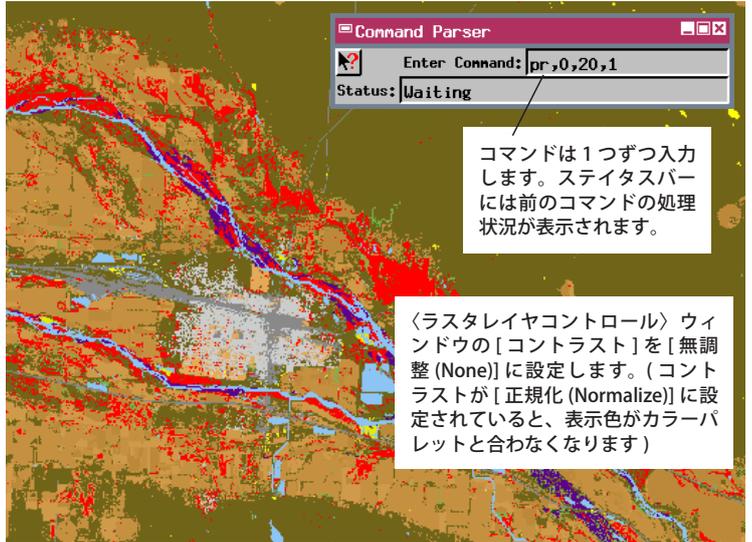


RGB 値によるカラー印刷

compar.sml スクリプトは 8 ビットラスタのセル値に定量的に色を割り当てる SML の例です。お使いのプリンタが色が安定して出力できる (カラーマネージメントを行っている) のであれば、必要な色をカラーの RGB 値入力によって正確に再現することができます。このスクリプトは、入力ラスタのセル値を指定し、それにカラーパレットの中から正確な RGB カラーを割り当てます。このスクリプトは個別の色を分類画像の各クラスに手早く割り当てたいという人のために書かれました。お使いのカラープリンタが色のドリフトを起こさなければ、ラスタデータが変わっても各クラスを一貫性のある色で出力することができます。また、このスクリプトではユーザの入力をどのように解析するかも説明しています。



このツールを選択すると、コマンドを入力するための〈コマンドパーサー (Command Parser)〉ウィンドウが開きます。ペイントコマンドは色番号で色を指定するよう求めます。スクリプトは、0 から 255 の色番号 (インデックス番号と呼ばれます) が書かれたテキストファイルから、指定された色番号を探し出します。各インデックス番号には、透明度と、赤・緑・青の各要素のパーセントが定義されています。このファイルは `t` コマンドを使って、任意のラスタのカラーパレットから作成することができます。テキストエディタやスプレッドシートを使って、このファイルを編集したりゼロから作成することもできます。入力するコマンドによって、スクリプトはこれらの色を全部一度にラスタのカラーパレットにコピーしたり (`lc` コマンド)、特定のラスタセルを色付けするために (`p` もしくは `pr` コマンド) 個々の色にアクセスしたい場合テキストファイルの色をロードします (`b` コマンド)。

ある色のインデックス番号 (または、その色で表わされるクラス) を見つける最適な方法は、標準カラーチャートを作成することです。カラーチャートがあると、各クラスを表わす色調の番号が利用しやすくなります。カラーチャートを参照すれば、必要とする色のインデックス番号を見つけ、compar.sml ツールを使って対応する全ラスタセルの色を塗り分けることができます。分類するクラスの数が多く、異なるクラスに対して似たような色を使用せざるを得ない場合に、

カラーチャートはより重要となります。分類するクラス数が少ない場合でも、明るさや色調が少しでも違っていると、クラス間の情報を伝達する上で見栄えが良く効果的です。カラーチャートを作るには多くの方法があります。一番簡単な方法は、クラスとそれに対応するカラー番号を一覧にすることです。左のチャートは TNTmips で作成したものです。

ラスタを色分けする前に、`b` コマンドを入力し、使用している全色のインデックス番号と各色要素の比率が書かれているテキストファイルを読み込みます。ラスタをペイントするには `p` コマンドと `pr` コマンドを使用します。これらのコマンドを使う上で必要なことは、クラスに対するカラー番号とペイントしたいラスタのセル値を知っておくだけです。

Group 1 - Group View 1			
View	Tool	LegendView	GPS Options
Woody Herbaceous Wetlands	0	Emergent Residential Wetlands	1
Herbaceous planted/cultivated urban/recreational Grasses	5	Herbaceous Fallow	6
Shrubland	10	Orchards/Vineyard	11
Deciduous Forest	15	Evergreen Forest	16
Barren Bare Rock	20	Barren Quarries	21
		Barren Transitional	22
		Low Intensity Residential	2
		High Int. Reside	
		Herbaceous Small Grains	7
		Herbaceous Pasture Hay	
		Grasslands	12
		Water	

CLASS: Emergent
Herbaceous Wetlands
Color # (index): 1
Red: 100%
Green: 0%
Blue: 0%
Transparent: 0%

サンプルの変数で使われるコマンド:

- `t` (カラーパレットをテキストファイルに出力する)
- `lc` (テキストファイルから新規の SML カラーパレットを作る)
- `b` (塗りつぶしに使う色のテキストファイルをロードする)
- `r,0,255` (セル値 0 から 255 を透明にする)
- `pr,0,20,1` (セル値 0 から 20 を #1 の色で塗る)
- `p,21,2` (セル値 21 を #2 の色で塗る)

TNTmips の SML (地理空間スクリプト言語) の使い方を解説するサンプルスクリプトが用意されています。参照用にスクリプトの主要部分を下に掲載しています。スクリプト全文は www.microimages.com/downloads/scripts.htm からダウンロードできます。

コマンド解析ツールスクリプト (compar.sml) の抜粋

```

proc OutputColorMap() {
  if (Group.ActiveLayer.Type == "Raster") {
    DispGetRasterFromLayer(rast,Group.ActiveLayer);
    cmap = ColorMapFromRastVar(rast);

    if (cmap.name == "") {
      PopupMessage("No ColorMap was found in raster object Paint function
      aborted\n Use lc command to save a csv file as a colormap in the raster
      object first\n or create one yourself");
      Command.value = "";
      Status.value = "OutputColorMap aborted";
      return;
    }

    myfile = GetOutputTextFile("c:/color.csv","Select file for output:","csv");
    fprintf(myfile,"%s,%s,%s,%s,%s\n","Index","Red","Green","Blue",
    "Transparency");
    Status.value = "Outputting Colormap";
    for i = 0 to 255 {
      mycolor = ColorMapGetColor(cmap,i);
      fprintf(myfile,"%d,%d,%d,%d\n",i,round(mycolor.red),
      round(mycolor.green),round(mycolor.blue),
      round(mycolor.transp));
    }

    fclose(myfile);
    Status.value = "Colormap outputted";
    Command.value = "";
  }
  else {
    Status.value = "Active Layer must be a raster w/colormap for this function";
    Command.value = "";
  }
}

```

t コマンドを入力すると OutputColorMap 関数が呼ばれます。ラスタのカラーマップを CSV ファイルにコピーします。コマンドライン構文: t

ラスタの下にカラーパレットを書き出す
ColorMapWriteToRastVar(Raster,colormap,Name\$,description\$)

ColorMapWriteToRastVar(rast,cmap,cmap.Name,cmap.Desc);

```

View.DisableRedraw = 1;
LayerDestroy(Group.ActiveLayer);
GroupQuickAddRasterVar(Group,rast);
Group.ActiveLayer.AllowDeleteLayer = 1;
View.DisableRedraw = 0;
View.Redraw(View);
Status.value = "Cell Painted";
Command.value = "";
}

```

この 6 行は編集したカラーパレットを使ってラスタを強制リロードする処理です

〈ファイル選択〉ウィンドウを開き出力ファイル名を指定

カラーパレット「cmap」中のインデックス i のカラーを返す

fprintf で CSV ファイルを出力します。内容例:

Index	Red	Green	Blue	Transparency
0	0	0	0	50
1	0	100	0	50
2	0	100	100	50
.
.
255	100	100	100	100

LoadColorMap 処理は lc コマンド入力時に呼び出されます。ラスタに対して新しいカラーマップを作ります。コマンドライン構文: lc

```

proc LoadColorMap() {
  local string line;
  myfile = GetInputTextFile("c:/colormap.csv","Select ColorMap file","csv");

```

入力ファイル名を選択するため
〈ファイル選択〉ウィンドウを開く

入力ファイルが有効かチェックする

```

line = fgetline$(myfile);
if ((NumberTokens(line,",") != 5) || ("Index" != GetToken(line,",",1))) {
  PopupMessage("This does not appear to be a valid colormap csv file\n
  The proper format is one row of labels then 256 \nnumeric lines of the
  form red,green,blue,transp \nwhere red,green,blue,transp are in the range
  0-100\nDisplay a raster with a colormap already and use the t command to
  export the colormap to a file to see how the file should look");
  Status.value = "Fatal Error ColorMap load halted";
  Command.value = "";
  return;
}

```

Status.value = "Loading Colormap";

```

for i = 0 to 255 {
  line = fgetline$(myfile);

```

入力ファイルが有効かさらにチェックする

```

if ((i > 0) && (StrToNum(GetToken(line,",",1)) == 0)) {
  PopupMessage("Bad Index Value Encountered while loading ColorMap
  file ColorMap load aborted");
  Status.value = "Fatal Error ColorMap load halted";
  Command.value = "";
  return;
}
mycolor.red = StrToNum(GetToken(line,",",2));
mycolor.green = StrToNum(GetToken(line,",",3));
mycolor.blue = StrToNum(GetToken(line,",",4));
mycolor.transp = StrToNum(GetToken(line,",",5));
ColorMapSetColor(cmap,i,mycolor);
}

```

GetToken(string\$, delimiters\$, tokenNumber) はトークン (区切り記号で分けられた文字列) を取得します

```

if (Group.ActiveLayer.Type != "Raster") {
  PopupMessage("Active Layer must be a raster object for this function");
  Command.value = "";
  Status.value = "Active layer must be a raster";
  return;
}
DispGetRasterFromLayer(rast,Group.ActiveLayer);
ColorMapWriteToRastVar(rast,cmap,"SML.color",
ColorMap created by SML script");

```

ラスタに対して新しいカラーパレット SMLcolor を書き出す ColorMapWriteToRastVar(Raster,colormap,name\$,description\$)

新しいカラーパレットでラスタを強制リロードするコードはここには出ていません。(Paint 関数中に同様のコードがあります)

```

Status.Value = "ColorMap saved as SML.color in raster object";
PopupMessage("Colormap was saved as a SMLcolor colormap under raster
object\nyou must select this colormap to see the changes\nif you were
displaying the raster with a different colormap\nThis is the only function
that behaves this way\nAll other function modify the most recently used
colormap");
Command.value = "";
}

```

以下は、ここに出ていない処理の外で宣言されているグローバル変数です。
class RASTER rast; class FILE myfile; array ablu[256];
class ColorMap cmap; array ared[256]; array atransp[256];
class Color mycolor; array agreen[256]; numeric hascolorarray;

```

proc Paint() {
  if (NumberTokens(Command.value,delim) != 3) {
    Status.value = "Not enough parameters for Paint function";
    return;
  }

  if (!hascolorarray) {
    PopupMessage("Use code b (SetColorArray function) to load a colormap
    to use first");
    Status.value = "Paint function aborted";
    Command.value = "";
    return;
  }

  local numeric cellvalue;
  local numeric colnumber;
  cellvalue = StrToNum(GetToken(Command.value,delim,2));
  colnumber = StrToNum(GetToken(Command.value,delim,3));
  if (Group.ActiveLayer.Type != "Raster") {
    PopupMessage("Active Layer must be a raster object for this function");
    Command.value = "";
    Status.value = "Active layer must be a raster";
    return;
  }

  DispGetRasterFromLayer(rast,Group.ActiveLayer);
  cmap = ColorMapFromRastVar(rast);
  if (cmap.name == "") {
    PopupMessage("No ColorMap was found in raster object Paint function
    aborted\n Use lc command to save a csv file as a colormap in the raster
    object first\n or create one yourself");
    Command.value = "";
    Status.value = "Paint function aborted aborted";
    return;
  }

  mycolor.red = ared[colnumber+1];
  mycolor.green = agreen[colnumber+1];
  mycolor.blue = ablu[colnumber+1];
  mycolor.transp = atransp[colnumber+1];

  ColorMapSetColor(cmap,cellvalue,mycolor);

```

p コマンドを入力すると呼び出されるペイント処理。カラーパレットの 'セル値' を配列の 'インデックス' 値にセット。コマンドライン構文: p, セル値, インデックス

'hascolorarray' は初期状態で false が設定されます。b コマンド (SetColorArray 関数) が実行されると true が設定されます。

アクティブレイヤからラスタ 'rast' を取得

ラスタのカラーパレットを返す

配列変数 (ared, agreen, ablu, atransp) には塗る色の値が入ります; 'colnumber' は 0 ~ 255 の値です; 配列は 1 から 256 の色数を持つので 1 を加える必要があります (SML の配列は 1 から開始します)

カラーパレットの色を設定する関数
ColorMapSetColor(colormap,index,color)