を調節することもできる。

容易にトーンを合わせることができる点は、LIC 法を利用する最も大きな利点である。既存の鉛筆画やペンとインクによる技法の多くは幾何学的なプリミティブを利用しているので、連続的なトーンを得るために、プリミティブの配置密度の調整に余計な手間を要する。例えば、観察に基づく鉛筆画モデル [15-16] や中川らのペン画自動生成法 [12] は、予め計算されたトーン値の参照テーブルを引用してマッチングを行っているため、連続トーンを生成することは難しい。また Salisbury ら [7-8] によって提案されたペンとインクによる技法は、差分画像を使ってストローク密度を制御している。入力画像を初期値とする差分画像は、まずストロークを滲ませた画像分を引かれる。続いて、差分画像内もっとも大きい値をもつ位置を次のストロークを置く場所とする。この処理は差分画像のすべての画素値が十分小さくなるまで繰り返さなければならない。これら既存の技法と比較して、我々の手法はより簡単にトーンマッチングを実現している。入力画像とホワイトノイズ画像の各画素をただ 1 回だけ参照すればよいので、処理時間はきわめて短くてすむ。

### 3.3 領域と境界の抽出

トーンの微妙な配合に加え、鉛筆画で 3 次元の奥行き感やシーンを構成するオブジェクト間の位置関係を伝える目的でよく利用されている技法として、輪郭線を描いたり、ストロークの外見を変更して、隣接領域の境界を強調することがあげられる。このような効果を創り出すために、既存のテクスチャベースの画像セグメンテーション技法を利用して入力画像をいくつかの小領域に分割し、個々の領域ごとに異なるベクトル場を定義することにする。現在の実装では、輝度ヒストグラムに基づくセグメンテーション技法 [21] を使って領域を抽出している。入力画像をまず小さな矩形領域に分割する。次に似た輝



度ヒストグラムをもつ隣接領域を統合する。このプロセスを統合可能な隣接領域が存在しないまで繰り返す。輝度ヒストグラムの近似度は、両者の各レベルの値の差の総和によって計算できる。この差が与えられた閾値よりも小さければ2つの領域は統合される。図6(b)は、その技法を用いて、図6(a)に示された入力画像を分割した結果である。図からわかるように、砂利を敷いた領域や道の領域が、内部にテクスチャをもっているにもかかわらず、きちんと抽出されていることがわかる。図6(c)は、図6(b)の結果に対して領域ごとに異なるストロークの方向を与え、LIC アルゴリズムを適用して生成した鉛筆画である。よい領域分割結果が得られるかどうかは初期の分割ステップにおける矩形領域のサイズや統合ステップにおける近似度の閾値の設定にも大きく依存する。これらのパラメータの最適値は入力画像に依存し、多くの場合、試行錯誤に頼るほかない。分割ステップにおける矩形領域の最適サイズは入力画像のテクスチャサイズに設定するのが望ましい。テクスチャサイズを求める技術はいくつか知られているので、これらを利用すれば、最適値の自動設定も可能と思われる。現在のデフォルト値は5×5に設定されている。

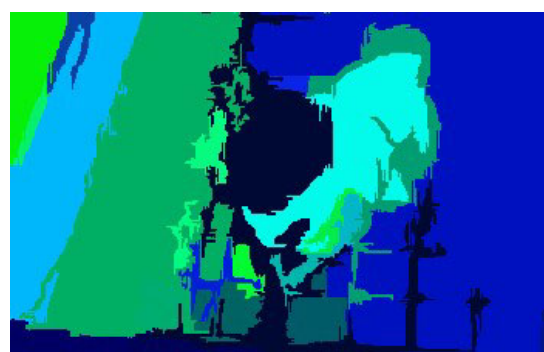
NPR 画像が写実的な画像よりも優れているところは、抽象化や念入りな詳細度制御によって、ユーザに必要な情報だけに注目させて、余分なところを無視させられることにある。特に抽象化は、鉛筆画によるスケッチに欠かせない。これを実現するには、入力画像の意味論的解析が必要となるが、それをユーザの介入なしに実現することはきわめて難しい。現在は、領域抽出の最終段階で、一定面積以下の小領域を隣接する大きな領域に統合するようにしている。

輪郭抽出については、領域境界だけでなく、領域の内部構造も表現するために、テクスチャエッジ検出ではなく、微分オペレータを利用している。内部輪郭線は、領域内部のテクスチャを表現する上で重要である。図6(c)では、結果の LIC 画像に Sobel オペレータをほどこすことによって道路の格子模様が検出できていることがわかる。微分オペレータに

より検出したエッジの濃度は隣接する領域の濃度の



(a)



(b)



(c)

図6: 画像の領域分割. (a) 入力画像; (b) 分割された入力画像; (c) 領域ごとにストロークの方向をランダムに与えて生成した鉛筆画. 画像提供: <http://www.vector.co.jp/soft/data/art/se019516.html>

差に依存する。人間が絵を描く場合は通常輪郭線を領域内部よりも少し濃く描く傾向がある。これは検出したエッジの各ピクセルに対して、その濃度を入力画像における周辺ピクセルの濃度の平均値より高く設定することで実現できる。また、実際に鉛筆画を描く場合、隣接領域のトーンの違いがあまり明確でないほど境界線を強く描く場合もある[22]。これ

を実現する方法としてエッジ画像の濃度を反転させてから輪郭線として利用することが考えられる。現在の実装では、Sobel オペレータを施した結果をそのまま輪郭線画像として利用している。

### 3.4 ストローク方向の決定

模倣トーンを利用する場合、ストロークの方向は鉛筆画らしさを決める重要な要素となる。またストロークの方向は、オブジェクトの形やテクスチャを決める上でも重要である。もし、テクスチャが方向性をもつとしたら、ストロークはテクスチャの方向にそっていなければならない。例えば、人間の頭髪を描く場合、ストロークは髪が流れる方向を向いている必要がある。テクスチャの流れを検出し、それに沿ったストロークを生成する手法はすでいくつか提案されている。中川ら[12]のペン画生成手法ではSobel オペレータを用いて濃淡の勾配を検出することでテクスチャの方向を検出している。齋藤ら[19]はフーリエテクスチャ解析法[21]を用いて局所のテクスチャの流れを検出し、それをもとにストロークの向きを決定する。本システムで実装している手法は齋藤らが利用したフーリエテクスチャ解析による手法とほぼ同じである。この方法では、テクスチャが方向性をもつならば、周波数領域に変換した後、そのパワースペクトルはテクスチャに直交する方向に大きな値をもつという性質を利用する。従って、各画素に対して局所テクスチャ方向を決定するために、その画素を中心とする正方形ウィンドウをとり、周波数領域に変換する。そして、周波数領域を小さな角度でセクタに分割し、パワーの総和の最大値をもつセクタを決定する。最後にこのピーク値の全セクタの平均値との比を求め、閾値よりも大きい場合には、そのセクタと直交する方向をその画素におけるテクスチャの方向とする。もし閾値以下の場合には、テクスチャは方向性をもたないとし、そのピクセルを含む領域に対してランダムに設定された方向を使用する。フーリエ解析を施す窓領域のサイズの最適値は入力画像のテクスチャにも依存するが、一般に小さすぎると、テクスチャの統計的な特性を表すことができず、ピクセルの位置に依存して

正しいテクスチャの方向が得られない場合がある。一方大きすぎると、局所的なテクスチャ方向が抽出できない問題がある。現在の実装では、この窓領域サイズもユーザが対話的に設定できる。デフォルト値は $18 \times 18$ に設定されている。図7に示す例では、この技法によって自動的に図7(a)の木目から図7(b)のストロークが生成できている。

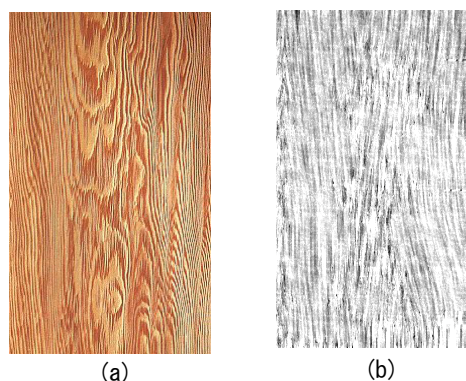


図7: 入力テクスチャ(a)の方向から自動的に生成されたストローク(b)。

### 3.5 紙のモデル

言うまでもなく、鉛筆画の品質は、画用紙の地合いに大きく依存している。紙の微細構造を観察すると、紙表面には高低差がある。それが原因となって鉛筆の芯先から顔料が削り取られ付着する。このような紙の微細構造をモデル化する試みがこれまでも続けられてきたが、典型的には紙の地合いをハイトフィールドで表現している[5, 13, 14]。Sousa と Buchman は、紙のデジタルサンプルのグレイスケール値からそのようなハイトフィールドを近似的に求めている[15]。ここでは、より簡単に直感的な方法を用いて、紙の地合いの影響を含めることにした。実際、我々は結果のLIC画像から紙のデジタルサンプル画像の差分をとり、明るさを補正することで、実用上十分な品質の画像を得ている。実際、デジタル化された紙のサンプルにおける陰影は、ハイトフィールドの山と谷によって決まり、それは画用紙に顔料が付着する量とも直接的な関係をもっていることから、この方法は簡便ながら、鉛筆画の物理的原理にも合致したものであるといえる。図8(a)と(b)に示した紙のサンプルを用いて得られた鉛筆画



をそれぞれ図 8(c) と (d)に示す.

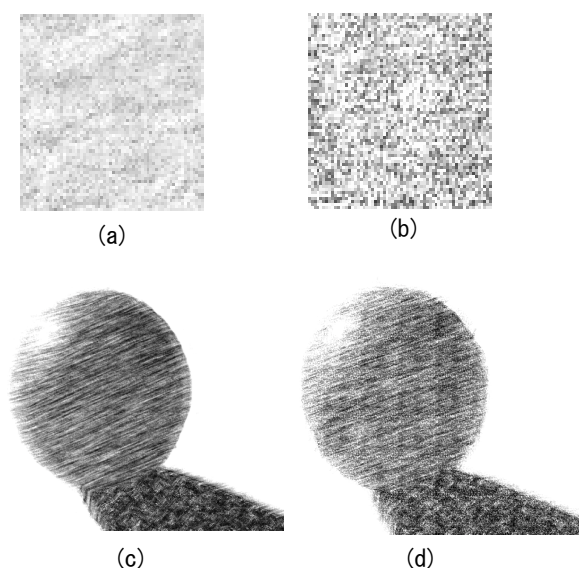


図 8: 画像(c)と(d)はそれぞれデジタル化された紙のサンプル(a)と(b)を合成して得られた鉛筆画.

#### 4. 実験結果

我々は、第 3 章で述べてきた技法を実装し、Windows アプリケーションとして鉛筆画自動生成システムを開発した。入力画像が指定されると、デフォルト値と入力画像からシステムが導出した値を用いて、自動的に対応する鉛筆画が生成される。ユーザは、画像全体の明るさ、領域分割するための領域サイズと近似度閾値、ストロークの太さと長さを決定するためのホワイトノイズの粒度、畳み込みカーネルの長さ等のパラメータ値を対話的に指定することができる。ただし、これらのパラメータは画像全体に対して設定され、現在のところ、領域によってストロークの太さや長さを変えることができない。また、デフォルトで設定されている値は必ずしも最適なものではない。自分自身のスタイルで描きたいユーザは、マウスで境界線を指定することにより領域を指定し、領域ごとに一本のサンプルストロークを描くことでストロークの方向を指定することができる。なお、本論文に示した画像については、領域とストロークの指定に関して一切ユーザの介入がなかった。

図 9(b) と (c)は、図 9(a)の写真から生成された 2 枚の鉛筆画である。図 9(b)を生成する際、パワーピーク比を決める閾値は最低値としたため、各画素におけるストロークは必ず最大パワーセクタと直交する方向にとられている (3.4 節参照)。雲のテクスチャは特定の方向をもたないので、各画素に対してほとんどランダムなストロークが選択されており、ちょうど写実的トーンを用いた実際の鉛筆画のような風合いを出すことに成功している。一方、図 9(c)は領域ごとにランダムに決められた方向のストロークが使われているので、平行ストロークによる模倣トーンを用いた効果が表出している。ストローク方向の決定法を除いて、この 2 枚の画像生成に使用したそのほかのパラメータ値はすべて同じである。どちらの画像からも、実際の鉛筆画における消しゴム効果(erasing)が得られていることに注意されたい。消しゴム効果はハイライトによく用いられる [20]。

図 5 と図 8 のボールの画像でも、ハイライトの自然な表現が、LIC 法によって一切の付加的な計算なしに実現できている。図 10(b) は、図 10(a)の写真から自動的に生成された鉛筆画である。テクスチャの方向選択によって、樹木や山々のテクスチャが区別して描かれている。さらに自動的に検出された領域輪郭も、画像の品質向上に貢献している。

図 6(c)に示した画像の解像度は  $352 \times 228$  であり、Pentium III Xeon 933MHz の PC を用いておよそ 1 分で全処理が計算できている。図 9 の画像の解像度は  $1024 \times 768$  であり、同じ環境で、図 9(b)と(c)の鉛筆画を得るのにおよそ 20 分を要した。解像度が高い場合の処理の遅さが目立つが、全計算時間の 90% は画像の領域分割に要しており、5%はテクスチャの方向を決定することに費やされている。図 10 の画像解像度は  $1024 \times 768$  であるが、局所テクスチャの方向をそのまま各画素に用いているため、画像セグメンテーションのステップを省略することができた結果、全実行時間は 1 分 25 秒です。領域分割コードの最適化によって処理速度を劇的に改良することができると考えられる。また、より高速な領域分割法について現在調査を行っている。