

写真からの対話的イラスト生成 Interactive Illustration Generation from Pictures

岩田 大輝[†] 豊浦 正広[†] 茅 暁陽[†]
Taiki Iwata Masahiro Toyoura Xiaoyang Mao

1. はじめに

情報メディアとして写真が存在する。しかし、写真には提示が必要でない情報が含まれることがある。逆に、提示する対象が目立たないこともある。それに比べ、イラストでは、必要な情報のみを強調することや、必要でない部分を省略することができる。植物学分野で用いる挿絵や医学標本の説明図、指名手配犯の似顔絵など、提示する側が情報を取捨選択してイラストを生成する例は多くある。

ユーザが写真からイラストを生成する場合、ユーザによって強調したい情報は千差万別であり、画像上の特徴によって決定されるものでないことは多い。また、選択した情報をどのように表現するかも、ユーザごとに異なる。我々は、このようなイラスト生成をコンピュータと対話的に行うことを目指す。

コンピュータによるイラスト生成のためには、様々な手法が提案されている。既存の描画ツールでイラストを生成する場合、ユーザが描画したい領域や描画スタイルをすべて設定する必要があるため、ユーザへの負担が大きい。また、描画ツールには様々な描画スタイルが用意されているが、ユーザ独自の描画スタイルを表現するには不十分である。イラスト生成の代表的なものとして、ワシントン大学の研究グループによるペンとインク調画像生成に関する一連の研究[1]がある。用意されているペンとインク調のパターンを用いて、対話的にペンとインク調の画像を生成できるシステムや、3次元モデルをペンとインク調にレンダリングする方法が開発された。また、Luら[2]は医用イラストの色合いをボリュームデータに適用することで、3次元イラスト調画像を自動生成する手法を開発した。しかし、これらの研究もユーザの意図に合わせて情報の取捨選択や、ユーザ独自の描画スタイルの反映はできない。

そこで本研究は、写真からユーザの意図に合わせて対話的にイラストを生成する手法を提案する。ユーザが部分的に描画例を与えることで、ユーザの描画したい領域を写真上から自動選択し、ユーザの描画例からユーザ独自の描画スタイルの特徴を学習して、イラストを対話的に生成していく(図1)。

ユーザが描画を加えたい領域および輪郭を特定するために、領域の場合は模様によってその特徴を定義し、輪郭はその両側の色分布によって定義する。描画例が与えられた位置での領域・輪郭の特徴を抽出し、同じ特徴を持つ領域・輪郭に描画例と同様な描画スタイルを適用することで、ユーザのイラスト生成を支援する。最終的には、ユーザの希望に合わせた様々な描画スタイルを提供するシステムを目指す。今回の実装では、ペンストロークについての描画スタイルの適用まで行った。

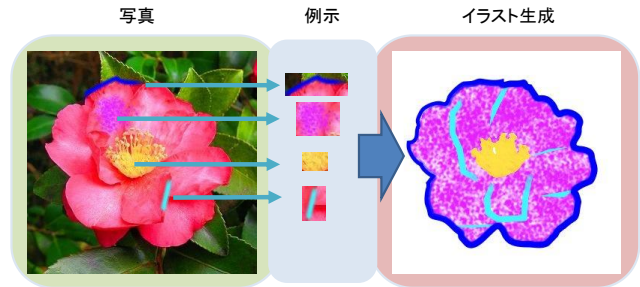


図1. 写真からのイラスト生成

2.では、ユーザが部分的に加えた描画例からユーザの描画したい領域をどのような特徴量を用いて探索するかを述べ、どのようにユーザの描画模様を特定の領域に対して反映させるかを述べる。

3.では、ユーザが与えた描画例のストロークからどのように輪郭に反映させるかを述べる。

4.でイラスト生成のためのインターフェースについて説明する。

5.で生成結果を示し、6.でまとめる。

2. 領域の描画

イラスト生成のためには、まずユーザの部分的な描画例から、ユーザが描画を加えたい領域を特定する。描画例が与えられた位置の模様を抽出し、同じ模様を持つ領域に描画例と同じ描画スタイルを適用する。

領域の描画には、Image Analogies[3]を利用する。Image Analogies は画像Aと変換後の画像A'を教師として与えると、任意の入力画像Bに対してAからA'と同じ変換を施しB'を得る技術である。

本研究の場合には、ユーザの描画領域をA、描画例をA'とし、同じ模様を持つ領域Bに対して、Image Analogiesを適用して、出力B'を得る(図2)。

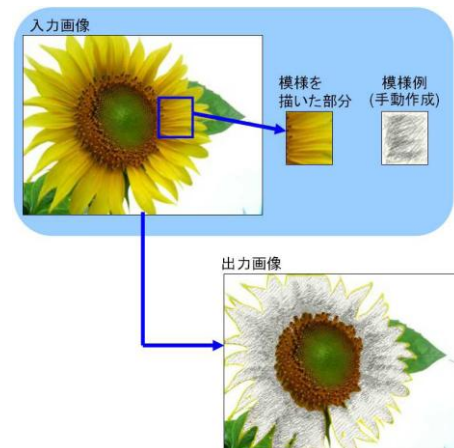


図2. 領域描画

[†] 山梨大学大学院医学工学総合研究部, Graduate School of Engineering and Medicine, University of Yamanashi

描画領域の特徴は、描画領域内の画素に対して $n \times n$ のウィンドウを設定し、 $n \times n$ の特徴ベクトルとする。同じ模様を持つ領域を抽出するためには、判定対象となる画素周辺に同様の $n \times n$ のウィンドウを設定する。A 上を走査することで得られるすべての $n \times n$ のウィンドウに対して、ウィンドウ内の色の距離を計算する。A 上で対象画素と十分に距離が小さいウィンドウが得られれば、対象画素は A と同じ模様を持つと判断する。比較計算には残差逐次検定法(SSDA 法)を利用することで、計算時間の短縮を行うことができる。

3. 輪郭の描画

輪郭は、特徴が異なる 2 つの領域の境界を示す。ユーザの例示した輪郭線と類似した輪郭を抽出するためには、まずユーザの例示した輪郭線から特徴を抽出する。入力画像内でエッジが現れる部分を探索対象として特徴を計算し、例示された輪郭線の特徴と比較を行う。比較した結果、類似度が高ければ類似輪郭として抽出する。エッジが現れる部分を探索対象としているのは、輪郭が特徴の異なる 2 つの領域の境界を示すことから、画素の変化が激しいエッジとして抽出できるためである。これにより、画像全体を探索する必要がなくなり、処理が高速化する。

3.1 ユーザの例示方法

本研究では例示の際にペンタブレットを用いて写真上の輪郭の一部をなぞってもらい、輪郭線を挟んで一方の領域にプロットしてもらい、図 3 はユーザが花びらの輪郭を抽出するために、花びらの輪郭の一部をなぞった例である。輪郭線は隣接する二つの領域の境界であるため、輪郭線の例示だけでは、コンピュータがどちら側の領域の輪郭線として抽出すればよいのか知ることができない。そこで、どちらの領域の輪郭線を抽出すればよいかを示すためにユーザの意図する領域をクリックしてもらい、図 3 では花びらの領域内にプロットしている。これにより抽出したい領域を知ることができるため、ユーザが抽出したい輪郭特徴を計算することができる。輪郭特徴として対象領域内の色分布とテクスチャ特徴を用いる。

実際にユーザが加えた輪郭の特徴計算の対象領域はプロットから例示した輪郭線の始点と終点に囲まれた領域となる。この領域と同様な特徴量を持つ領域を画像全体から探索する。探索方法については 3.3. で詳しく述べる。

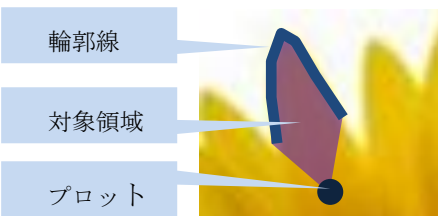


図 3. ユーザの例示の様子

3.2 テクスチャ特徴量を考慮した輪郭抽出

写真内の描画したい対象がほぼ単色の場合、色分布を特徴量とすることで対象輪郭を抽出することができる。しかし、オブジェクトの輪郭付近にテクスチャ特徴がみられる場合は、領域全体の平均色が同じであっても局所的に色の分布が異なる場合があるため、色だけを調べるのでは不

分である。そこで本研究ではテクスチャを持った画像に対してもうまく輪郭抽出できることを目指し、色分布特徴だけでなくテクスチャ特徴も考慮した輪郭抽出を行う。

テクスチャとは、ある濃度や色の 2 次元的なパターンの集合であり、テクスチャの一様性によって特徴付けることができる。テクスチャ解析の方法を大別すると、統計的手法と構造的手法に分けられる。前者はきめが細かく規則的でないようなものの解析に用いられる。後者はテクスチャを構成する要素やその配列が規則的なものに対して適用され、テクスチャ要素とその配列規則によってテクスチャ構造を記述する際に用いられる。今回のような場合、テクスチャを構成する要素やその配列に規則性があるとは限らないので前者の統計的手法を用いてテクスチャ特徴を計算する。

2 次元的な画素パターンの特徴をとらえる方法として、濃度共起行列を用いたテクスチャ特徴の計算方法[4]がある。濃度共起行列とは画素値 i の点からある方向 θ に一定の距離 d だけ離れた点の画素値が j である確率 $P_{\langle d, \theta \rangle}(i, j)$ ($i, j=0, 1, 2, \dots, n-1$) を要素とする行列 $P_{\langle d, \theta \rangle}$ と定義される。ここで、 n は画素値のレベル数を表しており、今回の特徴抽出では処理時間も考慮して $n=16$ とした。画素間の距離 d は、特徴計算するテクスチャの画素パターンに適した距離を設定する必要がある。

太田らの研究[4]では、濃度共起行列から得られるテクスチャ情報に対する最適な画素間の距離を、高解像度衛星画像に対して森林領域のテクスチャとしてとらえた場合を例に検討した。その結果、画素間の距離は汎用性が高い $d=1$ と設定することが望ましいと考察している。従って、本研究でも画素間の距離を $d=1$ とした。方向については点対称の関係にある i と j が入れ替わっても同じとみなし、 $\theta = 0^\circ, 45^\circ, 90^\circ, 135^\circ$ の 4 方向とした。

求めた 4 方向の濃度共起行列に対して、それぞれの正規化を行なう。次に、式(1)に示すような 4 つの特徴量を計算し、本研究のテクスチャ特徴量とした。テクスチャ特徴量に関しては、4 方向に対して特徴計算を行なうので 1 画素あたり 16 次元の特徴ベクトルを持つ。

$$\left. \begin{aligned} q_1 &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \{P_\theta(i, j)\}^2 \\ q_2 &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (i-j)^2 P_\theta(i, j) \\ q_3 &= -\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P_\theta(i, j) \log P_\theta(i, j) \\ q_4 &= \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} i \cdot j P_\theta(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \end{aligned} \right\} \quad (1)$$

$$\left. \begin{aligned} \mu_x &= \sum_{i=0}^{n-1} \left\{ i \cdot \sum_{j=0}^{n-1} P_\theta(i, j) \right\} \\ \mu_y &= \sum_{j=0}^{n-1} \left\{ j \cdot \sum_{i=0}^{n-1} P_\theta(i, j) \right\} \\ \sigma_x^2 &= \sum_{i=0}^{n-1} \left\{ (i - \mu_x)^2 \cdot \sum_{j=0}^{n-1} P_\theta(i, j) \right\} \\ \sigma_y^2 &= \sum_{j=0}^{n-1} \left\{ (j - \mu_y)^2 \cdot \sum_{i=0}^{n-1} P_\theta(i, j) \right\} \end{aligned} \right\} \quad (2)$$

本研究で提案するシステムは、ユーザが対話的にイラストを生成することを可能にする必要がある。しかし、今回追加したテクスチャ特徴量の算出は時間的高い処理であるため、毎回画像内の探索点についても特徴量を計算しては対話的な処理を行なうことはできない。そこで、前処理として画像を読み込む段階で特徴量の計算を格納しておくことで、ユーザが例示を与えてからは対話的にイラストを生成できるようにした。

3.3 判別分析法を用いた境界の分割手法

類似している輪郭を抽出するためには、単純な $n \times n$ のウィンドウではうまくいかない。なぜなら、オブジェクト間の境界を無視して背景の特徴まで混合してしまうからである。したがって、輪郭の探索の際の特徴計算では、境界上にあってオブジェクトと背景を完全に分離させて特徴量を計算しなければならない。

本研究では、探索画像の注目点に対し近傍をとり判別分析法を用いて前景と背景を分離する(図4)。

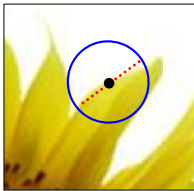


図4. 輪郭周辺の色分布とテクスチャ特徴

判別分析法とは、あらかじめ2つのクラスに分類できることを前提にクラスの分離度が最大、つまりクラス内分散 σ_w^2 が最大でクラス間の分散 σ_b^2 が最小になるような閾値を決める方法である。

分離度は、クラス間分散 σ_b^2 とクラス内分散 σ_w^2 の比である σ_b^2 / σ_w^2 で定義される。全体の画素値の平均と分散を m_t と σ_t^2 、A画素クラスを m_A と σ_A^2 、B画素クラスを m_B 、 σ_B^2 とし、 ω_A と ω_B をそれぞれA画素クラス、B画素クラスに属する画素の数とすると、クラス内分散 σ_w^2 は、式(3)で表される。

$$\sigma_w^2 = \frac{\omega_A \sigma_A^2 + \omega_B \sigma_B^2}{\omega_A + \omega_B} \quad (3)$$

クラス間分散 σ_b^2 は、式(4)で表される。

$$\sigma_b^2 = \frac{2\omega_A \omega_B (m_A - m_B)^2}{(\omega_A + \omega_B)^2} \quad (4)$$

全分散 σ_t^2 とクラス間分散 σ_b^2 、クラス内分散 σ_w^2 には、式(5)のような関係があるため、次の式(6)が成り立ち、閾値 t に関係なく全分散 σ_t^2 は一定のため、クラス間分散 σ_b^2 が最大になるように閾値 t を決めればよい。

$$\sigma_t^2 = \sigma_b^2 + \sigma_w^2 \quad (5)$$

$$\frac{\sigma_b^2}{\sigma_w^2} = \frac{\sigma_b^2}{\sigma_t^2 - \sigma_b^2} \quad (6)$$

探索点に対して、エッジにそって前景と背景を分離し、前景と背景で色分布特徴量とテクスチャ特徴量を計算する。ユーザが与えた例示の特徴量と前景と背景のどちらか一方の距離が近ければ類似輪郭と判断する。また、前景と背景

の両方が例示特徴との距離が近ければ、オブジェクト内のエッジであるといえる。

3.4 例示ストロークの反映

ユーザの例示と類似した輪郭線を抽出した後それに、例示の描画スタイルを反映させる。描画スタイルの反映方法として Curve Analogies[5]と呼ばれる方法がある。Curve Analogies は、曲線 A に対する描画 A' が与えられるときに、曲線 B を A' と同じスタイルの出力 B' に変換する手法である。しかし、Curve Analogies は曲線としての描画例の幾何形状しか反映できない。

本研究では、描画例の幾何形状だけでなく、輝度、色、太さなどを考慮したブラシストロークの反映も行う。現在、ペイントソフトでは様々なブラシが用意されており、ユーザが好みのブラシや色を選択して描画を行う事ができる。本研究では、筆ペン風の描画スタイルについて実装を行った。

3.4.1 抽出輪郭線のベクトル化

描画スタイルを反映させるためにはまず、抽出した輪郭線のベクトル化を行う。抽出した輪郭は画素の集合でしかない。そこで、点のつながりを探索し、1つの線分として扱えるようにベクトル化を行う。これにより、線分の形状や線分内の位置を元にブラシの太さや濃度を変化させることができ、様々なスタイルを適用することができる。

まず、8近傍の連結窓を用意し輪郭抽出画像に対して輪郭追跡を行う。ラスタスキャンによって、白画素から黒画素に変わる画素を探索する。進入方向を基点に番号順位左回りで黒画素がないか探索する。見つかった場合にはその黒画素に探索窓を移動する。もし、探索済み黒画素が探索開始点に戻ったら探索を終了し追跡結果を登録する(図5)。探索済み黒画素に達した場合に探索を終了させることで、一度の探索では分岐点の探索ができないが、ストロークの無駄な二度書きを防止することができる。

追跡輪郭線はチェーンコードで出力する。チェーンコードとは、開始点の座標値と方向コードを並べたものである。この処理を抽出した輪郭線すべてに対して行う。また、ベクトル化の際にある程度の長さをもった線分以外は、輪郭線として扱わないようにした。

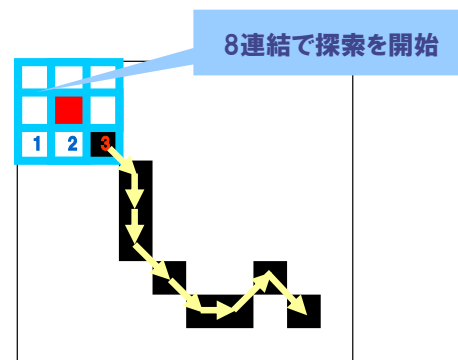


図5. 探索順序

3.4.2 筆ペン風スタイルの適用

ユーザの描画スタイルとして筆ペン風の描画スタイルの実装を行った。抽出した輪郭線のベクトル情報をもとに、筆ペンテクスチャを貼り付けていく。筆ペン風スタイルの特徴として、ひとつの線分に対して筆の書きはじめと書き終わり部分で太さを細くし、透過性を持たせた。また、線

分の中点で線の太さが最大になるようにした。最後に、テクスチャを貼り付けているので境界付近の色や角を目立たせないように処理を行った。

3.4.3 太さと透過性とぼかし表現

筆ペンの特徴を捉えるために、書きはじめと書き終わり部分の線の太さを変化させる。より筆ペンのストロークに近い表現をするためには、線の太さを急激に変化させるよりも滑らかに変化させたほうが良い。そこで今回は、輪郭線の線分の長さによって二次関数的に太さ及び透過性を持たせることにした(図6)。また、筆の太さの最小値を t_{min} 、最大値を t_{max} 、輪郭線分の長さを l とすると、始点からの距離 l_c によって太さ t_c が決まる(式7)。透過性についても同じように輪郭線分の中点をもっとも非透過性を持つように変更することで筆の書き始めと書き終わりの表現をした。

$$t_c = -\frac{4(t_{max} - t_{min})}{l^2} l_c^2 + \frac{4(t_{max} - t_{min})}{l} l_c + t_{min} \quad (7)$$

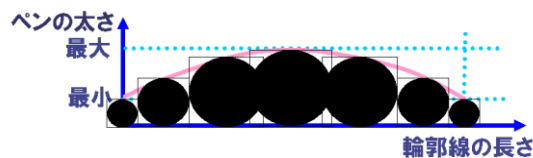


図6. ペンの太さの決定

輪郭線分情報を元に、ペンテクスチャの太さや透過性を変化させながら貼り付けを行う場合、テクスチャ間の境界部分の色の変化や角が目立ってしまう。そこで、ぼかし処理を行うことで、境界部分の色の変化や角の目立ちを軽減した。今回はガウシアンフィルタを用いてぼかしを行った。図7は、実際にガウシアンフィルタをかけた結果である。境界部分の色の変化や角が目立たなくなっていることが分かる。

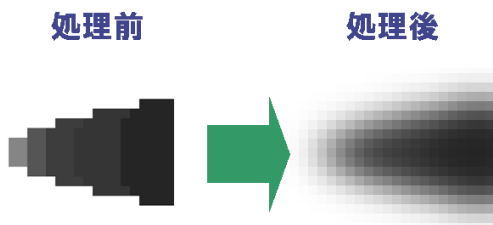


図7. ガウシアンフィルタによるぼかし処理

4. イラスト生成環境

ユーザが対話的にイラスト生成を行うためのインタフェースを実装した。図8に実行画面を示す。入力画像に対して、描画する模様をテンプレートとして与えることができ、鉛筆とブラシから輪郭スタイルを選択できる。

イラストは一般的に抽象化したものであり、ユーザが強調したい部分以外は省略して書かれることが多い。そのため、写真に写っている物体すべてに模様や輪郭線を適用するのではなく、ある程度模様や輪郭線を描画した後は一括して適用部分以外を省略する処理が行えるようにした。

模様の探索には、多解像度ピラミッド探索を利用し、また、輪郭の探索のためには、あらかじめ画像全体に対して特徴量を計算しておくことで、対話的なイラスト生成を実現した。

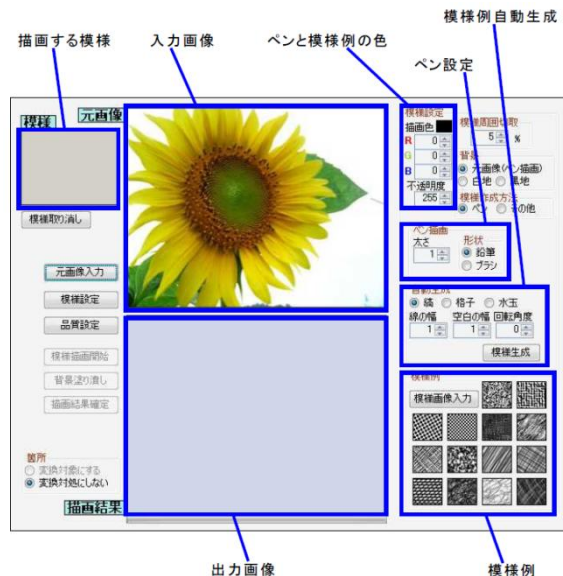


図8. イラスト生成の実行画面

5. 結果

5.1 模様描画結果

本研究の模様描画の手法を用いた結果を図9に示す。図9はひまわりの写真の花びらの部分に手で鉛筆画風の模様例を書き込み、本研究の手法を用いて変換を行った例である。出力された画像には花びらの部分に鉛筆画風の模様が貼り付けられた。この例で書き込んだ模様は、模様を描いた部分の明暗を多少考慮して作成したため、出力結果は明るい部分には明るい模様、暗い部分には暗い模様が貼り付けられた。意図している場所へユーザの模様スタイルが適用できていることが分かる。

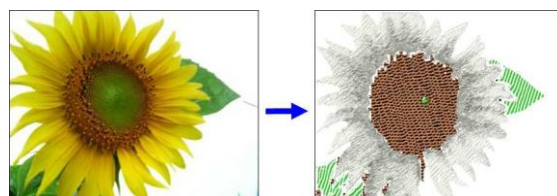


図9 模様描画結果

5.2 輪郭抽出結果

猫の茶色の毛の輪郭部分についてテクスチャ特徴量を考慮した輪郭抽出を行なった。図10はその結果である。色特徴のみの輪郭抽出結果に比べて茶色の毛の部分の輪郭が抽出できていることがわかる。しかし、茶色の毛で囲まれた領域のすべてを抽出することはできなかった。

この原因として濃度共起行列を用いたテクスチャ特徴量が、特徴計算時の近傍のサイズや共起行列の画素値のレベル、画素間の距離に大きく依存する事が挙げられる。また、自然画像に対して用いると注目画素を1画素ずらしただけでも特徴量がかなり異なることも考えられる。テクスチャ特徴計算時の近傍の最適なサイズは、テクスチャの画素パターンを十分に含めるサイズである。また、注目画素間の距離として $d=1$ で抽出を行なったが、粗いテクスチャを持つ画像に対しては、画素間の距離を大きくするなど、その

テクスチャにあった距離選択が必要であると考えられる。しかし、いずれも画像内に含まれるテクスチャによって再設定する必要があり抽出精度に偏りが出てしまう。そこでこれらのパラメータは固定し、画像に対して多重解像度分析を行う手法が考えられる。これにより、パラメータの設定を変えずに局所・大局にわたってテクスチャ特徴を計算することができると考えられる。

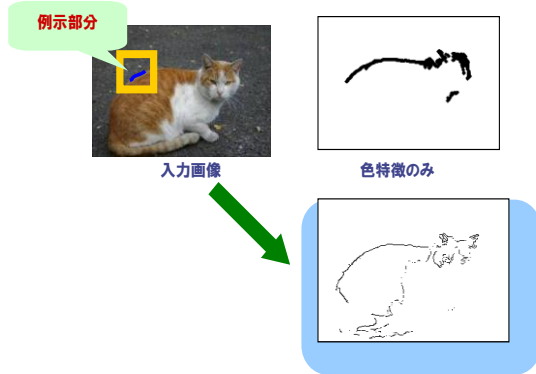


図 10. 輪郭抽出結果

5.3 筆ペン風スタイルの適用結果

本研究で実装した筆ペン風スタイルを抽出した輪郭線に適用した結果を図 11 に示す。図 11 に示すように入力線に対して、筆ペン風のストロークが適用できていることがわかる。しかし、入力線で急激に曲がっている部分については、反映結果が少し不自然に見える。これは、線の太さが始点からの距離によってのみ変化しないので、筆ペン本来の曲線部分の外側へのふくらみが表現できていないからである。今後は、連結情報を元に注目点の前後の点の配置も考慮する必要がある。



図 11. 筆ペンスタイルによる輪郭描画

5.4 統合結果

図 12 は、左が入力画像で右が生成したイラスト画像である。上の写真では、花のイラストが生成できている。同じ特徴を持つ領域・輪郭に描画例と同様な描画スタイルを適用することで、ユーザのイラスト生成を支援することができた。花の左上のおしべに対しては、輪郭が抽出できているが右下のおしべに対してはうまく輪郭が抽出できていない。これは、見た目以上に花卉とおしべの輝度に差がないためである。動的に抽出時の閾値を設定することでこれは解決できる。模様に関しては花びらの左下の先部分にうまく模様が適用できなかった。これはこの部分が、影にな

っているためである。この問題は、ヒストグラムの平坦化処理を施すなどの濃度分布の正規化によって解決できる場合がある。



図 12. イラスト生成の結果

6. まとめ

イラスト生成では、ユーザの目的に合わせて必要な情報のみを強調することが重要である。本研究では、ユーザがごく一部の描画例を与えるだけで重要と思われる領域や輪郭を自動抽出でき、ユーザ独自のスタイルを適用することができる。

今後の課題として、複雑な画像に対して適用し、領域および輪郭の抽出精度の向上があげられる。また、具体的な応用を見据えたインタフェースの改良も行う予定である。

参考文献

- [1]D. H. Salesin, and G. Winkenbach, "Computer-gene rated pen-and-ink illustration," SIGGRAPH, pp.91- 100, 1994.
- [2]A.Lu, and D.S. Ebert, "Example-based Volume Illustrations," IEEE Visualization 2005, pp. 655-662, 2005.
- [3]A.Hertzmann, C. Jacobs, N. Oliver, B. Curless, D. Salesin, "Image Analogies," ACM SIGGRAPH 2001. pp.327-340, 2001.
- [4]太田徹志, 村上拓彦, 加治佐剛, 溝上展也, 吉田茂二郎, "同時生起行列から得られるテクスチャ情報に対する最適ピクセル間距離の検出-高解像度衛星画像を用いたスギ林区分分を目指して-", 九州森林研究, pp.137-140, 2006.
- [5]A. Hertzmann, N. Oliver, B. Curless, S. Seitz, "Curve Analogies," Eurographics Workshop on Rendering, pp.233-246, 2002.