

第 13 回 X線マイクロ・ナノモグラフィー研究会

大阪産業大学梅田サテライトキャンパス(大阪駅前第3ビル)、2012/3/1

CT用解析ソフトの開発状況 (slice* から si_* への移行)

産業技術総合研究所(AIST)

地質調査総合センター(GSJ)

中野 司 (tsukasa.nakano @ aist.go.jp)

SLICE (大文字)

<http://www-bl20.spring8.or.jp/slice/>

- X線CT画像用の基本的な3次元解析ソフト
- UNIX (Linux/Cygwin/MacOSX) 環境で動作
- GUI なし (コマンドライン・インターフェースのみ)
- 単機能プログラムの集合体 (機能の重複アリ)
- 3次元画像: ディレクトリ / スライス画像: TIFF
- 画素値 ($\leq 16\text{bit}$) の「単位」を揃えてある白黒、グレースケールおよび擬似カラー画像

SLICE (総称) と slice* (プログラム群の名前)

SLICE = tiff* + slice* + stl_* + si_bev* + ...

- tiff* : TIFF 画像の処理プログラム群 (~1996)
- slice* : 3次元画像処理 (~2006)
- stl_* : CAD志向の3次元像処理 (2003~2007)
- si_bev* : 3次元像の鳥瞰図描画 (2006~2008)
- ... : ファイルや画像処理用のユーティリティ

上相さんによる SLICE のマニュアル類作成 : 2006~2008

stl_* : <http://www.gsj.jp/GDB/openfile/files/no0448/0448index.html>

slice* (および tiff*) の問題点

- 全体的に未熟なプログラムコード
- 省メモリ設計に伴う低処理速度
- UNIX コマンド ls をプログラム内部で使用
- 汎用 TIFF ライブラリ libtiff を使用
- 32bit 長の整数使用に起因する問題:
2G 以上の画素数で「数え上げ」の誤り発生

コードの部分修正ではなく大幅書き換えが必要

→ si_* の開発(2006～)

si_* の開発(その1)

- slice* のプログラムコードを全面書き換え
無駄を省くことにより高速化(当社比 1.5倍?)
- MinGW (Minimalist GNU for Windows) を利用した
Windows 用実行ファイル(現状は 32bit 版のみ)
- 見かけ(機能や使用法)は slice* と概ね互換
- 起動パラメータの処理法を改変(強化)
スライス画像のファイル名と配置の指定: nameFile
しきい値などの画素値の範囲の指定: rangeList
- 2次元画像(含 RGB 形式)用処理プログラムも用意

si_* の開発 (その2)

プログラム本体

- slice* と互換のものもののコードは概ね作成終了
- 機能ごとのプログラム群をまとめた書庫ファイル

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

[mcl.zip](#)、[pvr.zip](#)、[mask.zip](#)、[trim.zip](#)、[osp.zip](#)、
[ovoid.zip](#)、[rar.zip](#)、[cm.zip](#)、[affine.zip](#)、[2cc.zip](#)、...

マニュアルの類は不十分 (今後作成予定)

- 紹介 E-mails の PDF を上記サイトに転送済み

[mcl.pdf](#)、[pvr.pdf](#)、[mask+trim.pdf](#)、[of.pdf](#)、...

si_* と slice* の対応関係

書庫	si_*	slice*	説明
mcl	mcl, mhl	MCL, MHL	クラスタ・ラベリングなど
pvr	pvr	PVR	画素値置換・出現頻度調査
mask	mask	PVM	画像のマスク処理
trim	trim, paste, area	IT	画像領域のトリミング関連
osp	osp, osp2, osp3, ed, de, ...	OSP, OSP3 ED, DE, ...	物体像表面の「皮むき」 Erosion-Dilation 処理など
ovoid	of, op, ...	OF, O, ...	物体像の楕円体近似
rar	rar, sir, sie, pnr, pne	RAR, BIC, SIM	画像の直角回転 等方縮小・拡大
affine	affine, tli, ir	NSG, TLI, IR	Affine(1次の座標)変換
cm	cm, rgb, ...		擬似カラー処理など
2cc	2cc, acc, ...	2CFC, AC, ...	2点相関係数

si_* 特有の起動パラメータ処理(その1)

例: クラスタ・ラベリング用プログラムの起動

```
sliceMCL orgDir nameFile lower upper {newDir} > CLtxt  
si_mcl   orgDir nameFile rangeList   {newDir} > CL.txt
```

- 入出力する画像やテキストファイルの形式は同じ
- **rangeList** でしきい値などの画素値の範囲を指定

128: 画素値 128

128-255: 画素値 128~255

128-: 画素値 128 ~ (65535)

-64,128,192-: 画素値 0~64、128 もしくは 192~

si_* 特有の起動パラメータ処理(その2)

- **nameFile** でスライス画像のファイル名と配置を指定
ファイル名(文字列):

指定したディレクトリの下のスライス画像のファイル名を
スライスの配置の順に列記したテキストファイルの名前

—(負の符号)

ディレクトリの下ファイルすべてを英数字順で選択

—ファイル名: 1枚のスライス画像のファイルだけを選択

— **r**: ファイル名の英数字順の逆順にスライスを並べる

— **n**: ファイル名に含まれる数値順にスライスを並べる

— **nr** | — **rn**: ファイル名の数値の逆順に並べる

si_* の使用法

- 書庫ファイル (*.zip もしくは *.taz)

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

プログラムのコンパイル (Windows では不要)

実行ファイルのコピーもしくは実行パスの設定

- HASPET 用のパッケージ (for Linux)

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

画像再構成 → SLICE 用の画像 → 縮小画像 → 鳥瞰図

- 個々のプログラムの使い方

HASPET によるハヤブサ・サンプルの初期分析の話

最近書いた3次元像の鳥瞰図描画プログラム群の話

hvd(仮称): 3次元像の鳥瞰図描画プログラム群

stl_bev* と si_[s,m,t,x,y,z]_bev* の全面書き換え

- プログラム群の統合
 - カラーの鳥瞰図画像を自動判別 → 「C」を廃止
 - 物体像表面を常にスムージングして描画 → 「SS」を廃止
- 起動パラメータの指定を簡素化
 - 鳥瞰図描画用のデフォルト・パラメータ値を導入
 - 環境変数 BEV_*** による指定でそれらの変更が可能
- マルチ・スレッディングなどによる高速化
 - 鳥瞰図画像のファイル書き込みの並列化
 - 物体像表面のレンダリング(塗りつぶし)の並列化
- 立体視画像作成・3次元ディスプレイ出力に対応

鳥瞰図描画プログラム群の統合(その1)

STL データを用いた鳥瞰図描画 (stl_bev*)

(1)~(4) stl_bev、stl_bev_C、stl_bev_GIF、stl_bev_C_GIF

(5) stl_bev_SS + 6個以上の起動パラメータの指定が必要

(6) stl_bev_C_SS + 13 個以上

(7) stl_bev_GIF_SS + 7 個以上

(8) stl_bev_C_GIF_SS STL_files scale gamma bias ¥

bgR bgG bgB fgR fgG fgB scR scG scB GIF

プログラム (5)~(8) を以下の2種類に統合

(1) **stl_bev_0** -line|STL ... format

(2) **stl_bev_[1-4]** -line|STL ... GIF

鳥瞰図描画プログラム群の統合(その2)

3次元画像上の物体像の鳥瞰図描画 (si_?_bev*)

- 一種類の物体像を描画 (si_s_bev*)

si_s_bev と si_s_bev_nss + 8~12 個の起動パラメータ

→ **si_s_bev_[0-4]** directory nameFile rangeList format|GIF

- 複数の物体像を「色分け」して描画 (si_m_bev*)

si_m_bev と si_m_bev_nss + 12~16 個

→ **si_m_bev_[0-4]** directory nameFile colorFile format|GIF

- しきい値を変えながら識別した像を描画 (si_t_bev*)

- 2画像上の物体像を合成して描画 (si_[x,y,z]_bev*)

si_[t,x,y,z]_bev と si_[t,x,y,z]_bev_nss → **si_[t,x,y,z]_bev_[0-4]**

鳥瞰図描画プログラム群の「バージョン(仮称)」

プログラム名の最後の数字0~4

- 0では鳥瞰図を個別の **TIFF** 画像ファイルに格納。
また、**立体視画像**作成用のデータを出力可能。
 - 1~4はいずれも鳥瞰図を1個の **GIF** にまとめる。
 - 1:すべての処理を1スレッドで実行
 - 2:鳥瞰図のファイル書き込みを別スレッドで実行
 - 3:物体像表面の塗りつぶしを**複数スレッド+**で実行
 - 4:上記の2と3のマルチスレッド処理を併せて実行
- † stl_bev_[3,4] 以外のものは3スレッドに固定

環境変数 BEV_* (その1)

環境変数の設定、削除および確認法

csh|tcsh **setenv BEV_SIZE 800**

unsetenv BEV_SIZE

printenv | grep BEV_

sh|bash **BEV_SIZE=800 ; export BEV_SIZE**

export -n BEV_SIZE

env | grep BEV_

cmd (Windows のコマンドプロンプト)

set BEV_SIZE=800

set BEV_SIZE=

set

環境変数 BEV_* (その2)

鳥瞰図描画用のデフォルト・パラメータ値を変更

- 鳥瞰図画像(正方形)のサイズに関するパラメータ

説明	環境変数	設定値	デフォルト値
横縦画素数	BEV_SIZE	2以上の自然数	なし
画素の辺長	BEV_UNIT	unit Ux,Uy,Uz	1(== 1,1,1)
座標倍率 †	BEV_SCALE	scale Sx,Sy,Sz	1(== 1,1,1)

† BEV_UNIT と意味的には同じだが stl_bev_[0-4] ではこちらを使用

- 物体像表面の表示輝度 (= $1 - \text{ratio} + \text{ratio} \times \cos^{1/\text{gamma}}$)

gamma factor	BEV_GF	正の実数	1
ratio of shade	BEV_RS	0~1の実数	1

環境変数 BEV_* (その3)

- 鳥瞰図画像上の描画色の指定

背景色	BEV_BC	gray R,G,B	255(白)
枠線色	BEV_FC	{-}gray {-}R,G,B †	0(黒)
表面色	BEV_SC	{-}gray {-}R,G,B †	255
線の色	BEV_LC	{-}gray {-}R,G,B †	0

† 値に負の符号を付けて描線の取り止めなどを指定可能

- 立体視画像作成用データの出力(*_bev_0のみ)

BEV_ZBF ファイル名のフォーマット なし

- スレッド数の指定(stl_bev_[3,4]のみ)

BEV_THREADS スレッド数(自然数) 1

stl_bev_[0-4]

STL データを用いた鳥瞰図描画

起動法

stl_bev_0 —line|STL ... format

stl_bev_[1-4] —line|STL ... GIF

環境変数 BEV_*

stl_bev_0 ZBF

stl_bev_[3,4] **THREADS**

stl_bev_[0-4] SIZE、**SCALE**、GF、RS、BC、FC、SC、LC

標準入力(複数行を指定可)

lonBase latBase {lonStep latStep views}

標準出力(stl_bev_0 のみ)

pathFB lon lat h1 v1 h2 v2 {pathZB d1 d2}

stl_bev_[0-4] の実行例と 3次元画像上の物体像の STL データ作成法

stl_bev_0 の実行例 (カラーの地球儀の描画)

set BEV_SIZE=800 ← 800 × 800 画素の鳥瞰図画像にする

mkdir e_g ← 鳥瞰図の TIFF を格納するディレクトリ作成

echo 0 35 10 0 36 | stl_bev_0 e_g.stl e_g/%02d.tif

→ pathFB	lon	lat	h1	v1	h2	v2
e_g/00.tif	0	35	168	77	631	722
...
e_g/35.tif	350	35	132	56	667	743

3次元画像上の物体像の STL データ作成

si_[s,b]_stl spl - 1- s.stl ← si_s_bev* のものに相当

si_[m,c]_stl spl - spl.rgb m.stl ← si_m_bev* のものに相当

si_s_bev_[0-4]

3次元画像上の一種類の物体像を識別して描画

起動法

```
si_s_bev_0    directory nameFile rangeList format  
si_s_bev_[1-4] directory nameFile rangeList GIF
```

環境変数 BEV_*

```
si_s_bev_0    ZBF  
si_s_bev_[0-4] SIZE、UNIT、GF、RS、BC、FC、SC
```

標準入力(複数行を指定可)

```
lonBase latBase {lonStep latStep views}
```

標準出力(si_s_bev_0のみ)

```
pathFB lon lat h1 v1 h2 v2 {pathZB d1 d2}
```

実行例(視線方向を変えつつ全 spinel クラスタを表示)

```
echo 0 210 10 0 36 | si_s_bev_1 spl - 1- s.gif
```

si_m_bev_[0-4]

3次元画像上の複数の物体像を「色分け」して描画

起動法

```
si_m_bev_0    directory nameFile colorFile format  
si_m_bev_[1-4] directory nameFile colorFile GIF
```

環境変数 BEV_*

```
si_m_bev_0    ZBF  
si_m_bev_[0-4] SIZE、UNIT、GF、RS、BC、FC
```

標準入力(複数行を指定可)

```
lonBase latBase {lonStep latStep views}
```

標準出力(si_m_bev_0のみ)

```
pathFB lon lat h1 v1 h2 v2 {pathZB d1 d2}
```

実行例(視線方向を変えつつ全 spinel クラスタを色分け表示)

```
echo 0 210 10 0 36 | si_m_bev_2 spl - spl.rgb m.gif
```

colorFile

画像上の画素値ごとの表示・非表示と表示色を指定

- 各行に rangeList と gray|(R G B) を記述
- テキストファイル spl.rgb

```
1: 0
2: 1,8,15,22,29,36,43,50,57,64,71,78,¥
   85,92,99,106,113,120,127,134,141 255 255 255
3: 2,9,16,23,30,37,44,51,58,65,72,79, ¥
   86,93,100,107,114,121,128,135,142 255 255 0
4: 3,10,17,24,31,38,45,52,59,66,73,80,87,94,101,108,115,122,129,136,143 255 0 255
5: 4,11,18,25,32,39,46,53,60,67,74,81,88,95,102,109,116,123,130,137 255 0 0
6: 5,12,19,26,33,40,47,54,61,68,75,82,89,96,103,110,117,124,131,138 0 255 255
7: 6,13,20,27,34,41,48,55,62,69,76,83,90,97,104,111,118,125,132,139 0 255 0
8: 7,14,21,28,35,42,49,56,63,70,77,84,91,98,105,112,119,126,133,140 0 0 255
9: 144- 255
```

si_t_bev_[0-4]

しきい値を変えながら識別した3次元像を描画

起動法

```
si_t_bev_0    directory nameFile {colorFile} format  
si_t_bev_[1-4] directory nameFile {colorFile} GIF
```

環境変数 BEV_*

```
si_t_bev_0    ZBF  
si_t_bev_[0-4] SIZE、UNIT、GF、RS、BC、FC、SC
```

標準入力(複数行を指定可)

```
pvBase lonBase latBase {pvStep lonStep latStep views}
```

標準出力(si_t_bev_0のみ)

```
pathFB pv lon lat h1 v1 h2 v2 {pathZB d1 d2}
```

実行例(spinelのクラスタを大きなものから順に消去)

```
echo 1 0 210 1 10 0 36 | si_t_bev_3 spl - spl.rgb t.gif
```

si_[x,y,z]_bev_[0-4]

2個の画像上の物体像を [x,y,z] 方向に並べて描画

起動法

```
si_[x,y,z]_bev_0      directory1 nameFile1 colorFile1 ¥  
                      directory2 nameFile2 colorFile2 format  
si_[x,y,z]_bev_[1-4] directory1 nameFile1 colorFile1 ¥  
                      directory2 nameFile2 colorFile2 GIF
```

環境変数 BEV_*

```
si_[x,y,z]_bev_0      ZBF  
si_[x,y,z]_bev_[0-4] SIZE、UNIT、GF、RS、BC、FC、LC
```

標準入力(複数行を指定可)

```
xyzBase lonBase latBase {xyzStep lonStep latStep views}
```

標準出力(si_[x,y,z]_bev_0のみ)

```
pathFB xyz lon lat h1 v1 h2 v2 {pathZB d1 d2}
```


si_[x,y,z]_bev_[0-4] の実行例

3次元画像の縦断面の鳥瞰図

x 軸に垂直な断面図の連続表示

```
echo 332 240 210 -20 0 0 16 | ¥  
si_x_bev_4 spl - spl.rgb - - nul x.gif
```

y 軸に垂直な断面図の連続表示

```
echo 348 240 210 -20 0 0 17 | ¥  
si_y_bev_4 spl - spl.rgb - - nul y.gif
```

2個の画像上の物体像の合成(物体像の「変身」)

```
echo 0 240 210 20 0 0 25 | ¥  
si_z_bev_4 spl - spl.rgb uz1 - uz1.rgb z.gif
```

立体視画像の作成法

手順

1. プログラム*_bev_0 に指定する起動パラメータ format と同様な形式で立体視画像作成用データ (Z-buffer; 実体は TIFF 画像) のファイル名のフォーマットを環境変数 **BEV_ZBF** に設定する。
2. 必要ならそのファイル用のディレクトリを作成する。
3. 標準出力を適当なテキストファイルにリダイレクトするように指定して *_bev_0 を実行する。
4. そのファイルのデータを標準入力に流し込むように指定してプログラム **s3d_*** を実行する。

プログラム s3d_*

下記の5個のプログラムの起動法はすべて同じ

Philips 社の S3D 方式と同等な手法で立体視画像作成

- *_bev_0 の普通の鳥瞰図と同等な画像を作成

s3d_svm angle format

- 立体視用の2枚の鳥瞰図画像を並べて表示

s3d_align angle format

- 赤・青(正確には、赤・シアン)メガネ用立体視画像を作成

s3d_RCG angle format

- 裸眼3次元ディスプレイへの出力用画像を作成

SHARP 社の LL151D 用 s3d_LL151D angle format

VMJ 社の 22PBVDP 用 s3d_22PBVDP angle format

ただし、起動パラメータ angle は輻輳(フクソウ)角の半分の角度

立体視画像作成の実例

地球儀の立体視画像(グレースケール / カラー)

```
set BEV_SC=-255          ← グレースケールの場合
```

```
set BEV_ZBF=zb/%02d.tif ← Z-buffer のファイル名のフォーマット
```

```
set BEV_SIZE=800
```

```
mkdir fb zb
```

```
echo 0 35 10 0 36 | stl_bev_0 e_g.stl fb/%02d.tif > log.txt
```

```
mkdir svm align RCG LL151D 22PBVDP
```

```
s3d_svm 0 svm/%02d.tif < log.txt
```

```
s3d_align 30 align/%02d.tif < log.txt
```

```
s3d_RCG 3 RCG/%02d.tif < log.txt
```

```
s3d_LL151D 2.86 LL151D/%02d.tif < log.txt
```

```
s3d_22PBVDP 2.15 22PBVDP/%02d.tif < log.txt
```

*_bev_[1-4] の処理時間(秒; その1)

Windows (Vista) + GCC / dynabook SS (Core2Duo U9300@1.20GHz)

	stl_*	si_s_*	si_m_*	si_t_*	si_x_*	si_y_*	si_z_*
旧版	64	296	470	231	207	209	513
*_bev_1	26	95	130	271	242	253	611
*_bev_2	23	94	128	264	245	254	607
*_bev_3	17	64	98	173	142	151	344
*_bev_4	16	63	97	174	141	150	344

Linux (Knoppix) + GCC / dynabook SS (Core2Duo U9300@1.20GHz)

旧版	33	55	85	204	176	175	424
*_bev_1	18	59	91	227	193	201	499
*_bev_2	16	59	95	237	195	206	505
*_bev_3	13	43	86	199	118	125	301
*_bev_4	12	43	84	196	138	125	302

*_bev_[1-4] の処理時間(秒; その2)

Windows (Vista) + GCC / dynabook SS (Core2Duo U9300@1.20GHz)

	stl_*	si_s_*	si_m_*	si_t_*	si_x_*	si_y_*	si_z_*
旧版	64	296	470	231	207	209	513
*_bev_1	26	95	130	271	242	253	611
*_bev_2	23	94	128	264	245	254	607
*_bev_3	17	64	98	173	142	151	344
*_bev_4	16	63	97	174	141	150	344

Windows (Vista) + ICC / dynabook SS (Core2Duo U9300@1.20GHz)

旧版 (GCC)	65	287	474	232	208	209	511
*_bev_1	12	39	64	239	170	185	422
*_bev_2	11	39	62	216	171	180	412
*_bev_3	9	31	55	139	108	116	271
*_bev_4	9	31	53	141	107	116	269

*_bev_[1-4] の処理時間(秒;その3)

Windows (Vista) + GCC / dynabook SS (Core2Duo U9300@1.20GHz)

	stl_*	si_s_*	si_m_*	si_t_*	si_x_*	si_y_*	si_z_*
旧版	64	296	470	231	207	209	513
*_bev_1	26	95	130	271	242	253	611
*_bev_2	23	94	128	264	245	254	607
*_bev_3	17	64	98	173	142	151	344
*_bev_4	16	63	97	174	141	150	344

Linux (RHEL5) + ICC / DELL PRECISION T7500 (Xeon X5570@2.93GHz × 2)

旧版	9	12	22	98	59	59	145
*_bev_1	2	10	19	109	61	64	157
*_bev_2	2	10	19	98	62	64	158
*_bev_3	1	6	17	42	25	27	69
*_bev_4	1	7	17	44	25	26	67

TO DO

- hvd(仮称)用の書庫ファイルの準備
ソースコードなどのファイル+Windows用の実行ファイル+
デモ用のデータやバッチファイルの類+マニュアルの類
<http://www-bl20.spring8.or.jp/~sp8ct/tmp/hvd.taz>
<http://www-bl20.spring8.or.jp/~sp8ct/tmp/hvd.zip>
- si_* のマニュアルの類の整備
SLICE のホームページの「分室」にアップロードする？
- 新しい si_* などのプログラム開発・公開
ECS(3次元画像を用いた電流シミュレーション)プログラム
プログラム *ch で得た物体像の凸包(convex hull)の特徴抽出
その他(ご要望に応じます)