

Date: Tue, 08 Apr 2008 11:57:49 +0900  
From: Tsukasa NAKANO  
To: Junji Torii  
Cc: Shunji KASAMA  
Subject: t\_pvr+si\_pvr

---

とりいさま、

なかのです。そう言えば、slice シリーズで「32 (or 31) bit の壁」が切実な問題になっていた画素値ヒストグラムの取得と画素値の置換用のプログラムの改造も終わりました。

画素値ヒストグラムの取得と画素値の置換用プログラム  
対応する slice シリーズのプログラム名 : slicePVR

新しいプログラム名

2次元画像用 : t\_pvr

3次元画像用 : si\_pvr

書庫ファイル :

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/pvr.taz>

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/pvr.zip>

起動法

画素値ヒストグラムを取得する場合

```
t_pvr TIFF > output_2.txt
```

```
si_pvr directory nameFile > output_2.txt
```

画素値の置換を行う場合

```
t_pvr orgTIFF newTIFF < input.txt > output_3.txt
```

```
si_pvr orgDir nameFile newDir < input.txt > output_3.txt
```

ただし、

TIFF と orgTIFF : 処理を行う TIFF 画像のファイル名

directory と orgDir :

処理を行う 3次元画像 (スライス画像) のファイルを格納してあるディレクトリの名前

nameFile :

指定したディレクトリの下にある処理すべき画像ファイルの名前のリストを書き込んだファイル (name file) の名前 ("-" を指定すればすべてのファイルになる)。

**newTIFF** : 画素値を置換した TIFF 画像のファイル名

**newDir** :

画素値を置換したスライス画像のファイルを置くディレクトリの名前。

**orgDir** と同じでもよい。新しいものにする場合はあらかじめそれを自分で作っておく必要がある。なお、それぞれのスライス画像のファイル名は処理前のものと同じになる。

**output\_2.txt** :

画素値ヒストグラムのテキストデータを書き込むファイル。画像上に出現した画素値とその画素数の 2 個の整数値がタブコード区切りで各行に書き込まれている。

**output\_3.txt** :

画素値ヒストグラムのテキストデータを書き込むファイル。画像上に出現した画素値とその置換後の画素値、およびそれらの画素数の 3 個の整数値がタブコード区切りで各行に書き込まれている。

**input.txt** :

画素値の置換の指定のテキストデータ。旧来の **slicePVR** と同一の以下の 3 通りの行形式のいずれかで指定する。

old	new			← 画素値 old を new に置換
o1	o2	new		← o1~o2 を new に置換
o1	o2	n1	n2	← o1~o2 を n1~n2 に置換

改造のポイント

画素の個数に関する「31 bit の壁」を克服した。

画素値ヒストグラムの取得だけを行う場合の無意味なデータ入力を廃した。

取り急ぎ、

追記 (2014/11/20)

前記の書庫ファイル **pvr.taz** と **pvr.zip** には 32 ビット Windows 用の実行ファイル **t\_pvr.exe** と **si\_pvr.exe** を入れてあります。また、**gcc** (GNU C-compiler) を利用可能な計算機環境なら、書庫ファイルを展開したディレクトリに移動して "make" と端末入力すればソースコードから実行ファイル **t\_pvr** と **si\_pvr** をコンパイルできるはずです。

Date: Wed, 19 Nov 2014 13:01:51 +0900  
From: Tsukasa NAKANO  
To: "TSUCHIYAMA, Akira"  
Cc: 櫻間卓志, ToruMatsumoto, 道上達広, Yohei IGAMI, Aki Takigawa,  
Tatsuya OSAKO, 中村隆太, 山中 一平, MIYAKE, Kentaro UESUGI,  
Takashi Matsushima, カドカワトキユキ  
Subject: 輪郭線長や表面積の計測プログラム

---

つちやまさま、

なかのです。先日の E-mail で予告した計算機プログラム（4 個）を書きました。

On Mon, 17 Nov 2014 11:39:28 +0900 Tsukasa NAKANO wrote:

> On Sat, 15 Nov 2014 10:39:28 +0900 Akira TSUCHIYAMA wrote:

>

>> si\_stl\_A (or B) で個々の画像について表面積は求められますが、クラスタラベ  
>> リングした画像について、それぞれのクラスターでまとめて表面積求めたいと  
>> 思います。たしか、これできるようになっていた記憶があるのですが、以前の  
>> メールを探してもコマンドがうまく探せません。

>

> 実は si\_\* には表面積だけを測定するプログラムがありません。大昔に作った  
> sliceSIC と sliceMIC に相当するものを現在作成中なので、少々お待ち下さい。

今回書いたものは 2 次元画像用の t\_[s,m]ec と 3 次元用の si\_[s,m]fc です。

t\_ : TIFF ファイルとして与えられた 1 個の 2 次元画像を処理する。

si\_ : 複数のスライス画像から構成される 3 次元画像を処理する。

s : 1 種類の辺もしくは面だけを対象とした処理を行う。

m : 複数種類の辺もしくは面を対象とした処理を行う。

ec : edge counter

fc : face counter

これら 4 個のソースコードと (32 bit) Windows 用の実行ファイルなどを以前に紹介した t\_pvr や si\_pvr 用の書庫ファイルに入れておきました。今回書いたプログラムも以前のものと同まったく同じ方法でインストールできます。

<http://www-bl20.spring8.or.jp/~sp8ct/tmp/pvr.pdf>

新しいプログラムそれぞれの機能や使用法は以下の通りです。

#### t\_sec と si\_sfc

##### 機能

画素値の値域の指定によって 2 値化・識別した 2 次元画像上の物体像の輪郭線長（物体像と見なした画素とそれ以外のものに挟まれた画素の辺の総数）や、3 次元画像上の物体像の表面積（面の総数）を計測する。

##### 起動法

```
t_sec  TIFF  rangeList
si_sfc  directory  nameFile  rangeList
```

##### 起動パラメータの説明

**TIFF** : TIFF 形式画像のファイル名

**directory** と **nameFile** :

これらのパラメータで 3 次元画像を構成するスライス画像群を指定する。

**rangeList** : 物体像と見なす画素値の値域を指定する文字列

標準出力に書き出される 2 もしくは 3 個のテキストデータ（非負の整数値）

1 番目の値 : x 軸に垂直な辺もしくは面の総数

2 : y 軸に垂直な辺もしくは面の総数

3 (si\_sfc のみ) : z 軸に垂直な面の総数

#### t\_mec と si\_mfc

##### 機能

指定した値域の画素値を保持している画素のペアに挟まれている 2 次元画像上の画素の辺の総数や、3 次元画像上の面の総数を計測する。

##### 起動法

```
t_mec  TIFF  rangeList1  rangeList2
si_mfc  directory  nameFile  rangeList1  rangeList2
```

##### 起動パラメータの説明

**TIFF** : TIFF 形式画像のファイル名

**directory** と **nameFile** :

これらのパラメータで 3 次元画像を構成するスライス画像群を指定する。

**rangeList1** と **rangeList2** :

計測対象とする辺や面を挟む 2 個の画素の値の値域を指定する文字列。指定された値域の値の組み合わせのすべてを計測対象にするので、これら 2 個の並びの順は任意だが、実用的にはこれらに排他的な値域を指定すべきである（そうしないと同一の値の画素から構成される「物体像」の内部の辺や面の総数まで計測することになる）。

標準出力に書き出される各行に並ぶ4もしくは5個のテキストデータ

1 番目と 2 番目の値 (i と j) :

辺や面を挟んでいる画素の値。  $0 \leq i \leq j$  かつ下記の辺や面の総数すべてが 0 でないデータ行が i および j の値の昇順に出力される。

3 : それらの画素に挟まれた x 軸に垂直な辺や面の総数

4 : それらの画素に挟まれた y 軸に垂直な辺や面の総数

5 (si\_mfc のみ) : それらの画素に挟まれた z 軸に垂直な面の総数

注

起動パラメータ rangeList[1,2]の両方に "-" を指定して起動すると t\_mec と si\_mfc は画像上のすべての種類の辺や面の個数を計測します。ただし、その処理には画像上の画素の総数に関わらず最小限以下の容量の計算機メモリが必要です。

32 ビット CPU の計算機上で 8 ビット画素値の画像を処理

t\_mec : 255 KB 程度

si\_mfc : 383 KB 程度

32 ビット CPU の計算機上で 16 ビット画素値の画像を処理

t\_mec : 16 GB 程度 (実行不可能)

si\_mfc : 24 GB 程度 (実行不可能)

64 ビット CPU の計算機上で 8 ビット画素値の画像を処理

t\_mec : 510 KB 程度

si\_mfc : 765 KB 程度

64 ビット CPU の計算機上で 16 ビット画素値の画像を処理

t\_mec : 32 GB 程度 (そちらの Linux 機での実行は?)

si\_mfc : 48 GB 程度 (そちらの Linux 機での実行は?)

最後に、3次元クラスタ画像 mcl/ の上のクラスタの像それぞれの「表面積」の計測は以下のようにします (クラスタ画像上ではクラスタそれぞれの像はすべて画素値 0 を持つ背景の画素に囲まれている)。

単純に si\_mfc を実行してすべてのクラスタを調べる :

```
si_mfc mcl/ - 0 1- > result.txt
```

クラスタ番号 1~10 の像の表面積を調べる :

```
si_mfc mcl/ - 0 1-10 > result.txt
```

出力の各行の先頭の「0」を削除する :

```
si_mfc mcl/ - 0 1-10 | cut -f2- > result.txt
```

[x,y,z] 軸に垂直な3個の表面積の和の値を行末に付加する：

```
si_mfc mcl/ - 0 1-10 | (改行しない)
awk '{ printf "%s\t%s\t%s\t%s\t%g\n", (改行しない)
      $2,$3,$4,$5,$3+$4+$5 }' > result.txt
```

注：上記の中括弧 `{ }` の前後の引用符は single quote です。

CSV 形式ファイルに変換して出力：

```
si_mfc mcl/ - 0 1-10 | (改行しない)
awk '{ printf "%s,%s,%s,%s,%g\n", (改行しない)
      $2,$3,$4,$5,$3+$4+$5 }' > result.csv
```

とりあえず以上です。