

Date: Mon, 07 Apr 2008 19:35:21 +0900  
From: Tsukasa NAKANO  
To: Junji Torii  
Cc: Shunji KASAMA  
Subject: new-ver.-of-MCL-and-MHL

---

とりいさま、

産総研の中野司です。最近になって「slice シリーズ」のプログラム群の抜本的な改造を始めました。まだほんの少ししか手を付けていませんが、その発端となった新版のクラスタラベリング用プログラムの説明を記した E-mail を送ります。ただし、この E-mail では 3次元画像用のプログラム `si_mcl` と `si_mhl` だけを説明していますが、今回はそれらに加えて 2次元画像用のプログラム `t_mcl` と `t_mhl` も書きました。これらの起動法は 3次元用のものとほぼ同じです（これらのコマンド名だけをキー入力すれば起動法の簡単な説明が表示されます）。

なお、これら 4 個のプログラムのソースファイルと Windows 用実行ファイルは以下の書庫ファイルに入っています(説明の E-mail に添付してあった書庫ファイルは除去しました)。

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

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

どうぞお試し下さい。連絡が遅れてすみません。

P.S.

[1] 今回の改造は「32 bit の壁」を越えるためのものです。とは言え、「slice シリーズ」のユーザが使っている計算機は 32 bit 機が多いので、64 bit 機を使えば 32 bit の壁を越えることができ、かつ、32 bit 機でも従来のプログラムの不備による「31 bit の壁」を越えることができるようにしました。

[2] 今回のプログラムでは以前に紹介した Hoshen & Koppleman のアルゴリズムを少しだけ変えたものを使っています。このちょっとした改変でクラスタラベリングの処理がかなり速くなりました。

添付した E-mail

---

Date: Tue, 18 Mar 2008 17:19:06 +0900  
 From: Tsukasa NAKANO  
 To: Satoshi Okumura  
 CC: Kentaro Uesugi, Akira TSUCHIYAMA, Masayuki Uesugi, Michihiko NAKAMURA  
 Subject: new-ver.-of-MCL-and-MHL

---

おくむらさま、

なかのです。先日の E-mail に書いた 3次元画像の **multiple cluster labeling** や **multiple hole labeling** を行うためのプログラムの抜本的な改造を行いました：

旧版	新版	処理内容
sliceMCL	si_mcl	multiple cluster labeling を行う。
sliceMHL	si_mhl	multiple hole labeling を行う。

この E-mail に添付した書庫ファイル **mcl.taz** (削除しました) を適当なディレクトリにコピーした後 (Cygwin 上で) 以下を入力すれば実行ファイルをコンパイルできます：

```
tar xzf mcl.taz
cd mcl
make
```

こうして得られた実行ファイル **si\_mcl.exe** と **si\_mhl.exe** を実行パスに登録されているディレクトリにコピーすれば準備は完了です。

これらのプログラムの起動法は従来の **slice** シリーズのものと若干異なります：

```
multiple cluster labeler
  sliceMCL orgDir nameFile lower upper {newDir}
  si_mcl orgDir nameFile rangeList {newDir}
```

```
multiple hole labeler
  sliceMHL orgDir nameFile lower upper {newDir}
  si_mhl orgDir nameFile rangeList {newDir}
```

新版のプログラムでは 3次元画像の上で「固体の部分」と見なす画素値の範囲を起動パラメータ **rangeList** に与えた文字列で以下のようにして指定することができます (文字列の中に空白を挟まないこと)：

文字列	意味
100	画素値 100 の画素を「固体の部分」と見なす。
100-	画素値が 100 以上の画素を ...。
100-200	画素値が 100~200 の画素を ...。
100,200	画素値が 100 もしくは 200 の画素を ...。
-100,200-	画素値が 100 以下もしくは 200 以上の画素を ...。

なお、新しいプログラムは従来のもものと完全に互換な出力を行います。

これらのプログラムを先日の **E-mail** に書いてあった巨大画像で試してみてください（これらでもそれを取り扱えない場合は対処法があります）。

とり急ぎ、

Date: Tue, 08 Apr 2008 10:29:05 +0900  
From: Tsukasa NAKANO  
To: Junji Torii  
Cc: Shunji KASAMA  
Subject: Re: new-ver.-of-MCL-and-MHL

---

とりいさま、

なかのです。E-mail ありがとうございます。昨日の E-mail に書き忘れていました。si\_mcl と si\_mhl を大メモリの 64 bit 機でコンパイルする場合は、

```
gcc -DIOM ...
```

のようにして定数 "IOM" を define して下さい ("Makefile" の先頭の所で comment out してあるコンパイラの設定をご覧ください)。si\_mcl や si\_mhl は 3 次元画像を 5 回スキャンして処理を行います。定数 IOM (Image On Memory) を define しておくと 3 次元画像すべてをメモリに読み込んだ後に処理を行いますが、そうでない場合にはスライス画像のファイルを合計 5 回アクセスします (ディスクキャッシュが大きければ問題ないですが、そうでないと ...)。とり急ぎ、

---

Date: Tue, 08 Apr 2008 10:41:16 +0900  
From: Tsukasa NAKANO  
To: Junji Torii  
Cc: Shunji KASAMA  
Subject: Re: new-ver.-of-MCL-and-MHL

---

とりいさま、

なかのです。すみません。先程の補足です。定数 IOM を define した場合は 2 値化後の 3 次元画像をメモリに保持します。つまり、3 次元画像の画素の総数の 1/8 のバイト数のメモリが余分に必要になります。

```
1024×1024×1024 画素の画像 → 128 MB が余分に必要  
2048×2048×2048 画素の画像 → 1 GB
```

と言うことで、32 bit 機用の実行ファイルでも IOM をオンにしてコンパイルすればよいのかもしれない (Win32 機で使用可能なメモリの最大値は 2 GB?)。とり急ぎ、

Date: Fri, 17 Jun 2011 13:58:53 +0900  
From: Tsukasa NAKANO  
To: TSUCHIYAMA, Akira  
Cc: Masayuki Uesugi, Kentaro UESUGI, Ryo Noguchi, 松本徹,  
Junya Matsuno, Takashi Nagano, Yuta Imai  
Subject: peak-cluster-search

---

つちやまさま、

なかのです。先程の E-mail に書いた画素値ペアのヒストグラムの 2 次元画像に関することですが、そのピークサーチをしたい場合はプログラム `t_pcs` (PCS == peak cluster search) が利用可能です。これは隣接する周囲の画素の値よりも高い画素値を持つ画素 (同じ値の画素群が隣接している場合はそれらのクラスタ) を識別して、それらそれぞれの (重心の) 座標値などを出力するプログラムです。

`t_pcs` (および 3 次元画像用プログラム `si_pcs`) のソースコードなどは以下の書庫ファイルに入っています (もちろん、HSL にもインストール済み) :

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

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

また、`t_pcs` の起動法は以下の通りです :

```
t_pcs orgTIFF > pc2.txt  
t_pcs orgTIFF newTIFF > pc2.txt
```

ただし、

`orgTIFF` : 画素値のピークサーチを行いたい TIFF 画像ファイルの名前

`newTIFF` :

`orgTIFF` の画素値のうちピークの画素 (のクラスタ) 以外のものを画素値 0 に置換した画像を入れるファイルの名前

`pc2.txt` :

画像上のピーク (のクラスタ) の情報を書き込んだテキストファイル。

1 行目 (1 個の整数値が書き込まれている)

画像上に出現したピーク (のクラスタ) の総数

2 行目以降 (8 個の数値がタブコード区切りで並んでいる)

[1] そのピーククラスタの画素値

[2] そのピーククラスタのサイズ (画素数)

[3,4] そのピーククラスタの重心の x および y 座標値

[5,6,7,8] そのピーククラスタの **bounding box** の情報 (座標値)

とりあえず以上です。

P.S.

t\_pcs に相当する 3 次元画像用の画素値のピーククラスタサーチプログラムは si\_pcs で、その起動法は以下の通りです。

```
si_pcs orgDir nameFile > pc3.txt
si_pcs orgDir nameFile newDir > pc3.txt
```

ただし、

orgDir と nameFile および newDir : 意味はわかりますよね?

pc3.txt :

画像上のピーククラスタの情報を書き込んだテキストファイル

1 行目 (1 個の整数値)

ピーククラスタの総数

2 行目以降 (タブコード区切りの 11 個の値)

[1] そのピーククラスタの画素値

[2] そのピーククラスタのサイズ

[3,4,5] そのピーククラスタの重心の座標値

[6,7,8,9,10,11] そのピーククラスタの **bounding box** の情報