

卒業論文概要書

CD

2007年 2月提出

学籍番号 1G03R229-7

学科名	コンピュータ ネットワーク工学科	氏名	吉木直人	指導 教員	大石 進一
研究 題目	写真のイラスト化				

概要

今日のCG技術の進歩により、

- ・誰でも絵が描けるようになった
- ・従来の画家の存在の危機

という二つの変化があったと考えられる。一つ目は、アプリケーションツールがデッサン、彩色などを行い、ユーザは自分の求める絵を描くことができるようになったという変化である。

二つ目は、技術がツールによってコピーされてしまうため、芸術家としての存在意義を失いかねるという結果になったという変化である。

本研究においては、コンピュータを用いた新しい芸術を作り出すことを目的に、写真を自動的に手描きのイラストに近づけるということを実験している。

1. 『加法混色』について

色彩を、色光の3原色(R,G,B)に分け、この3色の組み合わせによってすべての色を表現するという混色の方法。JavaにおいてはR、G、Bはそれぞれ整数値で0から255までの値をとり、すべての色を表現している。

2. イラストの定義

辞書的なイラストの定義とは、
「対象の特徴を抽象化し文字で表すことのできないものを見る人間に明快なイメージを持たせることを目的とした絵」となっている。
本研究では、“Timberland”という絵と上の辞書的なイラストの定義を元に、イラストの条件を以下の3つの条件を満たすもの、と定める。

- [条件 1]対象の特徴を抜き出し、簡略化できている
- [条件 2]輪郭線及びそれ以外の、“必要な線”が作られている
- [条件 3]色彩が単調化され、明快になっている

3. 製作

3.1 ポスタリゼーション

ポスタリゼーションとは、色の連続的な変化を、断続的なすることである。対象をイラストする場合、色を強調するために連続した色をあえて断続した変化にすることがある。

3.1.1 ポスタリゼーションの実装

あらかじめ閾値を定め、取り出した色情報(R,G,B)が閾値以下だった場合、その値を一つ下のレベルに下げる。これを実装すると以下のようなアルゴリズムになる。(例はR = redの場合、ただしpostalはint型の変数で、閾値を表す)

```
1 int red = ( red / postal ) * postal;  
2 if ( red <= postal ) red = 0;  
3 if ( red >= postal ) red = 255;  
   (一部抜粋)
```

3.2 輪郭線抽出

輪郭線抽出とは、輪郭線にあたる部分をその周辺の色情報から認識することである。輪郭線の認識をした場所に、輪郭線となる色をおく。

3.2.1 輪郭線抽出の実装

輪郭線抽出の方法として、微分輪郭線抽出法を用いた。これは、前後のピクセルのRGB値の差の絶対値をそれぞれ比較し、その差があらかじめ決めておいた値よりも大きかった場合、その部分を物体と物体の境界であるとして、輪郭線となる色を置く、ということを行う。この作業を全ピクセルに対して行う。
この輪郭線抽出のアルゴリズムの詳細に関しては、論文の3.3.2節を参照。

4. 結果

4.1 ポスタリゼーション

ポスタリゼーションの実験から、次のようなことがわかった。

- ・ 色彩の少ない画像の場合、ポスタリゼーションは大いに効果があり、イラストの条件の、[条件 1]と[条件 3]を満たす画像を作成することができる。
- ・ 色彩の多い画像の場合、同一平面内においても色の微妙な変化により、異なる色で塗ってしまうなどの問題が起き、[条件 1][条件 3]を満たすことができず、イラスト化に失敗するということがある。

4.2 輪郭線抽出

輪郭線抽出の実験から、次のようなことがわかった。

輪郭線抽出が成功し、イラストの条件の[条件 2]を満たすことができるかどうかは、ポスタリゼーションにより[条件 1]、[条件 3]を満たすことができるかどうかに依存している。ポスタリゼーションにより同じ物体/同一平面をきちんと塗り分けることができれば、輪郭線もきちんと抽出することができる。

5. 考察

5.1 ポスタリゼーションについて

ポスタリゼーションについては、4.1 に示した結果から、色彩の少ない画像の場合は、現状のプログラムでも[条件 1][条件 3]を満たすような結果を得ることができるが、色彩の多い画像の場合はそれが難しいことがわかった。

この問題に対しては、次のようなアルゴリズムを実装することで、解決することができるのではないかと考えた。

- ・ R,G,B の 3 色の閾値を個別に定める。
- ・ 1 つの色に関しても、閾値を単一ではなく変動的に求める
- ・ GUI でバーなどを作り、ユーザが画像を見ながらリアルタイムに閾値を調節することができるようにする

- ・ ユーザが、スポイトのように画面上の色を抽出し、強調したい色などを自分で設定できるようにする

5.2 輪郭線抽出について

輪郭線抽出については、4.2 に示した結果から、ポスタリゼーションがイラストの条件を満たすようにうまく動作すれば、輪郭線抽出も成功するということがわかった。つまり、輪郭線抽出においても、5.1 で示した方法実装すれば、輪郭線抽出に関する問題も、ある程度解決すると考えられる。

また、そのほかの問題として、輪郭線以外のノイズのようなものも輪郭線と認識して描写してしまうということがある。この問題に関しては、微分輪郭線抽出法以外の輪郭線の抽出法を取り入れるか、あるいはノイズを、その周りのピクセルの色情報から認識し、あらかじめ除去するようなアルゴリズムを別に作ることで、解決すると考えられる。

5.3 むすび

本研究の目的は、アプリケーションによって自動的に、どれほど手描きのイラストに近づけることができるかを一から構成し、実験してみることにあつた。現段階では、手描きのイラストまでとはいかないまでも、写真には見えないようなものを作ることは成功したと考えている。また、今までの研究から、今後どのように研究を進めていけばよいかという見通しがついたと考える。

参考文献

- [1] 掌田津那乃, "Java プログラミングの教科書" (株) エクシードプレス, 2001 年刊
- [2] 大石進一, 牧野光則, "グラフィックス" 日本評論社, 1994 年刊
- [3] 穴戸 輝光 "DIB/Wave とグラフィック・音声処理実験室" < <http://www.sm.rim.or.jp/~shishido/post.html> >
- [4] Sun Microsystems "JavaTM 2 SDK, Standard Edition ドキュメント V1.4.0" < <http://sdc.sun.co.jp/java/docs/j2se/1.4/ja/docs/ja/index.html> >
- [5] 株式会社システム計画研究所 "ISP imaging-developers" < http://image-d.isp.jp/commentary/color_cformula/HLS.html >
- [6] 同志社大学工学部知的システムデザイン研究室 "輪郭を抜き出す" < <http://miki lab.doshisha.ac.jp/dia/research/person/shuto/research/0605/bibun.html> >
- [7] WIKIMEDIA FOUNDATION "フリー百貨事典『Wikipedia』" < <http://ja.wikipedia.org/wiki/イラストレーション> >