紅葉（こうよう）処理アルゴリズムの開発

加藤 智也
名古屋文理大学 情報文化学部 情報文化学科 はせがわ研究室

平成 15 年 1 月 27 日　提出

要旨

ある特定の目的の画像処理を行う際に、多機能の汎用のソフトウェアを用いて処理を組み合わせて実現しようとすると画像処理に関する専門知識が必要となる。一方、特定の目的に特化されたアルゴリズムを適用すれば、用途は限定されるが、画像処理アルゴリズムを理解する必要が無い。そこで今回、画像処理の専門知識を必要としない自動画像処理の１つとして、処理内容を「紅葉（こうよう）処理」に限定した画像処理アルゴリズムを実現し、風景画像に適用して評価した。ここで「紅葉処理」とは、木の葉が紅葉する以前に撮影した風景画像を、まるで紅葉を撮影したかのように変換する画像処理を言う。「紅葉処理」は、デジタル画像データから、紅葉させる葉の部分のピクセルを抜き出し、抜き出したピクセルの色を変換する処理である。実際のプログラムでは、色相の値によって処理対象となるピクセルを抜き出し、色相の値および彩度の値を変えることにより、風景画像に紅葉処理を施した。プログラムの作成にはＪＡＶＡ言語を使用してあるため、インターネット上で誰でもこの「紅葉処理」アルゴリズムを利用することができる。このアルゴリズムを適用して、特定の画像については効果的な紅葉処理を施すことができた。ただし、現段階では、すべての風景画像について必ずしも効果的に葉の色を変えられる訳ではなく、場合に応じてピクセル選択範囲を変えて対応する必要がある。提案した「紅葉処理」は、今後、アルゴリズムの改良により適用範囲が広がれば、芸術表現の幅を広げるだけでなく、商用的にも応用価値が高いと考えられる。

１．はじめに

近年、デジタルカメラの普及はめざましく、デジタルカメラで撮影した画像データに処理を行う画像処理ソフトウェアは現在までに数多く開発されている。例をあげれば、Adobe 社の Photoshop、フリーウェアとして提供されている Gimp 等があるが、これらのような汎用のソフトウェアを使用するためには一定の知識や技術が必要である。私の経験からも知識を持っていない人が使うのは困難であると感じた。一方、特定の用途に限定された画像処理機能として、ユーザが処理アルゴリズムを理解する必要がない形で提供されているものもある。例えば、画質の改善（自動コントラスト改善、自動色調補正など）やアート効果（ポスタリゼーション、絵画調処理など）である。

そこで今回、作業の種類つまり機能は限定し、そのかわりに特別な知識・技術が無くても使用することができる画像処理アルゴリズ

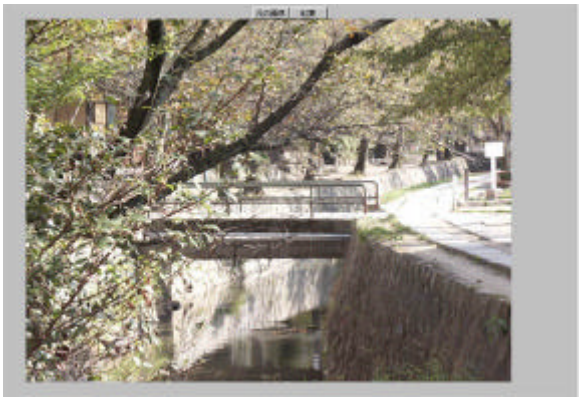
ムの１つとして「紅葉（こうよう）処理」の機能を実現するソフトウェアを試作して評価した。このソフトウェアが行う画像処理は緑の葉を紅葉に変えるというもので、葉が紅葉する以前に撮影した画像を、まるで紅葉を撮影したかのように変換することを目的としている。

このような画像処理は１枚の風景写真の季節を変える処理であり、デジタル画像の芸術表現の可能性を広げ、風景映像のロケーションが季節を選ばず実行できるようになる点で、商用的にも応用価値が高いと考えられる。

２．紅葉処理について

２.１．紅葉処理の適用

今回作成した紅葉処理ソフトウェアの適用例を図１に表示する。元の画像（図１(a)）に処理を施すと、紅葉した風景画像（図１(b)）になる。



(a)元の画像

(b)紅葉処理後の画像

図１．紅葉処理の適用例（１）

２.２．プログラムの実際

２.１.で示したソフトウェアを作成するにあたって今回は JAVA 言語を使用した。JAVA を使用するメリットとしては、①作成したソフトウェアがインターネット上で利用可能なこと、②動作環境がプラットフォームに依存しないため Windows、Macintosh、UNIX ワークステーションでも動作可能なこと、が挙げられる。

具体的な処理として、まず処理をしたい画像のピクセル値を獲得するために JAVA のクラスライブラリーPixelGrabber を使用した（ソース１参照）。

獲得したピクセル値は図２のように３２bit 整数値として保持されており、0bit 目から７bit 目までは青色のデータ、８bit 目から１５bit 目までは緑色のデータ、同様に赤色のデータ、アルファ透明度のデータがそれぞれ８bit ずつ格納されている。

図２．ＪＡＶＡのデフォルトカラーモデル
（出典：文献１）

```
PixelGrabber pg = new PixelGrabber(imageInput,0,0,WIDTH,HEIGHT,pixelInput,0,WIDTH);
```

ソース１．紅葉処理（ピクセル値の獲得）

```
alpha = (pixelInput[y*WIDTH+x]&0xff000000);
Red = (pixelInput[y*WIDTH+x]&0x00ff0000)>>16;
Green = (pixelInput[y*WIDTH+x]&0x0000ff00)>>8;
Blue =(pixelInput[y*WIDTH+x]&0x000000ff);
Color.RGBtoHSB (Red,Green,Blue,hsb);
```

ソース２．紅葉処理（ＨＳＢカラーモデルへの変換）

今回の処理は、色相の変換[2) 3)]を目的としているため、扱いやすくするためにＲＧＢカラーモデルをＨＳＢカラーモデルへと変換する。ＨＳＢカラーモデルとは、H(Hue＝色相)、Ｓ（Saturation＝彩度)、Ｂ（Brightness＝明度）で表現されるカラースペースで、Ｈ,Ｓ,Ｂはそれぞれ 0.0～1.0 の値で表される。前述（**ソース１**参照)のように PixelGrabber によって獲得したピクセル値を、alpha、Red、Green、Blue に分け RGBtoHSB を使用してＨＳＢカラーモデルへ変換した（**ソース２**参照)。

次に、葉以外のピクセル（橋や道など）の色相はそのままで、葉だけを色相変換するために、紅葉処理を施したいピクセル、つまり緑色をしたピクセルを抜き出す。その緑色の範囲は値を変えて何度か試して決定する必要がある。例えば、**図１(a)**の画像に対しては、0.1≦色相≦0.4 が適当であると判断した。

次に、抜き出したピクセルの色相を変換するとともに彩度を増す処理も施した。実際にはソース３に示すようなプログラムを記述して上記の抜き出しと画素値の変換処理を実現した。hsb[0]＝H（色相)、hsb[1]＝S（彩度)、hsb[2]＝B（明度）となっている（**ソース３**参照)。

さらに、実際の実行プログラムでは JAVA で記述したプログラムをアプレットとして HTML ファイルから呼び出して実行する形とした。実行時には、“元の画像”ボタンを押すことで元の画像を、“紅葉”ボタンを押すことで処理後の画像を表示できるようにした。

２.３．紅葉処理の適用結果

ボタンを押すだけで、画像の葉が紅葉し、秋に撮影したような画像にすることができる画像処理プログラムが完成した。その適応結果は**図１**に示した通りである。これなら、特別な知識を必要とすることなく、誰でも葉が紅葉したかのような画像処理を行うことができる。

次に、プログラムは変更せず画像を別のも

```
if((0.1<=hsb[0]) && (hsb[0]<=0.4)){
    pixelOutput[index++]=(alpha|Color.HSBtoRGB(hsb[0]+0.88,hsb[1]+0.2,hsb[2]));
}else{
    pixelOutput[index++] = (alpha|Color.HSBtoRGB(hsb[0],hsb[1],hsb[2]));
}
```

ソース３．紅葉処理（ピクセルの抜き出しと、色相・彩度変換）

のに入れ替えた場合を図３に示す。この場合も、全体として紅葉したかのような秋らしい画像になっているのではないかと思う。しかし、詳細に見ると、一部に色が変換されない葉もあることがわかる。

(a)元の画像

(b)紅葉処理後の画像

図３．紅葉処理の適用例（２）

数例の風景画像に適用した結果、同じような色相をもつ葉の画像であれば、ほぼ同じように紅葉の効果を得ることができるとわかった。ただし、画像によって葉の緑色が異なるため、変換されない葉ができる場合があることもわかった。

３．まとめと今後の課題

「紅葉処理」アルゴリズムを試作し評価した。このプログラムでは、色相の値によって指定したピクセルの色を変えることにより、特定の画像（例えば図１）については効果的な紅葉処理を施すことができた。その際に、汎用ソフトウェアのように画像処理に関する知識や技術を要求されることはない。しかし、固定的なアルゴリズムでは、すべての風景画像について効果的に葉の色を変えられる訳ではない（例えば図３）。この場合は、現段階では、プログラム中のピクセル選択範囲（ソース３参照）を変えて対応する必要がある。

また、このアルゴリズムでは、風景写真内に葉の色と同じ緑色をしたピクセルがある場合、葉なのか、そうではないのか識別することはできない。そこを識別することができれば、よりリアリティのある作品を制作できる画像処理になると思われる。

さらに、夏から秋だけでなく、四季の相互変換が出来れば、より用途が広がるだろう。また、今回の紅葉の色は、私が紅葉らしいと感じる色を実現したものであり、必ずしも実際の紅葉の色を再現したわけではなく、多くの人が好ましいと思う紅葉の色というわけでもない。よりリアリティのある紅葉処理を実現するためには、今後、現実の紅葉を写した画像の特徴との比較をおこなったり、複数の人間による評価を受け、その結果をアルゴリズムに反映していく必要がある。

なお、今回制作した「紅葉処理」は、以下の Web ページから実行可能である。

http://www.nagoya-bunri.ac.jp/~hasegawa/LAB/WK2002/work.html

[参考文献]

1) Mark C. Chan 他著、舟木 将彦 訳:「Java プログラミング 1001Tips」,オーム社,(1997)
2) 磯博：「ディジタル画像処理入門」,産能大出版部,(1996)
3) 八木伸行:「C 言語で学ぶ実践画像処理」,オーム社,(1992)