2011 年度　卒業研究論文

画像処理ソフト ImageJ とプラグイン

岡山理科大学

総合情報学部

情報科学科

澤見研究室

I08I011　大森将司

I08I012　岡本貴之

目次

1. はじめに 2
2. ImageJ 3
2.1 プラグインとマクロ 3
2.2 ImageJ のソフトウェア構造 4
2.3 プラグイン例 5
2.3.1 画像の比較 6
2.4 実装されているプラグインの一例 7
3. 色温度 8
3.1 色温度の相対強度を求める式 8
3.2 色温度計算のプログラム 9
3.2.1 色温度変更した時の結果 10
3.3 様々な色温度の画像 11
3.4 各色温度に対応する白色光 12
4 まとめ 14
参考文献 15

1. はじめに

世の中には様々な機能を持った画像処理ソフトがある．その中でこのようなプログラムがどのような仕組みで動いていて，それが自分たちで作れるかということに興味を持った．そこで，本研究では画像処理ソフトとして ImageJ を取り上げ，プラグインの仕組みについて詳しく調べ，新たに ImageJ 用のプラグインを作成して評価する．

2. ImageJ

画像処理ソフト ImageJ はアメリカ国立衛生研究所(NIH)で開発されたオープンソースでパブリックドメインの画像処理ソフトウェアでもある．画像処理及び解析機能が豊富で，あらゆる場面で活躍するソフトウェアである．Java1.1 以降の仮想マシン上で動作し，MacOS X，MacOS，Windows，Linux，ザウルス用のソフトウェアが用意されている．基本機能として，8 ビット，16 ビット，32 ビット画像を編集，解析，画像処理，保存，印刷することができる．

ImageJ は例として，デザイン，医学，工業，軍事産業などに利用される．

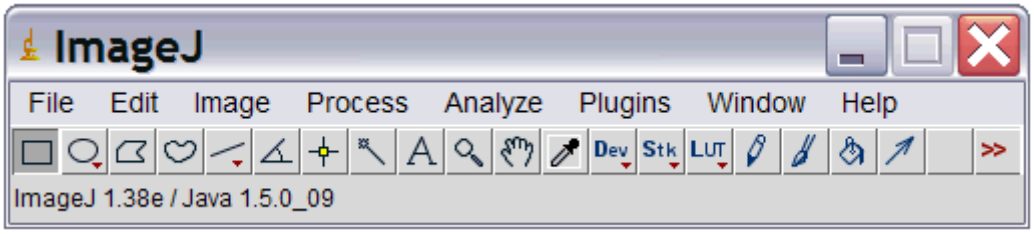

図 1 ImageJ のメニュー画面

2.1 プラグインとマクロ

プラグインとマクロは，標準に搭載されてない機能を拡張するのにあたって必要となるプログラムコードである．これを ImageJ へ実装するにあたっては，プラグインは Java プログラム言語で書かれており，クラスファイルにコンパイルを必要とする．マクロは Java に似た ImageJ のマクロ言語で書かれており，テキストファイルで保存する必要がある。この二つを比較すると，プラグインは動作が速くて融通が利き，マクロは書きやすくデバックも楽という特徴がある．本研究ではプログラム開発が容易で，動作が速く，動作環境が Java 仮想システム(JVM)なので，Windows や Mac OS などの他の OS に渡しても実行できるということからプラグインを選択し，作成した．

2.2　ImageJ のソフトウェア構造

ImageJ の内部構造は Java のコアシステムに基づいて，以下のようになっている．

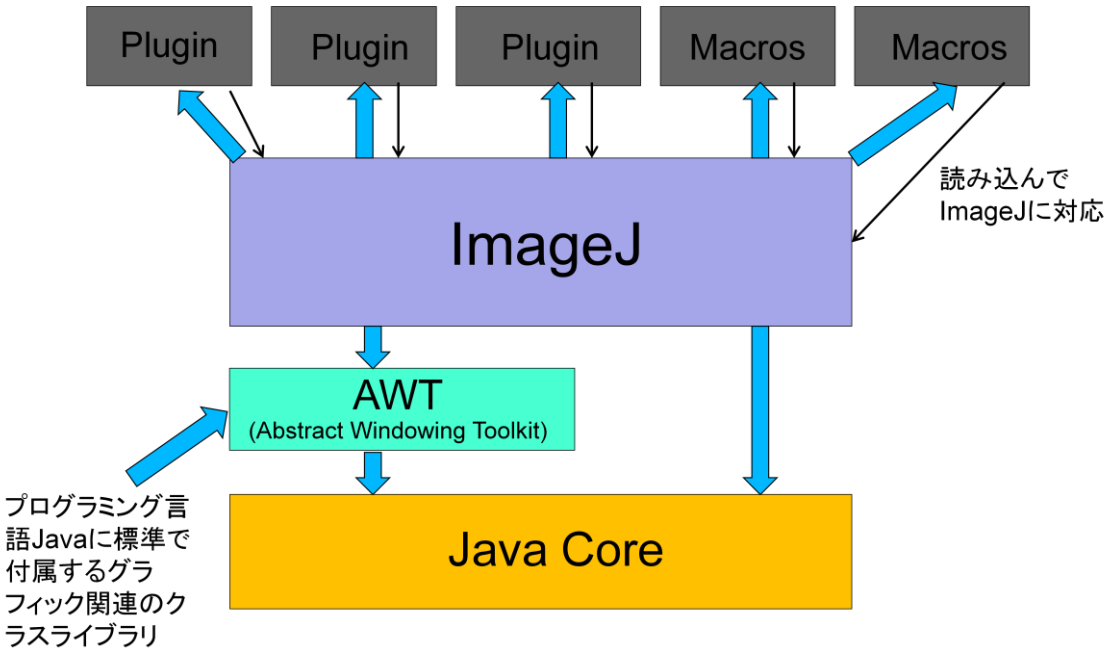

図 2 ImageJ のソフトウェア

Java モジュールであるプラグインと Java モジュール相当のマクロを読み込み ImageJ の機能を拡張することができる．そして，読み込むことで新規に実装したプラグインかマクロを用い，必要とする機能を選択して，画像を処理することができる．

2.3 プラグイン例

画像の輝度値を変更することで明るくするプログラムを例として取り上げ, プラグインの解説をする. 次に示すプログラム本文を作成し, Plugins フォルダに保存する. この際, ファイル名に（1 つ以上の）アンダースコア "_" が使われていれば, Plugins メニューのリストの一番下に新規プラグインとして追加される. ここでは, sample_Filter_Plugin　という名前で保存している.

```
import ij.ImagePlus;//最初から定義されているファイルを読み込む
import ij.process.ImageProcessor;
import ij.plugin.filter.PlugInFilter;
public class sample_Filter_Plugin implements PlugInFilter{
	public int setup(String arg, ImagePlus im) {
		return DOES_RGB; //RGB 画像で動作
	}
	public void run(ImageProcessor ip) {
		int[] pixels = (int[]) ip.getPixels();//ピクセルの値を取得する
		for (int i = 0; i < pixels.length; i++) {
			int c = pixels[i];
			// RGB 成分に色のピクセルを分割する
			int r = (c & 0xff0000) >> 16;
			int g = (c & 0x00ff00) >> 8;
			int b = (c & 0x0000ff);
			//色を 50 足して変更する
			r = r + 50;
			if (r > 255) r = 255;
			if (r < 0)    r = 0;
			g = g + 50;
			if (g > 255) g = 255;
			if (g < 0)    g = 0;
			b = b + 50;
			if (b > 255) b = 255;
			if (b < 0)    b = 0;
			//ピクセル配列に入れカラーピクセルを組み立て直す
			pixels[i]= ((r & 0xff)<<16 | ((g & 0xff)<<8) | b & 0xff);
		}
	}
}
```

プログラム 1 画像の輝度値の変更

2.3.1 画像の比較

プログラムの実行例を以下に示す(プログラム 1)．元の画像に比べてプログラムの実行後の方が明るくなっている．その結果，空が明るくなったり，影になっていた部分が見えやすくなることがわかる．

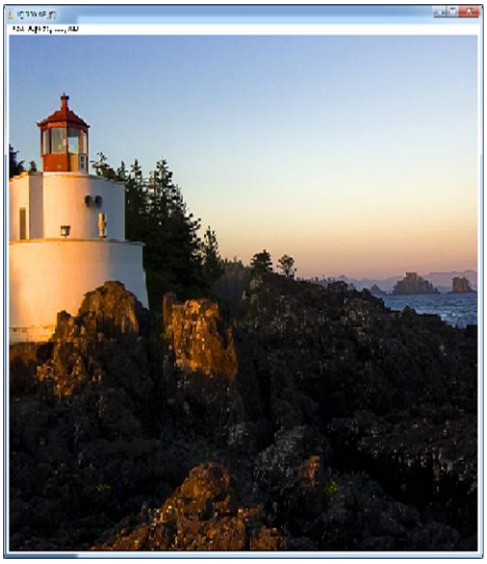

図 3 元の画像

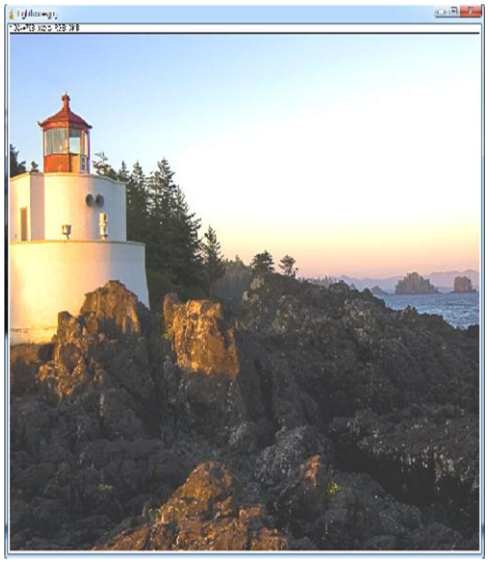

図 4 実行後

2.4 実装されているプラグインの一例

以下に標準で実装されているプラグインの一例を示す.

表 1 プラグインの一例

プラグイン	機能
3D	3D Viewer
Analyze	Grid
	Measure RGB
Examples	Scripts
Graphics	RGB Profile Plot
Stacks	Measure Stack
Tools	Display Pixel Values
	ImageJ Updater
My plugin	Iroondo 2900k

ImageJ の Plugins メニューにはあらかじめ様々なプラグインが実装されている．例えば，3 次元データで作成されたコンピュータグラフィックスの仮想空間を web 上で表示する 3D や，グリッド線を指定したり RGB の測定をする Analyze などのプラグインがある(表 1).

今回作成した色温度に関するプラグインは，My plugin として保存されている.

3. 色温度

色温度とは，照明の光の色を表すものである。ディスプレイやカメラ，照明器具など，様々な製品で色の基準とされており，色温度を表す単位は絶対温度 K(ケルビン)で示される．この絶対温度には，基準値となる代表的なものが複数あり，2900K，5500K，6500K，9300K などがある．

表 2 主な色温度と目安

色温度	目安（太陽光）	目安（人工光源）
2900K	夕日	家庭内電球
5500K	晴天	ストロボ光
6500K	曇り空	PC ディスプレイ
9300K	晴天の日陰	TV

3.1 色温度の相対強度を求める式

プランクは熱輻射に関する以下の公式を示している．

$$I = \frac{8\pi hc}{\lambda^{5}\left(e^{\left(\frac{hc}{kt\lambda}\right)} - 1\right)}$$

h はプランク定数 6.6260755E-34(Js)，k はボルツマン定数 1.380658E-23(J/K)，c は光速 2.99792458E+08(m/s)である．これにより求めたスペクトル強度 I と絶対温度 t(ケルビン;K)との関係を，以降では緑色(G)の値を 1 に正規化して利用する．

3.2 色温度計算のプログラム

以下に，色温度を計算するためのプログラムの一部をを示す.

```
int nIrondo = 2900;//色温度の選択
double h = 6.63E-34;// プランク定数
double k = 1.38E-23;//ボルツマン定数
double c = 3.00E+08;//光速
double rr = 7.00E-07;//r の波長
double gg = 5.46E-07;g の波長
double bb = 4.36E-07;b の波長
double z = 8*h*c*3.14;
double zr = Math.pow(rr,5)*Math.exp((h*c/(k*nIrondo*rr)-1));//r の値の計算
double zg = Math.pow(gg,5)*Math.exp((h*c/(k*nIrondo*gg)-1));//g の値の計算
double zb = Math.pow(bb,5)*Math.exp((h*c/(k*nIrondo*bb)-1));//b の値の計算
double ir = z/zr;
double ig = z/zg;
double ib = z/zb;
double rg = ir/ig;//g を正規化した時の r の値
double bg = ib/ig;//g を正規化した時の b の値
```

プログラム 2 色温度計算のプログラム

ファイルの読み込みや RGB 成分に色のピクセルを分割することなどは，前記の画像を明るくするプログラム(プログラム 1)と同じである．プランク定数やボルツマン定数を用いて，計算により色温度の相対強度を求めている．そして，g を 1 とした時の r と b の値を求める．

3.2.1 色温度変更した時の結果

実際に ImageJ 用に作成したプラグイン(プログラム 2)を使って画像の色温度を変更した(図 5). 元の画像と比べ，色温度を 2900K に変更した画像は緑色はそのままで赤色は一倍強く，青色は一倍弱くなるため，夕方のような印象を受ける．このことから，色温度が高くなるにつれ徐々に青みが，逆に低くなると赤み帯びてくること分かる．

元の画像(6500K)

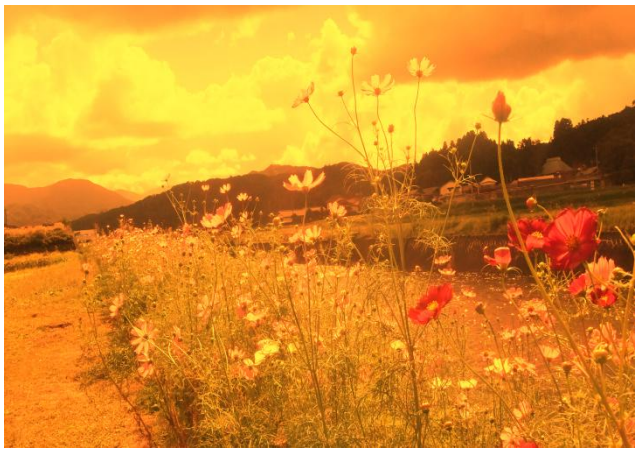

図 5　色温度 6500K から 2900K への変更

3.3 様々な色温度の画像

先に記した目安となる色温度 2900K，5500K，6500K，9300K に対応する画像を作成した(図 6-9).

色温度 2900K では夕日のような印象，色温度 5500K だと晴天時のような印象，色温度 6500K だと曇り空の時の印象，色温度 9300K だと晴天時の日陰のような印象を受ける.

図 6 色温度 2900K の画像

図 7 色温度 5500K の画像

図 8 色温度 6500K の画像

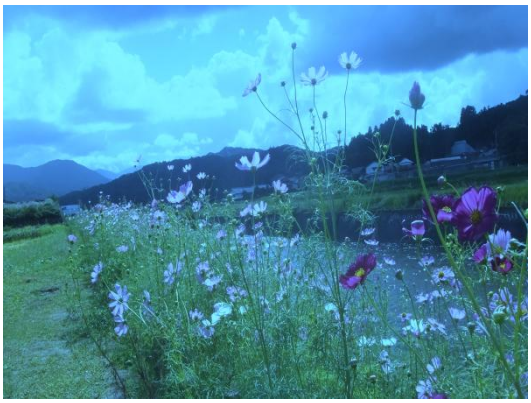

図 9 色温度 9300K の画像

3.4 各色温度に対応する白色光

色温度と三原色との定量的な関係から，黒体副射を基準にして，特定の比率で RGB 値を組み合わせることにより，ある色温度における白色光を一意的に定義することができる．さらに，ある色温度を基準にして，別の色温度における白色光を RGB 値の相対的な比率により求めることもできる．

いくつかの代表的な色温度として 2900K, 5000K, 6500K, 9300K をそれぞれ基準にした場合, 2800K から 10000K の範囲の各色温度に対応する白色光がどのような色に感じられるのかを以下に示す(図 10-13).

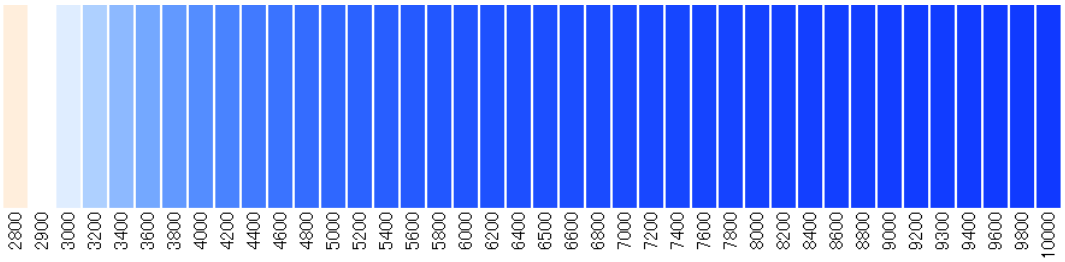

図 10 基準色温度を 2900K とした白色光

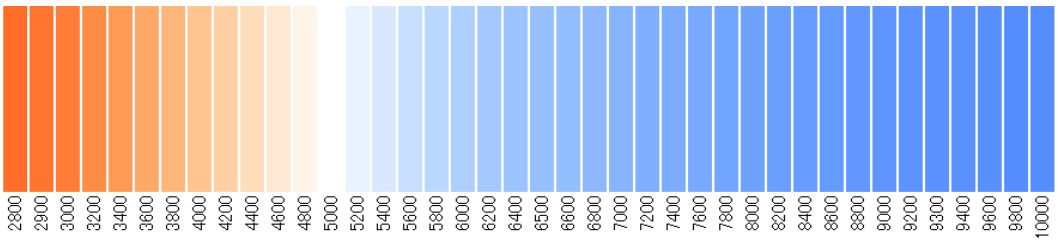

図 11 基準色温度を 5000K とした白色光

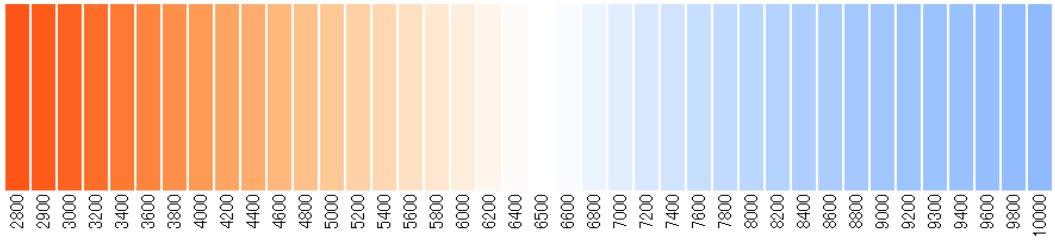

図 12 基準色温度を 6500K とした白色光

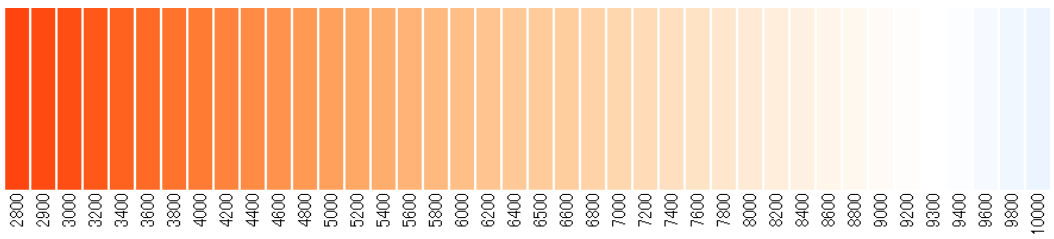

図 13 基準色温度を 9300K とした白色光

基準色温度が低い時に色温度を高くしていくと図 10 から図 13 の白色光は，より青みを増していき，基準色温度が高い時に色温度を低くしていくと白色光は，より赤みが増していくことがわかる．

4 まとめ

今回，画像処理ソフト ImageJ を使って画像の輝度を変更したり，色温度を変更するプラグインを作成し，予想していたものに近い結果を得ることができた.

色温度を高くすると画像の青みが増し，色温度を低くすると画像の赤みが増すことがわかった．しかし, 作成したプラグインには操作性の向上や, より正確な値を出すなど, まだ改善の余地はあると思う．そのことを踏まえて，今後はプログラムの内容について検討していきたい.

参考文献

[1] 色温度について
http://cafe.mis.ous.ac.jp/sawami/%E9%BB%92%E4%BD%93%E8%BC%BB%E5%B0%84.PDF

[2] java
http://java.com/ja/

[3] ImageJ 日本語情報
http://wiki.livedoor.jp/imagej/

[4] Wilhelm Burger/Mark J. Burge 2009 『Principless of Digital Image Processing』 Springer

[5] マイクロソフト windows7 ライブラリ「サンプルピクチャ」