Viewmat と花の色の世界

SHIMURA Masato

2015 年 6 月 12 日

目次

概要

花の色や自然素材で染色した古代色をネットで入手できる写真画像から *viewnmat* を用いて分解する。
花の色の分解から自然は実に複雑にできていて、点描派を超細密にしたような色彩の世界が現れる。

併せて画像ファイルを取り扱う *viewmat* は *QT* 版の *J8x* でどのように変わったか確認する。

はじめに

友人がユズリハの綺麗な画像をフェースブックにアップしていた。私も今春から目にした各種の花を 5 か月で 400 種近く写真に撮った。自然の織り成す多彩な色をカラーパレットに移し、グラフィックに用いることとしたい。

都で宮本武蔵と決闘した吉岡一門は、江戸初期に武士を捨て、都で染色を家業とした。黒の剣法染で有名だったようだ。この家が京で今も続いており、草木染など古式染を伝承している。当主吉岡幸雄氏は著書も多数あり、染の華麗な色に見とれるが決して RGB や CMYK などで数値化してはくれない。

*1

*2

日本固有色の RGB や CMYK の値はカラーブックなどで表されているが、これらの鮮

*1 CMYK シアン, マジェンタ, イエローなどでプリンターで用いられる

*2 武蔵との勝負は小説とは異なり吉岡側が優勢だったようだ

やかな色には及ばない。

そこで、ネットなどで入手できる写真のディジタルデータを解明することとした。

1　画像の形式

画像のフォーマットは天然素材の bmp と圧縮した gif,png,jpg などが良く使われている。

- bmp は圧縮していないので最高の素材だが、ファイルサイズが大きく、通信には不向き
- 圧縮後再び復元できるのは gif,png
- jpg は元に戻らないがファイルサイズが小さい。
- BPG は 2014 にフランス人プログラマが開発した jpg より効率的な圧縮方式

[*3]

また、写真の世界では本格的な RAW や TIFF が活躍している。RAW は一枚板の CCD や CMOS の R,G,B 用に割り当てた各ピクセルの画像エンジン通過前の生データで、カメラメーカー固有のソフトによって現像が必要なようだ。TIFF もきれいだが、非圧縮なのでデータが大きすぎ、net の世界向きではない。

[*4]

1.1　JPEG

ニコンの HP を見ていたら、JPEG にも 3 種類あり、$\frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ の圧縮比があるとのこと JPEG の非可逆圧縮の癖を探ってみよう。

カメラでシャッターを押してから SD カードなどに書き込むまでの間に画像エンジンが jpeg 圧縮を済ませてしまう。

JPEG は RGB でなく、YCrCb 変換を用いて輝度と色成分に分解して色成分を間引いたのち、離散コサイン変換（DCT) で空間周波数成分を分解した後、高周波成分を間引いて圧縮するようで、カラー TV の YUV 技術を受け継いでいる。

YCbCr 変換の一例

[*3] gif は圧縮方法で UNISYS 絡みのライセンスが懸念され、png が開発された。今では gif も問題が払しょくされ、再度利用されるようになっている。

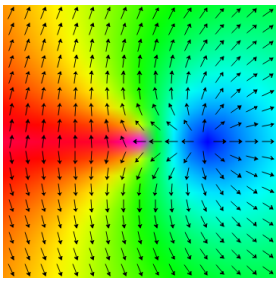
[*4] TIFF の JPEG 圧縮もある

$$\begin{pmatrix} Y\text{（輝度）} & Y = & 0.299 \times R & +0.587 \times G & +0.114 \times B \\ Cb & U = & -0.169 \times R & -0.3316 \times G & +0.500 \times B \\ Cr & V = & 0.500 \times R & -0.4186 \times G & -0.0813 \times B \end{pmatrix}$$

JPEG は見た目にはそれ程影響のない「そこそこの色」に変換されているといえる。不可逆変換なので、プログラムで強引に戻すか、そこそこを受け入れるかである。解析して分ったことは、自然は複雑で花の色は点描の多彩なグラディエーションであって、このRGB とは定まらないということである。

2　Viewmat の機能

viewmat はマトリクスのテーブルを白黒またはカラーパレットにしたがって表示するもので、複素数も取り扱える優れものである



- *0,1* のマトリクスの白黒表示
- 色番号のマトリクスのピクセル（画像）表示
- 複素数の方向の矢印表示
- 色番号のマトリクスを見る。

しかし、*PhotoShop* や *GIMP* のような高機能画像ツールではない。

2.1　viewmat の関数一覧

1. パッケージのロード
   最初の準備 *require 'viewmat'*
2. チュートリアル
   Help → *Studio* → *Labs* → *Graphics* → *viewmat* にチュートリアルが入っている。
3. 関数一覧

readpng

---

viewmat

viewrgb

---

*savemat_jviewmat_*

4. ユーザー定義関数 *viewmat_qt0.ijs*

trim_picture

cut_picture

find_rgb

2.2 画像の読み込み

J の定番 *toucan.bmp* が *addons/graphics/bmp* に入っている。*WIN8* に入っているツールで *png*、*jpg* に変換する。

J602 では *viewmat* の *jpeg* 読込みは *C.Reiter* の *addon* が必要であったが、*J803* ではその機能が組み込まれ、*jpeg* 画像が読込める。

[*5]

type	*name*	*size*	*tool*
bmp	*toucan.bmp*	*29,878*	*original*
png	*toucan.png*	*13,753*	*WIN8*
jpg	*toucan.jpg*	*14,562*	*WIN8*
jpg	*toucan_ps.jpg*	*9,457*	*PaintShop*

bmp から *jpg* への変換にはツールによる差がある。

2.3 画像要素の読込み

- *readpng* で *toucan* の画像は変換後の *png,jpg* も全て読み込める

[*5] Tex でも BoundingBox を記述すれば jpg が扱える。

```
a0=. readpng 'c:/temp/toucan.bmp'
a1=. readpng 'c:/temp/toucan.png'
a2=. readpng 'c:/temp/toucan.jpg'
a3=. readpng '/temp/toucan_ps.jpg'
```

- *viewmat* で読込み後の要素のサイズは全て同じであった。
  $ a0
  *144 200 NB. h(*縦） *w(*横）である。
- 読込んだ画像を見る。見かけでは差異は判らなかった。
  viewrgb L:0 a0;a1;a2;a3
- 読み込み後の *bmp(a0)* と *png(a1)* は全く同じもの
- *bmp(a0)* と *jpg(a2),jpg(a3)* は夫々異なる。
  $a0 = a1 \neq a2 \neq a3$

jpg は見た目ではわからない「そこそこ」の色に変換している。

2.4 色の取り出し

QT 版の *J803* は *J602* と色番号の付け方が全く異なる。*QT* ではマイナス記号を用いた別個の進法を用いている。

BMP .

- *J803*

```
   (<40;10) { a0
_59100

 |.256 256 256 #: (<40;10) { a0
36 25 255
```

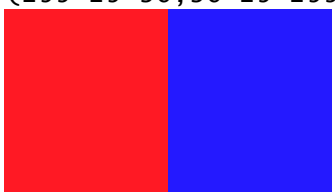
- *RGB* に変換する方式は *J602* と同じであり、*RGB* は同じ数値になる。
- *J6* は色番号を復号できる

```
   (<40;10) { a0
16718116
   |.256 256 256 #: (<40;10) { a
36 25 255
```

```
   (|.36 25 255)#.~ 256 256 256
16718116
```

- 基底変換のとき *RGB* 各 *8* ビットの数を、$2^3 = 256$ の *0* オリジン *255* でなく、*256* を用いているが、理由は不明（*J6* も同じ）
- *RGB* に変換するときに `|.` で *GBR* → *RGB* に回転させている。

```
(255 25 36,36 25 255) viewmat 0 1
```



JPEG .

- *JPEG* は不可逆圧縮なので、そこそこの色に戻る。

```
   (<40;10){a2
_60123
```

```
   |.256 256 256 #: (<40;10) { a2
37 21 255
```

- *RGB* と *JPEG* の比較。目では区別できない

```
 (36 25 255,: 37 21 255) viewmat ?20 20 $ 2
   savemat_jviewmat_ '/temp/compare_color0.jpg'
```

2.5 画像のセーブ

J803 では *jpg* もセーブできる。

```
savemat_jviewmat_ '/temp/complex_arrow.jpg'
```

3　花の画像で

ネットで手に入る花の画像は殆どが *jpg* である。

3.1　ゆずりは

RGB に戻すスクリプトを作成する

```
find_rgb=: 3 : '~. (|."1) 256 256 256 #: ;@> y'
```

色分解は次の *4* ステップで行う

1. 読み込み　　*readpng*
2. １次トリム　　*trim_picture*
3. 中心部分の抽出　　*trim_picture*
4. *RGB* 分解　　*find_rgb*

- *jpg* の読み込みとサイズの確認。

```
   c0=. readpng '/temp/yuzuriha.jpg'
   $ c0
640 960        NB. wh=960 640 である
```

- 一次トリム
  狙った花一輪を抽出する。落とすピクセル数を左右上下の順に指定する。

```
trim_picture=: 4 : 0
NB. x is cut-down pixel of  left right  upper floor
'L0 R0 U0 F0'=. 1 _1 1 _1 * x
R0}."1  L0}."1 F0}. U0}. y
)
```

見ながら狙った一輪を抽出する

```
 viewrgb    a1=.  200 450 300 150 trim_picture a0
   savemat_jviewmat_ '/temp/yuzuriha_trim0.jpg'
```

この段階で $wh = 310 \times 190 = 58900$　ピクセルある。

```
   $   a1
190 310
```

- *2* 次トリム。色を抽出する小区画を切り出す。10×10 では９０色近くが抽出されるのでできるだけ小さくとる

```
   viewrgb  a2=. 280 675 455 180 trim_picture a0
   $ a2
5 5
```

5×5 の区画を切り出した。

- 色の抽出。色は R を先にマトリクスソートを行っている。

```
   find_rgb=:3 : 0
```

```
tmp=. y
if. 1= # y do. tmp=. > y end.
/:~ ~. (|."1) 256 256 256 #: ; tmp
)
```

- *24/25* 色が抽出された。

```
find_rgb a2
 0 191 76
 1 192 77
 2 189 79
 2 190 78
 2 191 76
 2 194 77
 3 189 77
 3 193 82
 3 194 79
 3 195 78
 4 188 80
 4 195 80
 4 196 79
 5 188 78
 5 195 83
 7 194 85
 7 195 83
 7 196 87
 8 198 87
 9 197 92
10 194 88
10 197 95
11 197 91
12 197 93
```

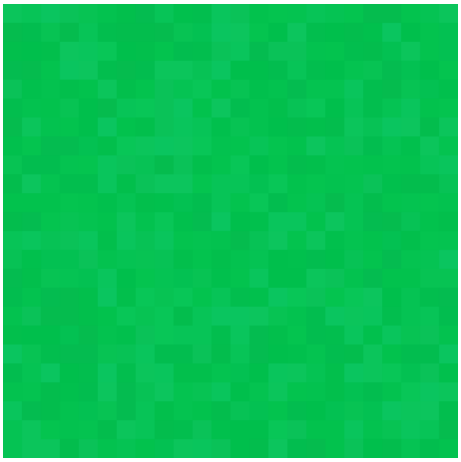
- 譲葉のカラーテーブル

```
color_yuzuriha=. find_rgb a2
```

```
  color_yuzuriha viewmat ?24 24 $ 24
```

```
savemat_jviewmat_ '/temp/color_yuzuriha.jpg'
```

- カラーテーブルを *24* 色の乱数で見る。



5 × 50 のピクセルで譲葉の色を抽出したところ、*24* 色が出てきた。100 × 100 では *90* 色程度抽出できる。自然界の花は、スラーやシニャック達点描派よりもずっと精密に色を微妙に連続に変化させている。ディジタルで離散抽出すると色数になるようだ。用途に応じ抽出範囲と色数を決めればよい。*gl2* のグラフィックスでは、べた塗りの *glpaint* にかえて *glpixel* を用いなければならない。

3.2　ゲラニウム

- *readpng*
  a1=. readpng '/temp/geraniumu.jpg'
- 先ず中心部のトリムを行う。

```
$ a2=. 250 200 150 150 trim_picture a1
```

```
46 30
```

- 大胆にトリミングを行う

```
    $ a3=. 260 210 170 166 trim_picture a1
5 5
```

- *RGB* を求める

  480 *times*346 を　5 × 5 たとき *25* 色全ての色が残る。デジカメの *CCD* は精巧に色を拾うので、重複は少ない。

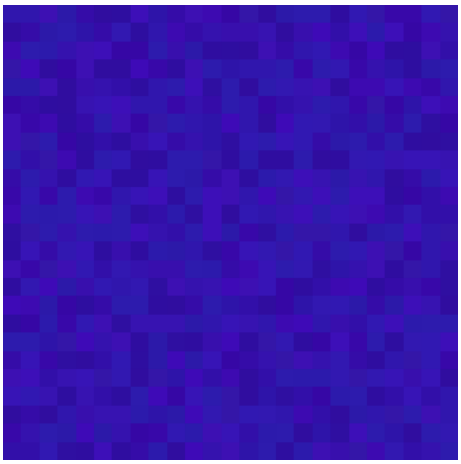
```
 /:~ 256 256 256 #:; a2
44 25 169                    52 22 168
44 26 172                    53 12 168
45 27 173                    53 18 174
45 27 173                    54  7 165
46 14 160                    55 19 177
46 20 169                    55 20 182
47 13 160                    56  8 169
47 15 161                    56 15 173
48 13 159                    61 10 177
48 22 173                    61 15 184
49 24 178                    61 16 179
50 15 169                    62 10 183
50 24 177
```

  ここで *BGR* → *RGB* という工程を省いた。[*6]

- カラーパレット

[*6] 入れると紫色系のパレットになるがその原因は理解できていない

```
color_geranium=. |. 256 256 256 #: ; a3
color_geranium viewmat ?25 25 $ 25
savemat_jviewmat_ '/temp/color_geranium.jpg'
```



まとめ

次のことが出来るようになった

- *jpeg* 画像の読み込み
- トリミングとセーブ
- 色の分解とカラーパレットの作成

解明されていない点

- *antibase(#:)* の基数（*0* オリジンで *255* か *256* か）
  viewmat の問題か、*JPEG* の問題か？
- *GBR → RGB* は必要か

付録 A　画像の分割

画像を細かく分割した後、目的のピースを取り出す方法。パレットを作成するには5×　５でも２５ピクセルあり、複雑な色の画像では８０－９０％の色が抽出される。 抽出した画像がどの部分かを探すのが大変なので、トリムの方が便利だ。

大きな画像から核心部分を取り出すため読み込んだ数値マトリクスを分割する。分割した画像を *viewmat* で再度見ると、各片がサムネイル画像になるのではなく、各一片は元サイズで粗く表示される。

- *J* ではマトリクスの分割には *Cut(;.n)* を用いる。インデックスは縦、横のカットする個所に *1* を立てる。
  [*7]

```
    (1 0 1 0 0;1 0 0 0 1 0 1) <;.1 i.5 7
+-----------+-----+--+
|0 1 2  3   | 4  5| 6|
|7 8 9 10   |11 12|13|
+-----------+-----+--+
|14 15 16 17|18 19|20|
|21 22 23 24|25 26|27|
|28 29 30 31|32 33|34|
+-----------+-----+--+
```

- 汎用の分割のスクリプト。画像のデータマトリクスも分割できる。

```
cut_picture=: 4 : 0
NB. ex.X=30  50 or ..100..
NB. 50 cut_picture y
NB. y is  reanpng bmp/png/jpg file
'TATE YOKO'=:  >: >.  x% ~ |. $ y NB.  $ a is 144 200
'W H'=:  |. $ y
ind0=. (1, x * >: i.TATE)e.~>:i. H
```

[*7] n=1 はカットのタイプ

```
ind1=. (1, x * >: i.YOKO)e.~>:i. W
(ind0;ind1)<;.1 y
)
```

- 画像を指定のピクセルの幅で分割する。（*50* ピクセルで分割したところ）

```
   $ L:0 ac0=. 50 cut_picture  a0
```

```
+-----+-----+-----+-----+
|49 49|49 50|49 50|49 51|
+-----+-----+-----+-----+
|50 49|50 50|50 50|50 51|
+-----+-----+-----+-----+
|45 49|45 50|45 50|45 51|
+-----+-----+-----+-----+
```

- 分割画像を同時に表示
  viewrgb L:0 ac0

References

J803 の入手　*jsoftware.com*　より
Script の入手 http://japla.sakura.ne.jp --> workshop 2015/06