Slice

Program Reference

SPring-8

目 次

tiffwl

書式：tiffwl TIFF

機能説明：

TIFF 形式の画像の縦横サイズを幅, 高さの順でタブ区切りの数値として表示する.

パラメータ：

TIFF

調べる tiff 画像

使用例：

次の例は test.tif の縦横サイズを表示する

tiffwl test.tif

slicePNC (**I**mage **T**rimmer)

書式：slicePNC directory nameFile

機能説明：

ディレクトリ中の３次元画像の画素数を（Nx Ny Nz 総画素数）のタブ区切りの形式で表示する．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

使用例：

次の例は BYTE ディレクトリのスライス画像データのピクセル数を表示する

slicePNC BYTE -

tiffrac

書式：tiffrac TIFF x1 y1 x2 y2 newTIFF

機能説明：

TIFF 形式の画像の縦横サイズを表示する.

パラメータ：

TIFF

もとの tiff 画像

x1 y1

切り取る範囲の原点側の座標

x2 y2

切り取る範囲の原点から遠い側の座標

newTIFF

新しい tiff 画像

使用例：

次の例は test.tif の (0,0) から (100,100) の範囲を切り出し、new.tif に保存する

tiffrac test.tif 0 0 100 100 new.tif

sliceIT (**I**mage **T**rimmer)

書式：sliceIT directory nameFile x1 y1 z1 x2 y2 z2 new directory

機能説明：

名前ファイルで指定された 3 次元画像データの物体像のディレクトリ directory に格納されている３次元画像から対角点の座標値が (x1, y1, z1) と (x2, y2, z2) で示される直方体領域の画素を切り出し、新しいディレクトリ new directory に格納する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. “-” が与えられた場合、directory の中のすべての画像が選択される.

x1 y1 z1

切り出す直方体領域の原点に一番近い点

x2 y2 z2

切り出す直方体領域の原点から一番遠い点

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリの画像データの、（10, 10, 10)、（100, 100, 100）で示される直方体領域を切り出し、it ディレクトリに保存する。

sliceIT BYTE - 10 10 10 100 100 100 it

関連項目：

sliceOSP3

tiffrar

書式：tiffrar TIFF X Y newTIFF

機能説明：

TIFF 形式の画像の軸を入れ替え、反転する.

パラメータ：

TIFF

もとの tiff 画像

X Y

軸の指定

newTIFF

新しい tiff 画像

使用例：

次の例は test.tif の x 軸と y 軸を入れ替え、new.tif に保存する

tiffrar test.tif +y +x new.tif

次の例は test.tif の左右の方向を反転し、new.tif に保存する

tiffrar test.tif -x +y new.tif

軸の指定：

軸は X Y に大文字、あるいは小文字の x,y で指定する。向きは+,-を x,y の前に指定する。X Y に +x と+y（+X と+Y）を指定すると元の画像と同じ画像になる

sliceRAR(**I**mage **R**otator)

書式：sliceRAR directory nameFile hAxis vAxis new directory

機能説明：

通常の 3D 画像（z 軸に垂直な連続スライス画像）を x,y 軸に垂直な連続スライス画像に変換する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

hAxis

変換後のスライス画像の垂直軸が変換前のスライス画像の何軸に対応するかの指定.

vAxis

変換後のスライス画像の水平軸が変換前のスライス画像の何軸に対応するかの指定.

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリの中のデータを水平軸が変換前の y 軸、垂直軸が変換前の z 軸、高さ方向が変換前の x 軸になるように変換する

sliceRAR BYTE - +y +z rar

軸の指定：

軸は hAxis vAxis に大文字、あるいは小文字の x,y,z で指定する。向きは+,-を x,y,z の前に指定する。hAxis vAxis に +x と+y（+X と+Y）を指定すると元の画像と同じ画像になる

注：

スライス画像のが画素サイズが違う場合は、スライス画像のアスペクト比が変わってしまうので注意

関連項目：

sliceNSG

tiffsr

書式：tiffsr TIFF xRatio yRatio angle [spp bg orbgR bgG bgB] newTIFF

機能説明：

TIFF 形式の画像を回転、拡大縮小する.

パラメータ：

TIFF

もとの tiff 画像

xRatio yRatio

各辺の拡大率

angle

回転角度

spp

グレースケール画像の場合は 1, カラー画像の場合は 3 を指定する

bg
bgR bgG bgB

回転した後、元の画像にはない領域の画素に与える画素値。spp に 1 を指定した場合は bg を、spp に 3 を指定した場合は bgR bgG bgB の三つの値（RGB 値）を指定する。画素値の値は画像の bit によって 0-255,0-65535 までとなる

newTIFF

新しい tiff 画像

使用例：

次の例はグレースケール画像 test.tif を各辺倍に拡大し、new.tif に保存する

tiffsr test.tif 2 2 0 1 0 new.tif

次の例はカラー画像 test.tif を時計回りに 45 度回転させ、new.tif に保存する

tiffsr test.tif 1 1 45 3 0 0 0 new.tif

sliceIR(**I**mage **R**otator)

書式：sliceIR directory nameFile [scaleX scaleY scaleZ] lon lat rot OoI new directory

機能説明：

与えられた経度（lon)、緯度（lat）で示される、ある方向を新しい z 軸とするような一連の断面画像を作成する．尚、本プログラムは三次元画像をいったんメモリに読み込むので動作環境及びデータのサイズによってはうまく動作しないことがある．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

scaleX scaleY scaleZ

新しい画像の x，y，z 方向の拡大率．指定を省略すると 1 倍になる

lon lat rot

新しい画像の経度 λ，緯度 ϕ，及び z 軸の周りの回転角 θ.

OoI

断面画像に 3 次元画像の外部の領域 (**O**ut **o**f **I**mage) が含まれる場合にそこに与える画素値

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリの中のデータの経度、緯度が（60, -20）移動した一連の画像群を作成する。元データの外部の画素が新しい画像の中に存在する場合、その画素に画素値 0 を与える。作成されたデータは ir ディレクトリに保存される。

sliceIR BYTE - 60 -20 0 ir

次の例は BYTE ディレクトリの中の 3 次元データを y 軸を中心に+5 度傾けた一連の画像を作る。また、元データの外部の画素が新しい画像の中に存在する場合、その画素に画素値 0 を与える。

sliceIR BYTE - 95 0 90 0 ir

３次元画像の座標と回転角について：

sliceIR では初期の z 軸の方向は-x に一致しており、x 軸が-y 方向に一致している（従って **sliceIR** を通して元通りの画像を得るためには lon=90, lat=90 または lat=90, rot=90 を指定する。下図及び表参照)。元の画像の x，y，z が新しい画像の x，y，z に一致する回転の場合の 3 次元画像の回転角を示す。指定した回転角と新旧の座標軸の対応関係は以下のようになっている。なお、新しい画像の z 軸の方向は x，y 軸の方向から自動的に決まるため、ここでは書かれていない。

関連項目：

sliceNSG

sliceNSG (**N**ew **S**lice **G**enerator)

書式：sliceNSG directory nameFile scaleX scaleY scaleZ OoI

機能説明：

３次元画像上の指定した点を通る任意の方向から見た画像を作成する。標準入力からの指定によって一度の起動で複数の断面を切り出すことができる.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前.“-” が与えられた場合、directory の中のすべての画像が選択される.

scaleX scaleY scaleZ

新しい画像の x，y，z 方向の拡大率.指定を省略すると 1 倍になる

OoI

断面画像に３次元画像の外部の領域 (**O**ut **o**f **I**mage) が含まれる場合にそこに与える画素値.

使用例：

次の例は echo を使い、CT データの中の (500,500,300) の位置を通り、経度、緯度が (20, 60) の方向からみた回転無しの画像、cs.tif を作成する。なお、このとき BYTE という名前のディレクトリ内のデータを用い、スケーリングファクターはすべて１、また三次元画像の外部の領域には画素値０を指定している。

echo 500 500 300 20 60 0 cs.tif | **sliceNSG** - 1 1 1 0

次の例は経度、緯度が (20,60) 、回転無しの方向からみた 500_500_300.tif と 100_200_400.tif の二つの画像を直接標準入力に指定することによって作成する。

sliceNSG BYTE - 1 1 1 0（リターン）
500 500 300 20 60 0 500_500_300.tif（リターン）
100 200 400 20 60 0 100_200_400.tif（リターン）
（コントロールを押しながら d）

次の例はこの内容を適当なテキストファイルに書いて保存し、ターミナル上で ”csh ファイル名 ” と入力すること等で実行できる。内容は上と同じ。

sliceNSG BYTE - 1 1 1 0 << fin
500 500 300 20 60 0 500_500_300.tif
100 200 400 20 60 0 100_200_400.tif
fin

断面の指定法：

sliceNSG は 1 行にかかれたそれぞれの断面を指定するデータを標準入力から読み込む。断面を指定する各行のデータは空白もしくはタブコードで区切られた以下の 7 個である。

x0, y0, z0
: 断面が通る画像上の点の座標値。座標値は 3 次元画像上の画素の位置を指定する場合と同じ単位の値である（起動時に指定されたスケーリングファクターが考慮されない値。浮動小数点数も可）。

λ, ϕ, θ
: 断面を見込む方向（断面の法線方向）を表す角度。ただし λ と ϕ は XYZ 空間での視線方向の経度と緯度を、また、θ は断面内での回転を意味する角度である。角度はすべて起動時に与えられたスケーリングファクターを考慮した座標系での値を度単位で指定する。

filename
: 新しい断面画像を入れるフィルの名前。ディレクトリ名や拡張子を含んだ完全な名前を指定する。

関連項目：

sliceNSX,**sliceNSY**,**sliceIR**,**sliceRAR**

sliceNSX

書式：sliceNSX directory nameFile scaleX scaleY scaleZ new directory

機能説明：

３次元スライス画像から、X 軸の方向から見た slice 画像を作成する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前.“-” が与えられた場合、directory の中のすべての画像が選択される.

scaleX scaleY scaleZ

新しい画像の x，y，z 方向の拡大率.指定を省略すると 1 倍になる

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリの中のスライス画像から x 軸方向から見たスライス画像を作成し、nsx ディレクトリに保存する。

sliceNSX BYTE - 1 1 1 nsx

関連項目：

sliceNSG,**sliceNSY**,**sliceIR**,**sliceRAR**

sliceNSY

書式：sliceNSY directory nameFile scaleX scaleY scaleZ new directory

機能説明：

３次元スライス画像から、Y 軸の方向から見た slice 画像を作成する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前.“-” が与えられた場合、directory の中のすべての画像が選択される.

scaleX scaleY scaleZ

新しい画像の x，y，z 方向の拡大率.指定を省略すると 1 倍になる

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリの中のスライス画像から y 軸方向から見たスライス画像を作成し、nsy ディレクトリに保存する。

sliceNSY BYTE - 1 1 1 nsy

関連項目：

sliceNSG,**sliceNSX**,**sliceIR**,**sliceRAR**

sliceIRC (**I**nclined **R**ectangular-parallelepiped **C**hecker)

書式：sliceIRC directory nameFile x0 y0 z0 dx dy dz [Mx My Mz] lower upper

機能説明：

sliceIRC はパラメータ lower と upper の値で決めた「物体像」の画素すべてがちょうどおさまる直方体領域の対角点の、回転後の画像上での座標値 (h1, v1, d1) および (h2, v2, d2) をこの順でタブコードで区切った 1 行にまとめて標準出力に書き出す.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

x0 y0 z0

回転前の画像上での回転中心の座標値（浮動小数点数も可）.

dx dy dz

回転前の座標系の値で表現した回転後の画像の z 軸（d 軸）の方向を指すベクトル（単位ベクトルでなくてもかまわない）の成分値（浮動小数点数でもよい）.

Mx My Mz

画像の拡大率（正の浮動小数点数）. 回転前の画像の座標系に対する比率を指定する. 1 未満の値を指定すると画像の縮小が行われるが、前述の「re-sampling 処理」に使う変数の制約のため、$1/(Mx \times My \times Mz) < 256$ でなければならない. これらの指定を省略すると $Mx = My = Mz = 1$ と見なされる.

lower upper

物体と見なす画素値の上限と下限値

使用例：

次の例は BYTE ディレクトリのスライス画像を x 軸の周りに 45 度回転させた後の座標を調べる

sliceIRC BYTE - 0 0 0 0 1 1 0 0 255

次の例は BYTE ディレクトリのスライス画像内部の画素値が 180-190 の画素を持つ物体を y 軸の周りに 45 度回転させた後の座標を調べる

sliceIRC BYTE - 0 0 0 1 0 1 0 180 190

関連項目：

sliceIRS,**sliceNSG**,**sliceIR**,**sliceRAR**

sliceIRS (**I**nclined **R**ectangular-parallelepiped **S**cooper)

書式：sliceIRS directory nameFile x0 y0 z0 dx dy dz [Mx My Mz] OoI [h1 v1 d1 h2 v2 d2] new directory

機能説明：

回転後の画像上の指定された直方体領域のデータだけをメモリに格納して３次元画像を回転させる．この領域を計算機のメモリ量に合わせて適切に指定してやれば、画像全体を分割して処理することができる．rmsd シリーズのプログラム add_3d を使えば、分割した複数の３次元画像をなめらかにつなぐことができる（sliceIRS は分割した画像がなめらかにつながるようなデータを出力する）．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

x0 y0 z0

回転前の画像上での回転中心の座標値（浮動小数点数も可）．

dx dy dz

回転前の座標系の値で表現した回転後の画像の z 軸（d 軸）の方向を指すベクトル（単位ベクトルでなくてもかまわない）の成分値（浮動小数点数でもよい）．

Mx My Mz

画像の拡大率（正の浮動小数点数）．回転前の画像の座標系に対する比率を指定する．１未満の値を指定すると画像の縮小が行われるが、前述の「re-sampling 処理」に使う変数の制約のため、$1/(\mathrm{Mx} \times \mathrm{My} \times \mathrm{Mz}) < 256$ でなければならない．これらの指定を省略すると $\mathrm{Mx} = \mathrm{My} = \mathrm{Mz} = 1$ と見なされる．

OoI

断面画像に３次元画像の外部の領域 (**O**ut **o**f **I**mage) が含まれる場合にそこに与える画素値

h1 v1 d1 h2 v2 d2

計算機のメモリ上に格納して「re-sampling 処理」を行う回転後の画像上の直方体領域の対角点の座標値（浮動小数点数も可)。これらの指定を省略すると回転後の画像全体が処理対象になる．

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例はスライス画像を x 軸の周りに 45 度回転させる (回転後の座標は sliceIRC で調べてある)

sliceIRS BYTE - 0 0 0 0 1 1 0 -1000 -707.107 0 0 463.862 1170.97 irs

次の例は上記のスライス画像を 4 分割し、x 軸の周りに 45 度回転させる

sliceIRS BYTE - 0 0 0 0 1 1 0 -1000 -707.107 0 -500 463.862 585.485 irs1
sliceIRS BYTE - 0 0 0 0 1 1 0 -1000 -707.107 585.485 -500 463.862 1170.97 irs2
sliceIRS BYTE - 0 0 0 0 1 1 0 -500 -707.107 0 0 463.862 585.485 irs3
sliceIRS BYTE - 0 0 0 0 1 1 0 -500 -707.107 585.485 0 463.862 1170.97 irs4

出力：

sliceIRS は起動パラメータ x0、y0、z0 で指定した回転前の画像上の回転中心に相当する回転後の画像上の点の座標値 h0、v0、d0 を（画像回転の処理を始める前に）標準出力に出力する

領域の指定：

sliceIRS で画像を分割して処理するには回転後の画像が占める領域の情報を処理（画像の回転）の前に知っておく必要があるので、それを調べるためのプログラム sliceIRC をもちいる。
従来のプログラムとは異なり、sliceIRS と sliceIRC は指定された回転中心の座標値 (x0,y0,z0) と回転後の画像の z 軸（d 軸）の方向を指すベクトルの値 (dx,dy,dz) を用いて画像回転を行う。
sliceIRS や sliceIRC で画像回転を指定するパラメータの (x0,y0,z0) は任意の座標値でよい（回転前の画像が占める領域になくてもよい）。また、(dx,dy,dz) は成分値がすべて 0 でなければどのようなベクトルでもかわない。これらの値を用いて以下の式に従った座標変換を行う。

$dx^2 + dy^2 == 0$ かつ dz > 0 の場合：
(H,V,D) = (X,Y,Z)
$dx^2 + dy^2 == 0$ かつ dz < 0 の場合：
(H,V,D) = (X,-Y,-Z)
$dx^2 > 0$ かつ $dy^2 > 0$ の場合：
H = (- X * dy + Y * dx) / Dxy
V = - (X * dx + Y * dy) * dz / Dxy + Z * Dxy / Dxyz
D = (X * dx + Y * dy + Z * dz) / Dxyz

ただし、

(x,y,z)：回転前の画像上の座標値
(h,v,d)：回転後の画像上の座標値
(x0,y0,z0)：回転前の座標系の値で示した回転中心の座標値
(h0,v0,d0)：回転後の座標系の値で示した回転中心の座標値
(X,Y,Z) = (x,y,z) - (x0,y0,z0)
(H,V,D) = (h,v,d) - (h0,v0,d0)
$Dxy^2 = dx^2 + dy^2$ $Dxyz^2 = dx^2 + dy^2 + dz^2$

指定されたベクトルが z 軸と平行な $dx^2 + dy^2 == 0$ の場合、上記のように回転前後の対応している座標軸のそれぞれが平行になる。また、そうでない場合の座標変換の概要は以下の通り。

h 軸は xy 平面内にある。
v 軸方向を指す単位ベクトルの z 成分は正の値になっている。
d 軸は指定したベクトルの方向を向いている。

領域の計算：

分割した画像がなめらかにつながるように、sliceIRS は以下の 2 つの手続きからなる「re-sampling 処理」によって回転後の画像上の画素値のそれぞれを回転前の画素値から計算する：

[1] 画素値の over-sampling
そもそもの画像が占めている直方体の内部領域において回転後の画素のそれぞれに少なくとも 1 個の回転前の画素が対応するように、回転前の画像上の各画素を細分した「sub-pixels」のそれぞれをスキャンして座標値の変換（回転）の処理を行う。

[2] 画素値の平均の計算
回転後の画像上のそれぞれの画素に対しては、それに対応する回転前の画像上の「sub-pixels」の値の平均を画素値とする。

上の手続き [1] は通常の画像拡大の、また、[2] は画像縮小の処理で使われている。sliceIRS では回転と同時に画像を拡大・縮小することが可能で、そうではない「等倍」の画像回転の際にもこれら 2 つの手続きを実行する。

sliceIRS に指定した（回転前の画素に対する）[x,y,z] 方向の倍率を M[x,y,z] とすると、手続き [1] で回転前の各画素を「sub-pixels」に細分する [x,y,z] 方向の区画の数 I[x,y,z] は以下の値になる。

I[x,y,z] = floor(M[x,y,z]) + 1

ただし、floor() は小数点以下を切り捨てた整数化を行う関数。

この式から「等倍」の画像回転では I[x,y,z] = 2 となり、手続き [1] でスキャンすべき「sub-pixels」の個数は元の画像の画素数の $2^3 = 8$ 倍になる。つまり、手続き [1] には多大な処理時間を要す。

手続き [2] で行う画素値の平均の計算のために、sliceIRS は回転後のそれぞれの画素に対して以下の作業用のメモリ（変数）を使う：

変数 NUM
その画素に対応する回転前の「sub-pixels」の個数を保持する。

変数 SUM
「sub-pixels」の画素値（回転前の画素値）の和を保持する。

sliceIRS では変数 NUM として 8 bits 長の符号なし整数を使っているので、回転と同時に画像を縮小する場合の倍率 M[x,y,z] は以下の式を満たしている必要がある（sliceIRS にこの関係式を満たさない倍率を指定するとエラーになる）。

1 / (Mx × My × Mz) ≦ 255（== NUM に保持可能な最大値）

処理したい画像の画素値が 8 bits 以下で、かつ、回転と同時に画像を縮小しないなら 16 bits 長の符号なし整数を SUM に割り当てれば十分である。そうでない場合には以下の関係式などを考慮してコンパイル時に SUM 用のメモリサイズを指定する必要がある。

画素値の最大値 / (Mx × My × Mz) ≦ SUM に保持可能な最大値
or
画素値の最大値 × 8 ≦ SUM に保持可能な最大値

なお、sliceIRS には手続き [2] の計算中に発生した変数 SUM の値のオーバーフローに対する備えがない。

注 1：

従来の slice シリーズのプログラムでも回転前に画像を分割しておき、回転の後にこれらを統合することは一応可能。しかし、回転によって分割面が「斜め」になった場合はこれらの画像を add_[2,3]d で単純にモザイクできない。また、分割面付近の画素に対して画像の「外部」の値を拾ってしまうこともあるので、これらがなめらかにつながるとは限らない。

関連項目：

sliceIRC,**sliceNSG**,**sliceIR**,**sliceRAR**

sliceBIC (**B**rowse **I**mage **C**reator)

書式：sliceBIC directory nameFile Mx My Mz new directory

機能説明：

このプログラムはディレクトリ directory の nameFile で指定されるファイル名の画像を読み込み、その上の Mx * My * Mz 画素のブロックの平均画素値を格納した新しい画像を作成し、newDir に保存する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

Mx My Mz

縮小する画素数.

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例はディレクトリ BYTE の中の 3 次元画像データの $2 \times 2 \times 2$ 画素を 1 画素に縮小した画像データを作成し、bic ディレクトリに保存する。

sliceBIC BYTE - 2 2 2 bic

関連項目：

sliceTIL

tiffspp

書式：tiffspp TIFF

機能説明：

TIFF 形式の画像のカラーを調べる．グレースケールの場合は 1, カラーの場合は 3 を表示する

パラメータ：

TIFF

調べる tiff 画像

使用例：

次の例は test.tif のカラーを表示する

tiffspp test.tif

tiffbps

書式：tiffbps TIFF

機能説明：

8bit の場合は 8,16bit の場合は 16 など、TIFF 形式の画像の BPS を調べる.

パラメータ：

TIFF

調べる tiff 画像

使用例：

次の例は test.tif の BPS を表示する

tiffbps test.tif

tiffhc

書式：tiffhc TIFF [x1 y1 x2 y2]

機能説明：

TIFF 形式の画像のヒストグラムデータを作成する.

パラメータ：

TIFF

もとの tiff 画像

x1 y1

ヒストグラムを作成する範囲の原点側の座標

x2 y2

ヒストグラムを作成する範囲の原点から遠い側の座標

使用例：

次の例は test.tif のヒストグラムデータを作成し、hist.txt に保存する

tiffhc test.tif > hist.txt

出力：

tiffhc は以下のようになっている

1 行目：spp, bits, 幅、高さ
2 行目：画像に出現する画素値の数

を出力した後、3 行目以降にグレースケールの場合は

画素値　数

カラーの場合は

R G B 数

を出現数の多い順に出力する

slice.PVC (**P**ixel **V**alue **C**ounter)

書式：slice.PVC directory [new directory]

機能説明：

それぞれのスライス画像の画素値の出現頻度分布を調べ、そのデータ（ASCII テキストの整数値）をスライス画像ごとに別々のファイルに格納する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

new directory

画素値の出現頻度分布のデータファイルを入れるディレクトリ名。スライスごとに作成されるファイル名はそれぞれ拡張子が ”.tif” から ”.tbl” となる.

使用例：

次の例は echo を使って slicePVR の標準入力に空の指定を行い、hist.txt という名前のファイルに BYTE ディレクトリの中のファイルの画素値の出現頻度を保存する。

echo ” | **slice.PVC** BYTE - | cut -f1,3 > hist.txt

画素値の出現頻度分布のデータファイルの形式：

各行はタブで区切られた 2 つの整数値からなり、最初の数値は画素値で 2 番目がその画素値を持つ画素数。これらの行は画素値の小さいものから順に並んでいる

関連項目：

slicePVR

slicePVR (**P**ixel **V**alue **R**eplacer)

書式：slicePVR directory nameFile [new directory]

機能説明：

名前ファイルで指定されたスライス画像すべてに対して以下の処理を順に行う．[1] 画素値の出現頻度分布を調べ、それを ASCII テキストで標準出力に書き出す．[2] 標準入力から読み込んだ指定に従って画素値の置き換えを行う．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は echo を使って slicePVR の標準入力に空の指定を行い、hist.txt という名前のファイルに BYTE ディレクトリの中のファイルの画素値の出現頻度を保存する。

echo ” | **slicePVR** BYTE - | cut -f1,3 > hist.txt

次の例は BYTE ディレクトリの中のファイルの画素値が 1 から 65535 の画素の値を 0 に置き換え、その画像を新しいディレクトリ、pvr に保存する。また、結果の出力を表示しない。

slicePVR BYTE - pvr > /dev/null（リターン）
1 65535 0（リターン）
（コントロールを押しながら d）

次の例は slicePVR の標準入力に echo を使って、画素値 0-255 の画素の値を 0-127 に、画素値 255-65535 の画素の値を 128-255 に置き換えるよう指定し、新しいディレクトリ pvr に保存する。その際の結果を pvr.txt に保存する。

(echo 0 255 0 127; echo 255 65535 128 255) | **slicePVR** BYTE - pvr > pvr.txt

画素値の置き換えの指定法：

画素値の置き換えを行うときは各行に以下のような 2，3 もしくは 4 個の整数値を空白もしくはタブコードで区切って、標準入力から指定する。

org new

画素値 org が new に置き換えられる。ただし org と new はともに 0-65535 であること（以下同じ）。

org1 org2 new

画素値が org1 から org2（org1 ≦ org2）の範囲の画素を new に置き換える。例えば 0 32768 0 と指定すると画素値が 0-32768 の画素は値が 0 に置き換えられる。

org1 org2 new1 new2

画素値 org1 から org2 を new1 から new2 に線形補間して置き換える。例えば 0 255 255 0 と指定すると、新旧の画素値 new と org の関係は org=0∼255 について new=255-org となる。

上の指定は混合して複数回行っても良いし、画素値の範囲が重複しても良い（最後の指定が優先される）。尚、置き換えの指定がなかった画素値は元のまま保たれる。

画素値の置き換えの指定法：

各表にタブで区切られた 3 つの整数値が出力され、最初の数値は元の画素値、2 番目は置き換え後の画素値、そして最後の数値はその画素値を持つ画素数。これらの行は元の画素値の小さい物から順に並んでいる。
注）slicePVR が出力する画素値の出現頻度分布のデータの形式を slicePVC で作成されるファイルと同じにしたければ

echo ” | **slicePVR** directory nameFile | **cut** -f1,3

のようにして cut コマンドで各行の 2 番目の数値 (置き換え後の画素値) を取り除けばよい。

関連項目：

slicePVC

sliceSCL(**S**ingle **C**luster **L**abeler)

書式：sliceSCL directory nameFile

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なしてそのつながりを調べ、それによって物体像と判定された画素のうちで指定された画像中の点に一番距離が近い cluster だけを探索する。そして、探索された cluster ごとに別々の３次元画像（BPS が１ビットの mask 画像）として格納する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

使用例：

次の例は BYTE ディレクトリの中の 3 次元データの、画素値が 65 から 80 までの画素を物体と見なし、cluster labeling を行い (200,200,200) の点に最も近いクラスタの mask 画像を mcl ディレクトリに保存する。また、その結果を scllog.txt に出力する

echo 65 80 200 200 200 scl | **sliceSCL** BYTE - > scllog.txt

cluster 探索の指定法：

標準入力の各行を読み出して、それぞれの行で空白もしくはタブコードで区切られた以下の３種類のパラメータの値を用いて cluster の探索を行なう（パラメータの個数としては５もしくは６個で、最後以外はすべて整数値）

lower upper

物体像と見なす画素値の下限（lower $\geqq$ 0）と上限（upper<65535）

x y z

cluster に近い画像上の点の座標値

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

cluster 探索の結果の出力形式：

cluster が見つかった場合には cluster ごとに以下の４種類の情報（12 個の整数値）がタブコードで区切られた１行として標準出力に書き出される：[1] lower と upper、[2] 与えられた点（x、y および z）に最も近い cluster 上の画素の座標値 x’、y’ および z’、[3] cluster のサイズ（cluster に属する画素の個数）、[4] cluster の広がりの情報。これらのうち後２者は sliceMCL が標準出力に出力する cluster の情報と同様のものである。

関連項目：

sliceMCL

sliceMCL(**M**ultiple **C**luster **L**abeler)

書式：sliceMCL directory nameFile lower upper new directory

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なしてそのつながりを調べ、cluster labeling を行う．得られたすべての cluster それぞれの情報を ASCII テキストとして標準出力に書き出した後、新しいディレクトリ名の指定があれば、物体像でない背景の画素には 0、それ以外には画素が属する cluster のサイズ（cluster に属する画素数）の順に 1 から付けられた番号（cluster 番号）を画素値とする三次元画像（cluster 画像）のスライス画像をそこに作成する．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像と見なす画素値の下限（lower $\geqq$ 0）

upper

物体像と見なす画素値の上限（upper<65535）

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの中の 3 次元データの、画素値が 65 から 80 までの画素を物体と見なし、cluster labeling を行いその結果の 3 次元データを mcl ディレクトリに保存する。また、その結果を mcllog.txt に出力する

sliceMCL BYTE - 65 80 mcl > mcllog.txt

次の例は BYTE ディレクトリの中のファイルの画素値が 1 の画素の部分を物体と見なし、cluster を調べ、そのうち大きい物から 200 個までのサイズ、広がりを表示する。

sliceMCL BYTE - 1 1 | head -202

Cluster の情報データの出力形式：

1 行目には cluster の総数（背景と見なした画素をまとめた物はつながっていない可能性もあるが一つの cluster としている）が出力される。2 行目には背景の画素の cluster についての、3 行目以降には cluster 番号の順でそれぞれの cluster についてサイズ (画素数) と cluster の広がり x1, y1, z1, x2, y2, z2（cluster に属する画素の座標値は x1 ≦ x ≦ x2, y1 ≦ y ≦ y2, かつ z1 ≦ z ≦ z2）を示す合計 7 つの整数値が 1 行にタブコード区切りで出力される。

注 1：

このプログラムでは物体内部に穴 (背景と見なされる画素) がある場合、それを cluster の一部ではないとしている。このため、画像を単純に 2 値化して得た物体像の画素数は背景以外の cluster サイズの合計と一致するが、穴の中に島状に cluster が存在するという複雑な状況も生じうる。物体内部の穴がどの程度の画素数になるかは sliceMCL で得られた背景以外の cluster に属する画素の個数と sliceOSP で得られた物体像の画素の個数を比較すれば判る。

注 2：

このプログラムで作成される cluster 画像は BPS が 16bit なので、cluster 番号 65535 以降の cluster に属する画素の画素値はすべて 65535 になる（cluster 画像上では 65535 個までの cluster しか区別できない)。それに対して cluster labeling はすべての cluster について行われる（すべての cluster を調べられなければエラーが起こる）ので、標準出力に書き出される情報はすべての cluster についてのものである。

関連項目：

sliceSCL

sliceMHL(**M**ultiple **H**ole **L**abeler)

書式：sliceMHL directory nameFile lower upper new directory

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なし、この物体像に開いた穴（外部とつながっていない物体像以外の部分）に対して **sliceMCL** と同じ手法でクラスタラベリングを行う．得られたすべての hole それぞれの情報を ASCII テキストとして標準出力に書き出した後、新しいディレクトリ名の指定があれば、物体像でない背景の画素には 0、それ以外には画素が属する hole のサイズ（hole に属する画素数）の順に 2 から付けられた番号（hole 番号）を画素値とする三次元画像（hole 画像）のスライス画像をそこに作成する．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像と見なす画素値の下限（lower $\geqq$ 0）

upper

物体像と見なす画素値の上限（upper<65535）

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの中の 3 次元データの、画素値が 65 から 80 までの画素を物体と見なし、hole labeling を行いその結果の 3 次元データを mhl ディレクトリに保存する。また、その結果を mhllog.txt に出力する

sliceMHL BYTE - 65 80 mhl > mhllog.txt

次の例は BYTE ディレクトリの中のファイルの画素値が 1 の画素の部分を物体と見なし、hole を調べ、そのうち大きい物から 200 個までのサイズ、広がりを表示する。

sliceMHL BYTE - 1 1 | head -203

Hole の情報データの出力形式：

1 行目にはクラスタの総数（外部+物体像＋ hole の個数）が出力される。2 行目には背景となる部分 (画素値 0 となる部分)、3 行目には物体と見なした部分（画素値 1 となる部分）の、４行目以降には cluster 番号の順でそれぞれの hole についてサイズ (画素数) と hole の広がり x1, y1, z1, x2, y2, z2（hole に属する画素の座標値は x1 ≦ x ≦ x2, y1 ≦ y ≦ y2, かつ z1 ≦ z ≦ z2）を示す合計 7 つの整数値が 1 行にタブコード区切りで出力される。

注 1：

このプログラムで作成される hole 画像は BPS が 16bit なので、hole 番号 65535 以降の hole に属する画素の画素値はすべて 65535 になる（cluster 画像上では 65535 個までの hole しか区別できない)。それに対して hole labeling はすべての hole について行われる（すべての hole を調べられなければエラーが起こる）ので、標準出力に書き出される情報はすべての hole についてのものである。

関連項目：

sliceMCL

calc.fcc

書式：calc.fcc labelTable

機能説明：

sliceMCL で得られた cluster の情報から物体像をなす画素の個数の３次元画像の全画素数に対する割合（fraction ofoccupied pixels)、cluster の総数（cluster number）および cluster の連結度（connectivity of clusters）を計算し、標準出力の１行にタブコードで区切ってこの順番で書き出す.

パラメータ：

labelTable

sliceMCL で得られた cluster の情報を書き込んだファイル名。省略した場合は標準入力からデータの読み込みが行なわれるので、**sliceMCL** の出力データを直接流し込むパイプとして用いることができる.

使用例：

次の例は mcllog.txt 内のクラスタの情報を表示する

sliceMCL BYTE - 65 80 mcl > mcllog.txt
calc.fcc mcllog.txt

次の例は sliceMCL からのクラスタ情報の出力を直接 calc.fcc で受け取る

sliceMCL BYTE - 65 80 | **calc.fcc**

sliceOSP3 (**O**bject **S**kin **P**arer **3**D)

書式：sliceOSP3 directory nameFile threshold thickness [new directory]

機能説明：

名前ファイルで指定された 3 次元画像データの物体像の周囲にある背景の部分（空気層）を与えられた閾値（threshold）より小さい値を持つ画素を画像端から塗りつぶすことによって識別する．さらに与えられた厚さ（thickness）の画素の層を物体像の縁から皮をむく要領で強制的に取り除く．なお、このとき物体の内側の部分についてはこの皮むきは行われない．3 次元データの物体像の広がりを調べ、それがちょうど収まるように画像の端から塗りつぶした新しい 3 次元画像を作成する．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像の周囲にある背景の部分を判別する際に用いる画素値の閾値（threshold $\geqq 1$）

upper

物体像の縁から取り除く層の厚さ（thickness $\geqq 1$; thickness =0 の時は皮むきをしない）

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの画像データの、画素値が 50 以下の画素を画像の縁から塗りつぶし、3 次元的に 5 画素分の皮むきをする。作成された画像データを trim ディレクトリに保存する。作成された画像の情報が標準出力に出力される。

sliceOSP3 BYTE - 50 5 trim

次の例は BYTE ディレクトリ内の画像の 30 以上の画素を持つ画素を物体像と見なし、その物体の占める領域の情報を trim.txt ファイルに保存する。画像は保存されない。

sliceOSP3 BYTE - 30 0 > trim.txt

物体像の広がりのデータの出力形式：(num x1 y1 z1 x2 y2 z2)：

new directory の中の新しく作成された画像における物体像と見なした画素の個数 (num) とその広がりを 1 行でタブ区切りのデータとして標準出力に出力する。広がりのデータは 3 次元データの立方体の頂点のうち、原点にもっとも近い点 (x1 y1 z1) ともっとも原点から遠い点 (x2 y2 z2) の 3 次元座標で表される

関連項目：

sliceOSP

sliceDE (**D**ilation **E**rosin)

書式：sliceDE directory nameFile lower upper layers new directory

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なして、まず物体像表面の画素を一層ずつ画素を付け加えて太らせること（Dilation）を指定した回数（layers）繰り返し、残った像の表面に Dilation と同じ回数だけ一層ずつ剥ぎ取ること（Erosion）繰り返す。この結果、Dilation によって埋められた物体の穴の部分などはそのまま埋められ、それ以外の物は Erosion によって元の像とほぼ同じに復元される。

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像と見なす画素値の下限（lower $\geqq$ 0）

upper

物体像と見なす画素値の上限（upper<65535）

layers

Dilation/Erosion を行う層の数

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの画像データの画素値が 100-120 の部分を物体と見なし、3 層の Dilation/Erosion を行う

sliceDE BYTE - 100 120 3 de

注：

slice シリーズでは erosion や dilation は物体の voxel が角になっている部分では行われない．　このため、layers の数を大きく取ると、元の形状から形がずれてくることがある

関連項目：

sliceED, **ed_asm**, **de_asm**

sliceED (**E**rosin **D**ilation)

書式：sliceED directory nameFile lower upper layers new directory

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なして、まず物体像表面の画素を一層ずつ剥ぎ取ること（Erosion）を指定した回数（layers）繰り返し、残った像の表面に一層ずつ画素を付け加えて太らせること（Dilation）を Erosion と同じ回数だけ繰り返す。この結果、Erosion によって消去した像は除去され、それ以外の物は Dilation によって元の像とほぼ同じに復元される。

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像と見なす画素値の下限（lower $\geqq$ 0）

upper

物体像と見なす画素値の上限（upper<65535）

layers

Erosion/Dilation を行う層の数

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの画像データの画素値が 100-120 の部分を物体と見なし、3 層の Erosion/Dilation を行う

sliceED BYTE - 100 120 3 ed

注：

slice シリーズでは erosion/dilation は物体の voxel が角になっている部分では行われない．　このため、layers の数を大きく取ると、元の形状から形がずれてくることがある

関連項目：

sliceDE, **ed_asm**, **de_asm**

de_asm

書式：de_asm directory nameFile lower upper erode dilate new directory

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なして、まず物体像表面の画素に一層ずつ画素を付け加えて太らせること（Dilation）を指定した回数 dilate だけ繰り返し、その後像の表面を一層ずつ剥ぎ取ること（Erosion）を指定した回数（erode）繰り返す。この結果、Dilation によって埋められた物体の穴の部分などはそのまま埋められ、それ以外の物は Erosion によって元の像とほぼ同じに復元される。

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像と見なす画素値の下限（lower $\geqq$ 0）

upper

物体像と見なす画素値の上限（upper<65535）

dilate

Dilation を行う層の数

erode

Erosion を行う層の数

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの画像データの画素値が 100-120 の部分を物体と見なし、3 層の Dilation、4 層の Erosion を行う

de_asm BYTE - 100 120 3 4 deasm

出力：

正常に動作するとタブコードで区切られた複数の整数値からなる 5 行 24 個の整数値の表示が行われる

[1] オリジナルの画像の x,y,z 方向の画素数 Nx, Ny, Nz
[2] オリジナルの画像中で物体像がちょうど収まる直方体領域の x,y,z 方向の広がり x1,y1,z1,x2,y2,z2
[3] dilation の後に物体がちょうど収まる直方体領域の広がり x3,y3,z3,x4,y4,z4
[4] erosion の後で物体像がちょうど収まる直方体領域の広がり x5,y5,z5,x6,y6,z6
[5] 新しい画像の画素数 x6-x5+1,y6-y5+1, z6-z5+1

注 1：

slice シリーズの他のプログラムとは違い、de_asm が物体像がちょうど収まる直方体領域だけをスライス画像ファイルとして格納する。新しい画像ファイルは独自のスライス番号（0∼）に基づいたファイル名でディレクトリに格納される

注 2：

slice シリーズでは erosion/dilation は物体の voxel が角になっている部分では行われない. このため、erode dilate の数を大きく取ると、元の形状から形がずれてくることがある

関連項目：

sliceDE, **sliceED**, **de_asm**

ed_asm

書式：ed_asm directory nameFile lower upper erode dilate new directory

機能説明：

与えられた下限（lower）と上限（upper）の範囲の画素値を持つ画素を物体像と見なして、まず物体像表面の画素を一層ずつ剥ぎ取ること（Erosion）を指定した回数（erode）繰り返し、その後残った像の表面に一層ずつ画素を付け加えて太らせること（Dilation）を指定した回数 dilate だけ繰り返す。この結果、Erosion によって消去した像は除去され、それ以外の物は Dilation によって元の像とほぼ同じに復元される。

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

lower

物体像と見なす画素値の下限（lower $\geqq$ 0）

upper

物体像と見なす画素値の上限（upper<65535）

erode

Erosion を行う層の数

dilate

Dilation を行う層の数

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの画像データの画素値が 100-120 の部分を物体と見なし、3 層の Erosion、4 層の Dilation を行う

ed_asm BYTE - 100 120 3 4 edasm

出力：

正常に動作するとタブコードで区切られた複数の整数値からなる 5 行 24 個の整数値の表示が行われる

[1] オリジナルの画像の x,y,z 方向の画素数 Nx, Ny, Nz
[2] オリジナルの画像中で物体像がちょうど収まる直方体領域の x,y,z 方向の広がり x1,y1,z1,x2,y2,z2
[3] erosion の後で物体像がちょうど収まる直方体領域の広がり x3,y3,z3,x4,y4,z4
[4] dilation の後に物体がちょうど収まる直方体領域の広がり x5,y5,z5,x6,y6,z6
[5] 新しい画像の画素数 x6-x5+1,y6-y5+1, z6-z5+1

注 1：

slice シリーズの他のプログラムとは違い、ed_asm が物体像がちょうど収まる直方体領域だけをスライス画像ファイルとして格納する。新しい画像ファイルは独自のスライス番号（0∼）に基づいたファイル名でディレクトリに格納される

注 2：

slice シリーズでは erosion/dilation は物体の voxel が角になっている部分では行われない. このため、erode dilate の数を大きく取ると、元の形状から形がずれてくることがある

関連項目：

sliceDE, **sliceED**, **de_asm**

tiffmask

書式：tiffmask maskTIFF SPP BG FG newTIFF

機能説明：

TIFF 形式のマスク画像のそれぞれの画素値に従って、それが０なら背景（background）用に指定した値もしくは指定した画像の画素値を、また非０なら前景（foreground）用に指定した値もしくは指定した画像の画素値を新しい画像の画素値とする．

パラメータ：

maskTIFF

マスク画像の入った TIFF 形式のファイル名。マスクとして RGB カラー形式の画像を与えてもよい

SPP

新しく作成する画像の画素あたりの画素値データの個数（Samples Per Pixel）を１もしくは３と指定する。RGB カラー画像を作成するのなら SPP は３、それ以外なら SPP は１である．

BG

マスク画像で画素値が０の背景の画素に対して用いる TIFF 形式の画像ファイルの名前もしくは画素値。SPP が１の場合には、背景にする画像のファイル名と画素値の指定を区別するために画素値には負の符号 ”-”を付け加える（画素値 0 を指定する場合でも ”-0” のように記述する）。SPP が３で RGB カラー画像を作成する場合には、指定する画素値は 0~255 の範囲の３つの整数で、これらはそれぞれ背景に塗る色の R、G、B 成分の値であると解釈される（ただし SPP が１の場合と同様で、その R 成分の値を ”-” をつけて指定する）。また SPP が３で背景として RGB カラー画像以外の形式の画像のファイル名を指定した場合は、背景の画素値としてその画像の画素値ではなく、それが表示される際の色成分のデータが新しい画像に用いられる．

FG

マスク画像で画素値が非０の前景の画素に対して用いる TIFF 形式の画像ファイルの名前もしくは画素値。ファイル名もしくは画素値の指定の仕方は BG の場合と同様である．

newTIFF

新しく作成する TIFF 形式の画像ファイル名．

使用例：

次の例は sliceBEVM.DS によって作成されたボリュームレンダリング画像に、depth 画像から作成されたフレーム画像を用いて 3 次元のボックスの枠線を白く書き込む。

tiffmask mouse.w.tif 1 mouse.l.tif -255 mouse.tif

注 1：

tiffmask の起動時のパラメータ指定はやや複雑なので、以下にそれらすべての組み合わせを書き下しておく。

白黒もしくはグレースケールの画像を作成する場合

背景、前景とも画像ファイルを指定する場合

tiffmask maskTIFF 1 bgTIFF fgTIFF newTIFF

背景は画素値、前景は画像ファイルを指定する場合

tiffmask maskTIFF 1 -bg fgTIFF newTIFF

背景は画像ファイル、前景は画素値を指定する場合

tiffmask maskTIFF 1 bgTIFF -fg newTIFF

背景、前景とも画素値を指定する場合

tiffmask maskTIFF 1 -bg -fg newTIFF

RGB カラー画像を作成する場合

背景、前景とも画像ファイルを指定する場合

tiffmask maskTIFF 3 bgTIFF fgTIFF newTIFF

背景は画素値、前景は画像ファイルを指定する場合

tiffmask maskTIFF 3 -bgR bgG bgB fgTIFF newTIFF

背景は画像ファイル、前景は画素値を指定する場合

tiffmask maskTIFF 3 bgTIFF fgR fgG fgB newTIFF

背景、前景とも画素値を指定する場合

tiffmask maskTIFF 3 -bgR bgG bgB fgR fgG fgB newTIFF

ここで、bgTIFF と fgTIFF は背景と前景の画像ファイル名、bg と fg は背景と前景の画素値、そして、bgR、bgG、bgB と fgR、fgG、fgB はそれぞれ背景と前景に塗る色の R、G、B 成分の値（0 ～255）である。

注 2：

背景もしくは前景に指定する色成分のうちで R 成分以外の値が ”-” ではじまる場合、先頭の ”-” は取り除かれる。

注 3：

背景もしくは前景に指定するファイルの画像の横縦画素数はマスク画像のものと同じでなければならない。

関連項目：

slicePVM

slicePVM (**O**bject **P**ixel **V**alue **M**asker)

書式：slicePVM maskDir nameFile background foregroundDir [new directory]

機能説明：

指定されたマスク画像のそれぞれの画素の画素値に従って、それが０なら指定した背景（background）用の画素値を、また、非０なら前景（foregroundDir）に指定した画像の画素値をそれぞれの画素におさめた新しい画像を作成する。すなわち、前景用の画像の画素値をマスク画像を使ってマスクする。なお、前景用の画像のサイズ（画素数）はマスク画像のものと同じでなければならない。

パラメータ：

maskDir

マスク画像のファイルがあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

background

背景の画素に入れる画素値

foregroundDir

前景用の画像のファイルがあるディレクトリ名

new directory

新しいスライス画像ファイルを入れるディレクトリ名．

使用例：

次の例は BYTE ディレクトリの画像データに従って画素値が 0 なら黒に塗りつぶし、0 以外なら fore ディレクトリの画像の画素を入れる。

slicePVM BYTE - 0 fore pvm

関連項目：

tiffmask

sliceOF (**O**void **F**itter)

書式：sliceOF directory nameFile scaleX scaleY scaleZ

機能説明：

クラスタ画像に含まれるそれぞれのクラスタに３軸不等楕円体をあてはめ、その結果を標準出力に出力する。このプログラムは２次元画像中で物体（粒子）の形状を楕円近似してその長・単軸方向を解析するプログラムの３次元版として通常の３次元クラスタ画像に用いること以外に、slice2CFC で得た２点相関関数の値を使って統計的な意味での３次元物体の軸方向の解析に用いることもできる。

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．“-” が与えられた場合、directory の中のすべての画像が選択される．

scaleX scaleY scaleZ

画素の X、Y、Z 方向の辺の実際の長さを示すスケールファクター。**sliceOF** で得られる楕円体のパラメータはここで与えられた数値を単位とする．

使用例：

次の例はディレクトリ BYTE の中の３次元画像クラスタ画像に含まれるそれぞれのクラスタに X、Y、Z 方向の辺の長さのスケールを 2 とした３軸不等楕円体をあてはめる。

sliceOF BYTE - 2 2 2

標準出力に書き出されるクラスタを近似する楕円体の情報：

背景の画素を示す番号０のクラスタも含めて、クラスタ番号順で画像に含まれるすべてのクラスタそれぞれについて以下の 15 個の数値がタブコードで区切られて１行に出力される。

クラスタ番号
クラスタに属する画素の個数
クラスタの重心の X、Y、Z 座標値（画像上の座標表現）
クラスタの重心の X、Y、Z 座標値（実際の長さの座標系）
楕円体の３軸の方向を示す回転角 λ、ϕ、θ（単位は度）
楕円体の３軸の長さ a、b、c（実際の長さの単位）
軸長から計算された楕円体の体積（a × b × c × π × 4/3）

注１：

sliceOF の出力データの 1 行目は画素値 0 の背景を楕円体近似したデータが入っているので、他のプログラムでこのデータを使用する際は 1 行目を削除しなければならないことに注意

注２：

楕円体の 3 軸は軸長が 0 ≦ a ≦ b ≦ c となるように決められており、a、b、c 軸方向を示す単位ベクトル ea、eb、ec は以下のように XYZ 座標系成分で表現できる。

$$ea = \begin{pmatrix} -\sin\lambda\cos\theta - \cos\lambda\sin\theta\sin\phi \\ \cos\lambda\cos\theta - \sin\lambda\sin\theta\sin\phi \\ \sin\theta\cos\phi \end{pmatrix}$$

$$eb = \begin{pmatrix} \sin\lambda\sin\theta - \cos\lambda\cos\theta\sin\phi \\ -\cos\lambda\sin\theta - \sin\lambda\cos\theta\sin\phi \\ \cos\theta\cos\phi \end{pmatrix}$$

$$ec = \begin{pmatrix} \cos\lambda\cos\phi \\ \sin\lambda\cos\phi \\ \sin\phi \end{pmatrix}$$

これより、たとえばクラスタが最も延びている c 軸方向は ϕ が正の値なら Z 軸の正の方向であることなどがわかる。

関連項目：

slice2CFC

slice_NSO

書式：slice_NSO directory nameFile scaleX scaleY scaleZ OoI x0 y0 z0 λ ϕ θ newName

機能説明：

sliceOF で求めたクラスタ画像中の物体像を近似する３軸不等楕円体のデータを用いて、クラスタ画像と同じ広がりを持つ３次元画像中の物体像の重心に相当する点を通り、物体像の近似楕円体の a、b、c 軸方向から見た３枚の断面画像を作成する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

scaleX scaleY scaleZ

画素の X、Y、Z 方向の辺の長さを示すスケーリングファクター。なお、後に示す方法で sliceOF の出力データを用いる場合には、ここで与える３つの値の比を sliceOF に与えたスケーリングファクターの比と一致させなければならない（たとえば、sliceOF に 0.3、0.6、0.9 という値を与えたなら、ここでの３つの値の比は１：２：３でないといけない）

OoI

断面画像に３次元画像の外部の領域 (**O**ut **o**f **I**mage) が含まれる場合にそこに与える画素値.

x0 y0 z0

次元画像上でのクラスタ（楕円体）の重心（中心）の座標値。スケーリングを行なっていない画像上の画素の位置を指定する際と同じ単位の値を指定する.

λ ϕ θ

楕円体の軸方向を示す角度（単位は度）.

newName

新しく作成する断面画像のファイル名を示す文字列。これに拡張子 ".oa.tif"、".ob.tif"、".oc.tif" が付け加わったファイルにそれぞれ a、b、c 軸方向から見た断面の画像がおさめられる.

使用例：

次の例は BYTE ディレクトリのスライス画像データを用いて、重心が (100,100,100) にあるクラスタの断面図 ovoid.oa.tif, ovoid.ob.tif, ovoid.oc.tif を作成する。

slice_NSO BYTE - 1 1 1 0 100 100 100 45 23 60 ovoid

次の例は BYTE ディレクトリのスライス画像データを用いて、ovoid.data の中のデータのうち 2 番目に大きな物のクラスタの断面図を作成する

slice_NSO BYTE - 1 1 1 0 100 100 100 ‘ tail +2 ovoid.data | head -2 | tail -1 | cut -f3,4,5,9-11‘ ovoid

断面画像上の座標と楕円体の軸方法の関係：

作成される画像は楕円体の 2 軸と平行な断面である。楕円体の a、b、c 軸方向の半径を A、B、C（A ≦ B ≦ C）として、画像上では長い半径の軸方向が h 方向（断面を画面に表示したときの水平方向）となるようにしてある。つまり、

a 軸方向から見た画像では、h = c
b 軸方向から見た画像では、h = c
c 軸方向から見た画像では、h = b

sliceOF が出力するデータの編集：

sliceOF は複数の近似楕円体のデータを標準出力に書くので、slice_NSO を走らせるためにはそのうちひとつを抜き出す必要がある。また、sliceOF のそれぞれの楕円体のデータには slice_NSO に不要なデータも含まれているので、それを取り除かなければならない。ここでは sliceOF に入力したクラスタ画像上でクラスタ番号が N のクラスタを近似した楕円体のデータを sliceOF の出力（ファイル ovoid.data に書き込まれているとする）から取り出して編集する方法を示す。

tail +2 ovoid.data | head -N | tail -1 | cut -f3,4,5,9-11

最初の tail コマンドは便宜上クラスタ番号 0 とした物体像の背景の部分を近似した楕円体のデータを捨てている。2 番目の head と tail コマンドでクラスタ番号 N のクラスタに相当する楕円体を抜き出し、最後の cut コマンドでデータ行の必要な項目を抜き出して標準出力に表示している

slice2CFC (2-points Correlation Function Calculator)

書式：slice2CFC directory nameFile ratioX ratioY ratioZ [new directory]

機能説明：

２値化された３次元画像中の物体像の２点相関関数の分布を計算する。このプログラムは物体像の画素をその画素値が非０かどうかだけで判断するので、原画像としてクラスタ画像を与えてもよい.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. “-” が与えられた場合、directory の中のすべての画像が選択される.

ratioX ratioY ratioZ

原画像の X、Y、Z 方向の広がりに対する相関係数を計算 する距離空間の領域の広がりの程度を表す比率。いずれも０以上１未満の値でなければならない（通常は 0.5 以下の値を指定する）.

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリ中のスライス画像データの自己相関関数を調べ、cfc ディレクトリに画像を保存する。また、出力データを cfc.txt に保存する

slice2CFC BYTE - 0.3 0.3 0.3 cfc > cfc.txt

作成される相関係数の画像：

相関係数を示す８ビットの画素値の３次元画像を作成する。計算された相関係数の値０～１は０～255 にスケーリングされて画像化される。また、作成される画像の画素数は X、Y、Z 方向それぞれについて

$Lx = (\mathrm{int})(Nx \times \mathrm{ratioX} + 0.5)$
$Ly = (\mathrm{int})(Ny \times \mathrm{ratioY} + 0.5)$
$Lz = (\mathrm{int})(Nz \times \mathrm{ratioZ} + 0.5)$

によって計算される Lx、Ly および Lz を２倍して１を加えた値となる。

標準出力に書き出される相関係数の画像の情報：

相関係数の距離空間の原点が位置する画像上の画素の座標値 Lx、Ly、および Lz と原点における２点相関関数の値 C(0) がタブコードで区切られた１行として出力される。

注１：

原画像上の画素の位置を示すベクトルを r=(x, y, z)、画素値を B(r)＝０ or １、そして画像上の２点間の隔たり（距離）を示すベクトルをδ=(dx, dy, dz) として（dx、dy、dz は負の値も可)、２点相関関数 C(δ) とは B(r) × B(r+δ) を画像上で考えられるすべての r について平均した値である。また相関係数 C’(δ) は、C(δ) を原点δ＝０における値 C(０) で正規化した値である。

注２：

相関係数の値は画素の空間サイズと同じ間隔の距離空間における格子点において計算され、その原点δ＝０は３次元画像の中央の画素の位置となる。ただし、相関係数の画像上での座標軸の設定の仕方は原画像と同じで、Z（dz）軸方向はスライス面と垂直な方向でスライス番号が増加する向きを正とする。また、X（dx）および Y（dy）軸はスライス面と平行で、X 軸はスライス画像の右向き、Y 軸は下向きを正とする。

注３：

２点相関関数および相関係数はその定義より原点δ＝０に関して点対称な分布となる。それゆえこれらの計算は画像化する距離空間の領域（—dx—≦ Lx かつ —dy—≦ Ly かつ —dz—≦ Lz の領域）のおよそ半分の点についてのみ行なっている。しかしながら、出力される３次元画像はこれらの重複を含んだ領域内のすべての点のデータである。

注４：

相関係数の３次元画像のスライスのデータが格納されるファイル名は名前ファイルで指定されたものとは独立に決められる。すなわち、画像上の Z 座標値に相当する０からはじまる数値（適当な桁数になるように左に０が付け加えられる）に拡張子 ”.tif”が付け加わった名前のスライス画像のファイルが指定されたディレクトリの下に作成される。

注５：

２点相関関数の計算には膨大な時間がかかる。計算時間は原画像のサイズと相関係数の画像のサイズの両方に依存するので、できるかぎり切り詰めた原画像を与えて、必要最小限の広がりの距離空間の領域の相関係数を計算したほうがよい。

関連項目：

slicePVC

slice.CMA(Color Map Adder)

書式：slice.CMA directory coloMap new directory

機能説明：

解析に用いたスライス画像（もとの画像や cluster 画像）に画像表示の際の色の情報（color palette もしくは color map）を付け加えた新しいスライス画像ファイルを作成する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

coloMap

表示用の色情報データの入ったファイルの名前.

new directory

新しいスライス画像ファイルを入れるディレクトリ名.

使用例：

次の例は BYTE ディレクトリの中のデータに color.tbl の色情報を追加する

slice.CMA BYTE color.tbl rar

画像表示用の色情報のデータファイルの形式：

１行目には画素値の総数（画像の BPS が 8 なら 256、BPS = 16 なら 65536）を、２行目以降の各行にはそれぞれの画素値を表示する際の表示色の R（赤）、G（緑）および B（青）成分の強度を 0 ～65535 の数値（16 ビットの整数値）で書き込む。

注：

画像の画素値を histogram equlization したグレースケールで表示するための色情報ファイルは calc.hecm を使えば容易に作成できる。

関連項目：

calc.hecm

calc.hecm

書式：calc.hecm [histogramTable] lower upper level

機能説明：

複数のスライス画像もしくはそれぞれのスライス画像の画素値の出現頻度分布のデータのうち指定された画素値の範囲のものを用いて、画素値を histogram equlization して指定された諧調のグレースケールで表示するための色情報のデータを作成する.

パラメータ：

labelTable

画素値の出現頻度分布のデータファイル名。指定を省略すると標準入力からデータを読み込もうとする.

lower

画素値の出現頻度データとして用いる画素値の下限.

upper

画素値の出現頻度データとして用いる画素値の上限.

level

表示の際のグレースケールの諧調数.

使用例：

次の例はヒストグラムデータ data.hst を元にグレースケールのデータファイルを作る

calc.hecm data.hst 34568 35079 256 > he.cm

次の例は **slicePVR** を用いて出力されたヒストグラムデータを元にグレースケールのデータファイルを作り、**slice.CMA** でその情報を付加したスライス画像を作成する

echo ” | **slicePVR** BYTE - pvr | cut -f1,3 | **calc.hecm** 0 255 256 > he.cm
slice.CMA pvr he.cm cma

画素値の出現頻度分布のデータファイルフォーマット：

各行はタブで区切られた 2 つの整数値からなり、最初の数値は画素値で 2 番目がその画素値を持つ画素数。これらの行は画素値の小さいものから順に並べる

put_label

書式：put_label [TIFF] [name R G B size hv label]

機能説明：

put_label は短い文字列（ラベル）を Ghostscript（様々な OS で利用可能なフリーの PostScript インタープリタ）で画像化して 2 次元画像の縁辺部に埋め込むプログラムである。複数行に渡るラベルを指定した字体・サイズ・色で TIFF 形式の画像に書き込むことができる。また、一回の起動で一個の画像に複数のラベルを書き込んだり、複数の画像に個別のラベルを書き込むこともできる。このプログラムは UNIX 系 OS の端末環境（Cygwin を含む）と Windows の DOS 窓で実行可能である.

パラメータ：

TIFF

ラベルを埋め込む TIFF 形式の画像ファイルの名前

name

ラベルを画像化する際に用いる PostScript のフォントの名前

R G B

ラベルの色の R、G、B 成分値。0〜255 の範囲の整数値を指定する（例えば、すべて 0 なら黒、255 なら白色を意味する）

size

マスク画像の入った TIFF 形式のファイル名。マスクとして RGB カラー形式の画像を与えてもよい

hv

画像に埋め込むラベルの位置を指定する二文字

label

画像上に埋め込むラベル（4096 文字以下の文字列）.

使用例：

次の例は画像の左下隅に単純なラベルを黒で書き込む

put_label fig_1.tif Helvetica 50 0 0 0 RD Fig.1

次の例は画像の左下隅に黒で 2 行に渡るラベルを書き込む：

UNIX 系 OS の場合

put_label fig_1.tif Helvetica 50 0 0 0 RD Nakano et al.,\\nFig.1

Windows の DOS 窓

put_label fig_1.tif Helvetica 50 0 0 0 RD Nakano et al.,\nFig.1

次の例は１個の画像に複数のラベルを埋め込む

put_label fig_1.tif (リターン)
Helvetica 50 255 255 255 LU A(リターン)
Helvetica 50 0 0 0 RD Fig.1(リターン)
（コントロールを押しながら d, Dos 窓ではコントロールを押しながら z+リターン）

次の例は BYTE ディレクトリ中の画像にファイル名をラベルとして埋め込む

UNIX 系 OS の場合

ls BYTE | awk '{ print "bev/" $1,$1 }' | **put_label** Helvetica 50 0 0 0 RD

Windows の DOS 窓

dir /b BYTE | awk '{ print "bev/" $1,$1 }' | **put_label** Helvetica 50 0 0 0 RD

起動パラメータの指定：

TIFF

bit 数が 16 以下の画素値を持つ白黒、グレースケール、カラーマップもしくはフルカラー形式の画像を処理できるが、いずれの場合も R、G、B 成分の値が 8 bit 長のフルカラー画像に変換した後にラベルの埋め込み処理を行う。
ラベルを埋め込んだ画像はもとのものと同じファイルに上書きされる（それゆえ、ラベルを埋め込む前に画像のバックアップをとっておいた方がよい)。その画像がグレースケールで表現できる場合にはその形式でファイルに書き込まれる。

name

ラベルを画像化する際に用いる PostScript のフォントの名前。Ghostscript に実装されている ASCII 文字のフォントを指定できる（漢字フォントは指定できない)。Ghostscript に実装されているフォントは計算機ごとに様々だが、以下の 13 種類の Postscript の標準フォントならどの計算機でも使用可能なハズである：

Times-Roman、Times-Italic、Times-Bold、Times-BoldItalic、Helvetica、Helvetica-Oblique、Helvetica-Bold、Helvetica-BoldOblique、Courier、Courier-Oblique、Courier-Bold、Courier-BoldOblique、Symbol

なお、フォントとして Symbol を指定した場合は、ラベルに含まれる以下の上段に示した記号やローマ字が下段に示した記号やギリシャ文字に置き換わる（ただし、ここに表示することができない置換文字は▲で示した)：

記号
”$ ’@¥ ^` ∀∃∋▲∴⊥ ¯
大文字
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Α Β Χ Δ Ε Φ Γ Η Ι ϑ Κ Λ Μ Ν Ο Π Θ Ρ Σ Τ Υ ς Ω Ξ Ψ Ζ
小文字
a b c d e f g h i j k l m n o p q r s t u v w x y z
α β χ δ ε φ γ η ι φ κ λ μ ν ο π θ ρ σ τ υ ϖ ω ξ ψ ζ

hv

以下の６個のいずれか（小文字でもよい）を指定する：

LU 画像の左上隅に左揃えでラベルを埋め込む。
CU 画像の中央の上部にラベルをセンタリングする。
RU 画像の右上隅に右揃えでラベルを埋め込む。
LD 画像の左下隅に左揃えでラベルを埋め込む。
CD 画像の中央の下部にラベルをセンタリングする。
RD 画像の右下隅に右揃えでラベルを埋め込む。

label

空白やタブコードを含めた ASCII 文字からなる文字列を指定できる。ただし、連続した空白やタブコードは 1 個の空白に置換される。また、後で説明するように、コマンドラインからラベルを指定する場合には特殊文字をエスケープする必要がある。
ラベルの中に 2 文字のシーケンス "\n" があるとそこで改行を行う。ただし、パラメータ hv として "?D" を指定して画像の下部にラベルを置く場合には改行は上方向に行われる（つまり、この場合にはラベルの複数行は逆順で表示される）。

注 1：

put_label を起動すると、コマンドラインから指定した 1 個の画像に 1 個のラベルを書き込むことができる。ただし、ラベルに以下の特殊文字が含まれている場合は注意が必要である：

UNIX 系 OS の場合 !"#$%&'()*;<>?[\]^ {|}~
Windows の DOS 窓 &<>^ |

コマンドラインから入力したこれらの特殊文字は OS のコマンドインタプリタ（shell）によって特別な意味に解釈されてしまうので、それをさせないように以下のいずれかを行わなければならない：

[1] 特殊文字を含む文字列を以下の引用符で囲ってやる。
UNIX 系 OS の場合 一重もしくは二重引用符
Windows の DOS 窓 二重引用符

[2] それぞれの特殊文字の前に以下のエスケープ文字を置く。
UNIX 系 OS の場合 "\"
Windows の DOS 窓 "^ "

関連項目：

addlabel

tiff2gif

書式：tiff2gif TIFF [bps] GIF

機能説明：

TIFF 形式の画像を GIF 形式に変換する.

パラメータ：

TIFF

もとの tiff 画像

bps

出力画像の BPS。省略すると 1bit になる

GIF

出力する GIF 画像

使用例：

次の例は test.tif を 16bit GIF 形式に変換する

tiff2gif test.tif 16 test.gif

slice.T2G (Tiff 2 Gif)

書式：slice.T2G directory [colorMap] new directory

機能説明：

解析に用いた TIFF のスライス画像（もとの画像や cluster 画像）を GIF 形式の画像ファイルに変換する。変換の際に TIFF 形式のスライス画像に表示用の色情報を付け加えることができる。なお、GIF 形式の画像ファイルでは画像の BPS は８以下でなければならないので、BPS が 16 ビットのスライス画像にこのプログラムを適用すると変換の際にスライス画像ごとに異なった仕方で画素値の総数が８ビットに削減される。それが好ましくないならば、slicePVR を使ってスライス画像の画素値を８ビット以下に系統的に置き換えておく必要がある.

パラメータ：

directory

元画像ファイルのあるディレクトリ名.

colorMap

表示用の色情報データの入ったファイルの名前。指定を省略するとスライス画像に既存の色情報のデータ（普通は画素値０が黒で画素値の最大値が白となるようなグレースケール）が用いられる.

new directory

GIF 形式で書き込まれたスライス画像ファイルを入れるディレクトリ名。この下のスライス画像ファイルの名前は拡張子が ”.tif” から ”.gif” となる.

使用例：

次の例はディレクトリ BYTE の中のファイルを gif ファイルに変換し、gifs ディレクトリに保存する。

slice.T2G BYTE gifs

gif_movie

書式：gif_movie gif directory nameFile DT RC ANIMATION_GIF

機能説明：

このプログラムはディレクトリ gif directory の中の GIF 画像をつなぎ合わせて、アニメーション用 GIF 画像（アニメーション GIF）のファイルを作成する.

パラメータ：

gif directory

元の GIF 画像ファイルのあるディレクトリ名

DT

アニメーションのフレーム間の遅延時間（delay time)。単位はμ sec.（100 万分の１秒）で、普通は０で OK.

RC

アニメーションの繰り返し回数（repeat count)。65536 を指定するとアニメーションを無限に繰り返す.

ANIMATION_GIF

アニメーション GIF のファイル名

使用例：

次の例はディレクトリ gifs の中の gif ファイルを用い、遅延無しで無限ループのアニメーション gif 画像を作成する。

gif_movie gifs - 0 65536 anime.gif

sliceBEVM.DS (**B**ird **E**ye's **V**iew **M**apper **D**epth **S**can)

書式：sliceBEVM.DS directory nameFile scaleX scaleY scaleZ

機能説明：

３次元画像中のクラスタの鳥瞰図（Bird Eye's View Map）画像を作成する際に必要になるすべて同じ横縦画素数の４種類の画像（depth、label、shade および edge 画像）を作成する。なお、このプログラムは標準入力から読み込んだ指定に従って様々な視線方向と光源方向の鳥瞰図用の画像を１度の起動で作成することができる.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

scaleX scaleY scaleZ

画素の X、Y、Z 方向の辺の長さ（長さの単位長）を示すスケーリングファクター.

使用例：

次の例は BYTE ディレクトリの中のデータを用い、sliceBEVM.DS の標準入力に echo を使って、視点の経度、緯度が (120, 20)、光源の位置が（110 10）の鳥観図を作成し、bevm という単語を含んだ名前の一連の画像ファイルを作成するよう入力する。

echo 120 20 110 10 bevm | sliceBEVM.DS BYTE - 1 1 1

標準入力からの鳥瞰図の指定：

視線方向もしくは光源方向を変えた鳥瞰図ごとに以下の５個のデータを空白もしくはタブコードで区切って１行にまとめて指定する（数値はフリーフォーマットの浮動少数点数）

λ_v ϕ_v

鳥瞰図の視線方向の経度と緯度（単位は度）

λ_l ϕ_l

鳥瞰図の光源方向の経度と緯度（単位は度）。

newName

作成する４種類の画像ファイルの名前の一部となる文字列。この文字列の後に ".d.tif"（depth 画像）、".l.tif"（label）、".s.tif"（shade） および".e.tif"（edge）という拡張子が付け加わった名前のファイルにそれぞれの画像が書き込まれる。

鳥瞰図の投影法：

もとの３次元画像上の座標 (x, y, z) と鳥瞰図画像上の座標 (h, v) の関係は sliceNSG のものとほぼ同じである（ただし λ を λ_v、ϕ を ϕ_v と読み換え、sliceNSG で用いていた角度 θ を０と見なす)。具体的には以下のとおり。

```
 hx =   sin λ v
 hy = - cos λ v
 hz =   0
 vx =   cos λ v * sin φ v
 vy =   sin λ v * sin φ v
vz = -          cos φ v
 xx1 〜 zz の表記は sliceNSG と同じ
 h = xx * hx + yy * hy + zz * hz - h0 + 1
 v = xx * vx + yy * vy + zz * vz - v0 + 1
```

最後の (h, v) の表記からもわかるように、同じ視線方向を示す経度と緯度の値を指定した場合に、sliceBEVM.DS で作成される鳥瞰図画像は sliceNSG が作成した断面画像の上下左右それぞれに１画素幅の余白がついた広がりを持つことになる。また、作成される鳥瞰図画像の横縦画素数についても、上の余白の存在を除けば sliceNSG のものと同様である。

作成される画像の説明：

depth 画像（最大で画素値が 16 ビットの画像）

鳥瞰図画像とは物体表面のそれぞれの点を投影した視線方向に直交するスクリーンであると考えられる。depth 画像の画素にはそこに投影された物体表面の点のうちで最も手前にある点（すなわち視線方向から見える点）のスクリーンからの奥行きを示す値が入っている。ただし、この値は視線方向から見て最も奥にある物体表面の点に対して１、手前にある点ほど大きな値になるようにしてある（その意味で奥行きでなく手前行？)。depth 画像は鳥瞰図画像に３次元の線分を描く際の陰線処理で使用される（後述するプログラム bevmLD で使用)。それ以外にも、depth 画像の等高線図を描けば、線画の鳥瞰図として使える。

label 画像（最大で画素値が 16 ビットの画像）

鳥瞰図に投影された物体表面の点が属しているクラスタのクラスタ番号がそれぞれの画素に入っている。

shade 画像（最大で画素値が８ビットの画像）

物体表面の陰影を計算する際に必要になる視線方向から見える点での物体表面の法線ベクトルと光源の方向を示す単位ベクトルの内積の値が画素におさめられている。ただし、内積が負ならば０、そうでない場合には０〜255 に正規化された値が画素値となる。また、物体がなくて向こう側が見通せる点の画素値は０にしてある。sliceBEVM.DS ではコンピュータグラフィックスで通常行なわれている物体表面のスムージング（曲面のあてはめ）を行なわないため、物体表面の法線ベクトルは直方体の画素の面が持つ６種類だけである。さらに、どのような視線方向からも画素の面のうち３つしか見えないことに対応して鳥瞰図の陰影計算に用いられるベクトルは３種類だけとなる。したがって、向こう側が透

けて見える点に相当する画素の値０と合わせて、shade 画像には４種類以下の画素値だけが含まれる。

edge 画像（画素値が１ビットの画像）

物体像を構成する画素の辺を描いた線画の鳥瞰図。

label 画像（最大で画素値が 16 ビットの画像）

鳥瞰図に投影された物体表面の点が属しているクラスタのクラスタ番号がそれぞれの画素に入っている。

注：sliceBEVM の作成する４種類の画像はそのままでは鳥瞰図とは言えないものである。これらを後述するプログラムで合成して何種類かの鳥瞰図を完成することができる。

関連項目：

bevmLD, bevm.WD, bevmGS, bevmCS

bevmLD (**B**ird **E**ye's **V**iew **M**ap **L**ine **D**rawer)

書式：bevmLD depthTIFF maskTIFF

機能説明：

sliceBEVM.DS で作成された depth 画像を用いて、鳥瞰図上に３次元の線分を描くためのマスク画像を作成する。なお、線分のデータは標準入力から読み込む.

パラメータ：

depthTIFF

depth 画像のファイル名.

maskTIFF

鳥瞰図に重ね合わせることができるように隠線処理を加えつつ描いた３次元の線分の線画画像をおさめるファイル名.

使用例：

本プログラムは bevm.WD から起動される。使用法は bevm.WD を参照

３次元の線分データの書式：

標準入力から読み込まれる３次元の線分データを構成する点の座標 (x, y, z) はもとの３次元画像の画素の位置と同じ単位の値である（たとえば X 方向の画素数を Nx とすると、x = -0.5 〜 Nx - 1)。線分ごとに、

点の個数（N）
１番目の点の座標値（x1、y1、z1）
...
N 番目の点の座標値（xN、yN、zN）

のようなセットのデータを指定する。ただし、データは改行、空白もしくはタブコードで区切られたフリーフォーマットの数値である（N は整数でそれ以外は浮動少数点数)。

関連項目：

bevmLD, bevm.WD, sliceBEVM.DS, bevmCS

bevm.WD (**B**ird **E**ye's **V**iew **M**ap **W**indow **D**rawer)

書式：bevm.WD depthTIFF maskTIFF

機能説明：

bevmLD を用いて、鳥瞰図上にもとの３次元画像の領域を示す直方体の枠の線分を描くためのマスク画像を作成する.

パラメータ：

depthTIFF

depth 画像のファイル名.

maskTIFF

鳥瞰図に重ね合わせることができるように隠線処理を加えつつ描いたもとの３次元画像の領域を示す直方体の枠の線画の画像ファイル名.

使用例：

次の例は sliceBEVM.DS で作成された depth 画像を用い、3 次元ボックスを描いた画像を作成する。

bevm.WD mouse-bevm.d.tif mouse-bevm.w.tif

関連項目：

bevmLD, bevmGS, sliceBEVM.DS, bevmCS

bevmGS (**B**ird **E**ye's **V**iew **M**ap **G**ray scale image **S**ynthesizer)

書式：bevmGS labelTIFF shadeTIFF γ bias bg grayTIFF

機能説明：

sliceBEVM.DS で作成した label 画像と shade 画像を合成して８ビットのグレースケールで物体表面の陰影を表現した鳥瞰図画像を作成する.

パラメータ：

labelTIFF shadeTIFF

label 画像および shade 画像のファイル名.

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は０より大きい数値で、bias は０〜254 の整数.

bg

鳥瞰図上で物体がないために向こう側が透けて見える背景の部分の画素値。bg は０〜255 で、０を与えると背景は黒、255 なら白になる.

grayTIFF

グレースケールの鳥瞰図画像のファイル名.

使用例：

次の例は s**sliceBEVM.DS** で作成された mouse-bevm という名を持つ label 画像と shade 画像を用い、陰影処理を施した新しい断層画像 mouse.tif を作成する。

bevmGS mouse-bevm.s.tif mouse-bevm.l.tif 1 0 255 mouse.tif

鳥瞰図のグレースケールの計算法：

shade 画像におさめられた物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上のグレースケール I は以下のようにして計算される。γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

関連項目：

bevmLD, bevm.WD, sliceBEVM.DS, bevmCS

bevmO (**B**ird **E**ye's **V**iew **M**ap **O**blate)

書式：bevmO [x1 y1 z1 x2 y2 z2] scale λ_v ϕ_v λ_l ϕ_l newName

機能説明：

sliceOF で求めたような複数の 3 軸不等楕円体の軸方向および軸半径のデータを標準入力から読み込み、それらすべての楕円体の鳥瞰図画像を作成するための 4 種類の画像（depth、label、shade および edge 画像）を作成する。このプログラムが作成する 4 種類の画像は **sliceBEVM** が作成するものとほとんど同じ形式なので、**sliceBEVM** の場合と同じ手法でそれらから鳥瞰図を完成することができる.

パラメータ：

x1 y1 z1 x2 y2 z2

鳥瞰図の描画に用いられる 3 次元領域の広がりを指定する。この領域は標準入力から読み込まれた 3 軸不等楕円体全部がおさまるものでないといけない。指定を省略すると、すべての楕円体がおさまりきる 3 次元領域が鳥瞰図の描画領域となる。

scale

楕円体を鳥瞰図として投影する際に用いる XYZ 座標におけるスケーリングファクター（**sliceBEVM** の場合と異なり、X、Y、Z 方向のスケーリングファクターがすべて同じだと仮定する）.

λ_v ϕ_v

鳥瞰図の視線方向の経度と緯度（単位は度）.

λ_l ϕ_l

鳥瞰図の光源方向の経度と緯度（単位は度）.

newName

作成する 4 種類の画像ファイルの名前の一部となる文字列。この文字列の後に ".d.tif"（depth 画像)、".l.tif"（label)、".s.tif"（shade） および".e.tif"（edge）という拡張子が付け加わった名前のファイルにそれぞれの画像が書き込まれる。

使用例：

次の例は **sliceOF** で作成した楕円体のデータを用いて鳥観図を作成する。

tail +2 ovoid.data | cut -f6-11 | **bevmO** 1 20 20 30 40 bevm

最初の tail コマンドはクラスタ（物体像）の背景であるクラスタ番号 0 のクラスタを近似した楕円体のデータを取り除いている。クラスタ番号によって描画する楕円体を選択したければ、この次に head や tail コマンドをはさめばよい。次の cut コマンドは **sliceOF** が 1 行にまとめて出力するそれぞれの楕円体のデータのうち不要なものを取り除いている。

３軸不等楕円体のデータ：

bevmO はそれぞれの楕円体ごとに以下の９個のデータを標準入力から読み込む。これらは空白、タブもしくは改行コードで区切られたフリーフォーマットの浮動少数点数である

x0
y0
z0 楕円体の中心の座標値。
λ
ϕ
θ 楕円体の軸方向を示す角度（単位は度）。
A
B
C 楕円体の軸半径。

注 1：

楕円体全部がおさまる３次元領域の広がりは、楕円体のデータ読み込みが終了した時点で標準出力に書き出される。

注 2：

bevmO が用いる３次元座標系の長さの単位は入力した楕円体が用いている単位と同じものである。作成される鳥瞰図の画像の画素数などは **bevmO** の起動時に scale で指定されたスケーリングファクターによってほぼ決まる。

関連項目：

bevmLD, bevm.WD, sliceBEVM.DS, bevmCS,sliceOF

si_s_bev (**s**lice **i**mage **s**ingle value **b**ird **e**ye's **v**iwer)

書式：si_s_bev directory nameFile rangeList [VM] scaleX scaleY scaleZ γ bias bg fg GIF

機能説明：

スライス画像からポリゴンを用いた３次元画像上の物体像のグレースケールの鳥瞰図及びその視線方向を変えたアニメーションを作る.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前.“-” が与えられた場合、directory の中のすべての画像が選択される.

rangeList

物体像と見なす画素値の範囲.

VM

物体像の表面のデータを格納する仮想記憶（Virtual Memory）として使うテンポラリファイルの名前.省略できる.

scaleX scaleY scaleZ

画素の X、Y、Z 方向の辺の長さ（長さの単位長）を示すスケーリングファクター.

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は 0 より大きい数値で、bias は 0 ～ 254 の整数.

bg

鳥瞰図の背景の画素に与える画素値.

fg

鳥瞰図の上に元の３次元画像の直方体領域を表す枠を描く際に用いる画素値.

GIF

鳥瞰図（複数）を格納する GIF 形式画像ファイルの名前.

使用例：

次の例は BYTE ディレクトリの中のデータを用い、si_s_bev の標準入力に echo を使って、視点の経度、緯度が (120, 20) からみた鳥観図を作成する。

echo 120 20 | si_s_bev BYTE - 50-255 1 1 1 1 64 255 0 poly.gif

次の例は BYTE ディレクトリの中のデータを用い、si_s_bev の標準入力に echo を使って、視点の経度、緯度が (120, 20) から (0, 10) 刻みで 3 回移動させ、さらに (120, 50) から (10, 0) 刻みで 5 回移動させる鳥観図の動画を作成するよう入力する。

(echo 120 20 0 10 3 ; echo 120 50 10 0 5) | si_s_bev BYTE - 50-255 1 1 1 1 64 255 0 poly.gif

次の例は直接標準入力に指定することによって上と同様の動画を作成する。

si_s_bev BYTE - 50-255 1 1 1 1 64 255 0 poly.gif (リターン)
120 20 0 10 3(リターン)
120 50 10 0 5(リターン)
（コントロールを押しながら d）

次の例はこの内容を適当なテキストファイルに書いて保存し、ターミナル上で ”csh ファイル名 ” と入力することで実行できる。内容は同じ。

si_s_bev BYTE - 50-255 1 1 1 1 64 255 0 poly.gif << fin
120 20 0 10 3
120 50 10 0 5
fin

起動パラメータの指定：

rangeList

物体像と見なす画素値の範囲で、以下のような記法で指定する。

a	画素値 a
b-c	画素値 b∼c
-d	画素値 0∼d
e-	画素値 e∼ 画素値の最大値
f-g,h-i	画素値 f∼g もしくは h∼i
-j,k-	画素値 j 以下もしくは k 以上
-	すべての画素値（画素値 0∼ 画素値の最大値）

scaleX scaleY scaleZ

鳥瞰図の画像（正方形）の画素数を決めるパラメータである。３次元画像の画素の [x,y,z] 方向の辺長を３個の浮動小数点数で指定する。この場合、鳥瞰図の画像はこれらの値でスケールした物体像に応じた画素数になる。また、この 3 個のパラメータの代わりに、１個の自然数を size として指定するとその値が画像の横および縦の画素数になる。

γ bias

物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上の陰影の程度 I は以下のようにして計算される。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。通常は γ に 1〜2、また、bias に 16〜64 程度の値を指定すればよい (γ を大きな値にすると陰影の差が広がり、bias が 0 なら影の部分は真っ黒に、255 なら影がなくなる)

標準入力からの鳥瞰図の指定：

以下の２通りの形式のいずれかで視線方向を記述した行（行内の値は空白もしくはタブコードで区切る）を標準入力から指定する（数値は度単位で、フリーフォーマットの浮動少数点数)。

lon lat

鳥瞰図の視線方向の経度と緯度

bLon bLat sLon sLat count

これらのパラメータを使って、

for index= 0〜count-1

lon= bLon + sLon × index lat= bLat + sLat × index

として計算した経度（lon）と緯度（lat）の値の count 個の視線方向から眺めた複数の鳥瞰図それぞれを描く。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

si_m_bev (**s**lice **i**mage **m**ulti value **b**ird **e**ye's **v**iwer)

書式：si_m_bev directory nameFile colorFile [VM] scaleX scaleY scaleZ γ bias bgR bgG bgB fgR fgG fgB GIF

機能説明：

スライス画像からポリゴンを用いた３次元画像上の物体像のカラーの鳥瞰図及びその視線方向を変えたアニメーションを作る．

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前．"-" が与えられた場合、directory の中のすべての画像が選択される．

colorFile

画像上で物体像と見なさない画素値、もしくは物体像と見なす画素値とそれらに対する色の R、G、B 成分の値（0〜255、各行の複数の値は空白もしくはタブコードで区切る）を書き並べたテキストファイルの名前．

VM

物体像の表面のデータを格納する仮想記憶（Virtual Memory）として使うテンポラリファイルの名前．省略できる．

scaleX scaleY scaleZ

画素の X、Y、Z 方向の辺の長さ（長さの単位長）を示すスケーリングファクター．

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は 0 より大きい数値で、bias は 0 〜 254 の整数．

bgR bgG bgB

鳥瞰図の背景の画素に与える画素の RGB 値．

fgR fgG fgB

鳥瞰図の上に元の３次元画像の直方体領域を表す枠を描く際に用いる画素値．

GIF

鳥瞰図（複数）を格納する GIF 形式画像ファイルの名前．

使用例：

次の例は BYTE ディレクトリの中のデータを用い、si_m_bev の標準入力に echo を使って、視点の経度、緯度が (120, 20) からみた鳥観図を作成する。

echo 120 20 | si_m_bev BYTE - color.tbl 1 1 1 1 64 255 0 poly.gif

次の例は BYTE ディレクトリの中のデータを用い、si_m_bev の標準入力に echo を使って、視点の経度、緯度が (120, 20) から (0, 10) 刻みで 3 回移動させ、さらに (120, 50) から (10, 0) 刻みで 5 回移動させる鳥観図の動画を作成するよう入力する。

(echo 120 20 0 10 3 ; echo 120 50 10 0 5) | si_m_bev BYTE - color.tbl 1 1 1 1 64 255 0 poly.gif

次の例は直接標準入力に指定することによって上と同様の動画を作成する。

si_m_bev BYTE - color.tbl 1 1 1 1 64 255 0 poly.gif (リターン)
120 20 0 10 3(リターン)
120 50 10 0 5(リターン)
（コントロールを押しながら d）

次の例はこの内容を適当なテキストファイルに書いて保存し、ターミナル上で ”csh ファイル名 ” と入力することで実行できる。内容は同じ。

si_m_bev BYTE - color.tbl 1 1 1 1 64 255 0 poly.gif << fin
120 20 0 10 3
120 50 10 0 5
fin

起動パラメータの指定：

colorFile

物体像と見なす画素値の範囲を記述したファイルで、以下のような記法で指定する。

rL	rL の画素を物体像と見なさない
rL	rL の画素を物体像と見なさない
rL R G B	rL の画素を成分値 R、G、B の色で描く
rL R G B	rL の画素を成分値 R、G、B の色で描く
rL bR bG bB sR sG sB	rL の画素を (R,G,B) = (bR + sR × v, bG + sG × v, bB + sB × v)

ここで rL は range List の略であり、以下のようなフォーマットで表される

v	値 v の画素を物体像とする
v-	値 v 以上の画素を物体像とする
-v	値 v 以下の画素を物体像とする
v1-v2	v1 ～ v2 の値の画素を物体像とする
v1,v2,v3	値が v1、v2 もしくは v3 の画素を物体像とする。
v1,v2-v3	値が v1 以下か v2 ～ v3 の画素を物体像とする。

ただし、

[1] colorFile として ”-” を指定するとこれらの記述の行を標準入力から読み込む。
[2] 同じ画素値に複数の指定（物体像と見なさない指定や色成分の指定）がある場合、最後の指定が採用される。
[3] 最初の記述行が空行だったり、また、空のファイルを指定するなどして画素値の指定が行われなかった場合、以下のデフォルトの指定（画素値 0 の画素は表示せず、画素値 1 以上の画素は白色で表示する）が採用される。

0
1 画素値の最大値 255 255 255

scaleX scaleY scaleZ

鳥瞰図の画像（正方形）の画素数を決めるパラメータである。3 次元画像の画素の [x,y,z] 方向の辺長を 3 個の浮動小数点数で指定する。この場合、鳥瞰図の画像はこれらの値でスケールした物体像に応じた画素数になる。また、この 3 個のパラメータの代わりに、1 個の自然数を size として指定するとその値が画像の横および縦の画素数になる。

γ bias

物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上の陰影の程度 I は以下のようにして計算される。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。通常は γ に 1~2、また、bias に 16~64 程度の値を指定すればよい (γ を大きな値にすると陰影の差が広がり、bias が 0 なら影の部分は真っ黒に、255 なら影がなくなる)

標準入力からの鳥瞰図の指定：

以下の 2 通りの形式のいずれかで視線方向を記述した行（行内の値は空白もしくはタブコードで区切る）を標準入力から指定する（数値は度単位で、フリーフォーマットの浮動少数点数）。

lon lat

鳥瞰図の視線方向の経度と緯度

bLon bLat sLon sLat count

これらのパラメータを使って、

for index= 0~count-1

lon= bLon + sLon × index lat= bLat + sLat × index

として計算した経度（lon）と緯度（lat）の値の count 個の視線方向から眺めた複数の鳥瞰図それぞれを描く。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

slice.No

書式：slice.No directory new directory lsOptions

機能説明：

このプログラムはディレクトリ directory の中のファイル名を ls コマンドで並べ変えて new directory で指定される新しいディレクトリの下の０から始まる整数値＋拡張子”.tif” という名前のファイルとしてリンクする。なお、整数値（スライス番号）は最大のものに合わせて左側に ”0” が付け加えられる（たとえば 11 枚のスライスがある場合の最初のスライスのファイルは ”00.tif” となる）.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

new directory

新しいスライス画像ファイルを入れるディレクトリ名

lsOptions

ls コマンドのオプション指定。何も指定しなければ s コマンドのデフォルトであるアルファベット順にファイル名が並べ変えられる（”-r” とすれば逆順；”man ls” で ls コマンドのマニュアルを見よ）.

使用例：

次の例はディレクトリ BYTE の中のファイルを逆順に並べ替えたファイルを作成し、no ディレクトリに保存するためのスクリプト、no.sh を作成し、実際にそれを実行する。

slice.No BYTE no -r > no.sh

このコマンドはデバッグのため、実際にファイルをリンクせずそれを行なうためのコマンド行を表示するようにしてある。表示されたコマンドを確認した後に、

slice.No BYTE no -r | /bin/csh

のように出力をパイプを通して /bin/csh（C-shell）に流し込んでやれば実際にリンクする。

slice.NC

書式：slice.NC directory new directory lsOptions

機能説明：

LZW 圧縮されたデータの圧縮を解き、同じファイル名で格納し直す.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

new directory

新しいスライス画像ファイルを入れるディレクトリ名

使用例：

次の例はディレクトリ BYTE の中のファイルの LZW 圧縮をとく。

slice.NC BYTE nc

関連項目：

slice.LZW

slice.LZW

書式：slice.LZW directory new directory lsOptions

機能説明：

TIFF 形式のスライス画像のデータを画像毎に LZW 圧縮する、

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

new directory

新しいスライス画像ファイルを入れるディレクトリ名

使用例：

次の例はディレクトリ BYTE の中のファイルの LZW 圧縮する。

slice.LZW BYTE lzw

関連項目：

slice.NC

tiffdesc

書式：tiffdesc TIFF [desc newTIFF]

機能説明：

TIFF 形式の画像のタグ情報を表示、あるいは書き換えを行う．desc newTIFF を省略した場合はタグ情報の表示だけを行う

パラメータ：

TIFF

調べる tiff 画像

desc

書き換えるタグの情報

newTIFF

書き換えたタグを保存した、新しい Tiff 画像の名前

使用例：

次の例は test.tif のタグを表示する

tiffdesc test.tif

次の例は test.tif のタグを”test”に書き換える

tiffdesc test.tif test new.tiff

si_stl_B (**s**lice **i**mage stl **B**inary)

書式：si_stl_B directory nameFile PV1 PV2 [R G B] STL_FILE

機能説明：

スライス画像から読み込んだ 3 次元画像上で値域が PV1 PV2(PV1 と PV2 はともに整数値で $0 \leqq$ PV1 $\leqq$ PV2 $\leqq 65535$) の値を持つ画素を物体像と見なして、像の表面をできるだけ少ない個数の三角形で分割した STL データを作成する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

PV1

物体像と見なす画素値の範囲.

PV2

物体像と見なす画素値の範囲.

R G B

鳥観図にカラーをつけるときの RGB 値.

STL_FILE

ポリゴンデータを格納する STL 形式画像ファイルの名前.

使用例：

次の例は BYTE ディレクトリの中のデータを用い、画素値が 50 から 255 まで値を持つが素を物体と見なしたポリゴンデータを保存した stl ファイルを作成する。

si_stl_B BYTE - 50 255 poly.stl

標準エラー出力へのデータの出力形式：

si_stl_B は処理結果の 概要を示す以下の 12 個の整数値をタブコードで区切った 3 行にまとめて標準エラー出力に書き出す

3 次元画像の x、y、z 方向の画素数
3 次元画像上で物体像と見なした画素の総数
物体像の画素の座標値の最小値 (x、y、z 成分それぞれの値)

物体像の画素の座標値の最大値 (x、y、z 成分それぞれの値)
物体像の表面積 (表面に面する画素の面の総数)
物体像の表面を分割した三角形の総数

注：

なお、si_stl_[A,B] は物体像の画素に印をつけた 3 次元 2 値画像をメモリに保持するので、 実行に際してはその総画素数の 8 分の 1 に相当する bytes のメモリが最小限必要である。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

si_stl_C (**s**lice **i**mage stl **C**olor)

書式：si_stl_C directory nameFile STL_FILE

機能説明：

スライス画像から読み込んだ 3 次元画像上で値域が PV1 PV2(PV1 と PV2 はともに整数値で 0 ≦ PV1 ≦ PV2 ≦ 65535) の値を持つ画素を物体像と見なして、像の表面をできるだけ少ない個数の三角形で分割した STL データを作成する.

パラメータ：

directory

元のスライス画像ファイルのあるディレクトリ名

nameFile

使用するファイルリストファイルの名前. "-" が与えられた場合、directory の中のすべての画像が選択される.

STL_FILE

ポリゴンデータを格納する STL 形式画像ファイルの名前.

使用例：

次の例は BYTE ディレクトリの中のデータを用い、カラーポリゴンデータを保存した stl ファイルを作成する。

si_stl_C BYTE - poly_C.stl(リターン)
1 10 0 255 0 0 (リターン)
11 255 0 255 0 0 0 255(リターン)
（コントロールを押しながら d）

標準エラー出力へのデータの出力形式：

si_stl_C は処理結果の 概要を示す以下の 12 個の整数値をタブコードで区切った 3 行にまとめて標準エラー出力に書き出す

3 次元画像の x、y、z 方向の画素数
3 次元画像上で物体像と見なした画素の総数
物体像の画素の座標値の最小値 (x、y、z 成分それぞれの値)
物体像の画素の座標値の最大値 (x、y、z 成分それぞれの値)
物体像の表面積 (表面に面する画素の面の総数)
物体像の表面を分割した三角形の総数

標準入力から読み込むパラメータ指定行：

物体像と見なす画素値の範囲を記述したファイルで、以下のような記法で指定する。a
画素値 a の画素を物体像と見なさない
b c　　画素値 b∼c の画素を物体像と見なさない
d R G B　　画素値 d の画素を成分値 R、G、B の色で描く
e f R G B　　画素値 e∼f の画素を成分値 R、G、B の色で描く
g h R1 G1 B1 R2 G2 B2　画素値 i= g∼h の画素を以下の成分値の色で描く。

R= (int)(R1+(R2-R1)*(i-g)/(f-g)+0.5)
G= (int)(G1+(G2-G1)*(i-g)/(f-g)+0.5)
B= (int)(B1+(B2-B1)*(i-g)/(f-g)+0.5)

ただし、上式の右辺の計算では 0/0 == 0 と考える。

ただし、

[1] 同じ画素値に複数の指定（物体像と見なさない指定や色成分の指定）がある場合、最後の指定が採用される。
[2] 最初の記述行が空行だったり、また、空のファイルを指定するなどして画素値の指定が行われなかった場合、以下のデフォルトの指定（画素値０の画素は表示せず、画素値１以上の画素は白色で表示する）が採用される。

0
1 画素値の最大値 255 255 255

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

stl_bev_SS (stl **b**ird **e**ye's **v**iewmap **S**urface **S**moothing)

書式：stl_bev_SS STL_file scale γ bias bg fg

機能説明：

指定したファイル STL_files(複数も可) の中の STL データ が表す物体像を任意の視線方向から眺めた正射図法 (orthographic projection) の鳥瞰 図 (bird's eye viewmap) の画像 (正方形の画像; 複数も可) を作成して TIFF 形式画像ファ イルに格納する.

パラメータ：

STL_file

STL ファイル

scale

画素の辺の長さ（長さの単位長）を示すスケーリングファクター.

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は 0 より大きい数値で、bias は 0 ～ 254 の整数.

bg

鳥瞰図の背景の画素に与える画素値.

fg

鳥瞰図の上に元の 3 次元画像の直方体領域を表す枠を描く際に用いる画素値.

使用例：

次の例は poly.stl ファイルを用い、stil_bev_SS の標準入力に echo を使って、視点の経度、緯度が (120, 20) からみた鳥観図を作成する。

echo 120 20 poly.tif | stl_bev_SS poly.stl 1 1 64 255 0

次の例は次の例は poly.stl ファイルを用い、stl_bev_SS の直接標準入力に指定することによって鳥観図を作成する。

stl_bev_SS poly.stl 1 1 64 255 0 (リターン)
120 20 poly_120_20.tif(リターン)
120 50 poly_120_50.tif(リターン)
（コントロールを押しながら d）

次の例はこの内容を適当なテキストファイルに書いて保存し、ターミナル上で ”csh ファイル名 ” と入力することで実行できる。内容は同じ。

stl_bev_SS poly.stl 1 1 64 255 0 << fin
120 20 poly_120_20.tif
120 50 poly_120_50.tif
fin

起動パラメータの指定：

scale

鳥瞰図の画像（正方形）の画素数を決めるパラメータである。3次元画像の画素の辺長を浮動小数点数で指定する。この場合、鳥瞰図の画像はこの値でスケールした物体像に応じた画素数になる。

γ bias

物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上の陰影の程度 I は以下のようにして計算される。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。通常は γ に 1~2、また、bias に 16~64 程度の値を指定すればよい (γ を大きな値にすると陰影の差が広がり、bias が 0 なら影の部分は真っ黒に、255 なら影がなくなる)

標準入力からの鳥瞰図の指定：

視線方向と出力ファイル名を記述した行（行内の値は空白もしくはタブコードで区切る）を標準入力から指定する。

lon lat

鳥瞰図の視線方向の経度と緯度（数値は度単位で、フリーフォーマットの浮動少数点数)。

TIF_file

出力される鳥観図のファイル名。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

stl_bev_C_SS (stl **b**ird **e**ye's **v**iewmap **C**olor **S**urface **S**moothing)

書式：stl_bev_C_SS STL_file scale γ bias bgR bgG bgB fgR fgG fgB scR scG scB

機能説明：

指定したファイル STL_files(複数も可) の中の STL データ が表す物体像を任意の視線方向から眺めた正射図法 (orthographic projection) のカラー鳥瞰図 (bird's eye viewmap) の画像 (正方形の画像; 複数も可) を作成して TIFF 形式画像ファ イルに格納する.

パラメータ：

STL_file

STL ファイル

scale

画素の辺の長さ（長さの単位長）を示すスケーリングファクター.

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は 0 より大きい数値で、bias は 0 ～ 254 の整数.

bgR bgG bgB

鳥瞰図の背景の画素の RGB 値.

fgR fgG fgB

鳥瞰図の上に元の 3 次元画像の直方体領域を表す枠を描く際に用いる画素の RGB 値.

scR scG scB

色情報を持っていない三角形に塗 るべき色の RGB 値.

使用例：

次の例は poly_C.stl ファイルを用い、stil_bev_C_SS の標準入力に echo を使って、視点の経度、緯度が (120, 20) からみた鳥観図を作成する。

echo 120 20 poly.tif | stl_bev_C_SS poly_C.stl 1 1 64 255 255 255 0 0 0 0 0 0

次の例は次の例は poly_C.stl ファイルを用い、stl_bev_C_SS の直接標準入力に指定することによって鳥観図を作成する。

stl_bev_C_SS poly_C.stl 1 1 64 255 255 255 0 0 0 0 0 0(リターン)
120 20 poly_120_20.tif(リターン)
120 50 poly_120_50.tif(リターン)
（コントロールを押しながら d）

次の例はこの内容を適当なテキストファイルに書いて保存し、ターミナル上で ”csh ファイル名 ” と入力することで実行できる。内容は同じ。

```
stl_bev_C_SS poly_C.stl 1 1 64 255 255 255 0 0 0 0 0 0<< fin
120 20 poly_120_20.tif
120 50 poly_120_50.tif
fin
```

起動パラメータの指定：

scale

鳥瞰図の画像（正方形）の画素数を決めるパラメータである。3次元画像の画素の辺長を浮動小数点数で指定する。この場合、鳥瞰図の画像はこの値でスケールした物体像に応じた画素数になる。

γ bias

物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上の陰影の程度 I は以下のようにして計算される。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。通常は γ に 1~2、また、bias に 16~64 程度の値を指定すればよい (γ を大きな値にすると陰影の差が広がり、bias が 0 なら影の部分は真っ黒に、255 なら影がなくなる)

標準入力からの鳥瞰図の指定：

視線方向と出力ファイル名を記述した行（行内の値は空白もしくはタブコードで区切る）を標準入力から指定する。

lon lat

鳥瞰図の視線方向の経度と緯度（数値は度単位で、フリーフォーマットの浮動少数点数）。

TIF_file

出力される鳥観図のファイル名。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

stl_bev_GIF_SS (stl **b**ird **e**ye's **v**iewmap **G**if **S**urface **S**moothing)

書式：stl_bev_GIF_SS STL_file scale γ bias bg fg GIF_file

機能説明：

指定したファイル STL_files(複数も可) の中の STL データ が表す物体像を任意の視線方向から正射図法の鳥瞰図を描き、それらすべての画像を GIF 形式画像ファイル GIF_file("-" の指定で標準出力に書き込む) にまとめて格納する

パラメータ：

STL_file

STL ファイル

scale

画素の辺の長さ（長さの単位長）を示すスケーリングファクター.

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は 0 より大きい数値で、bias は 0 ～ 254 の整数.

bg

鳥瞰図の背景の画素に与える画素値.

fg

鳥瞰図の上に元の 3 次元画像の直方体領域を表す枠を描く際に用いる画素値.

GIF_file

出力される鳥観図（動画 gif）のファイル名。

使用例：

次の例は次の例は poly.stl ファイルを用い、stl_bev_GIF_SS の直接標準入力に指定することによって鳥観図の動画を作成する。まず (0,0) から経度方向に 10 度ずつ 5 回動かし、その後 経度方向に 5 度ずつ 6 回動かしている

stl_bev_GIF_SS poly.stl 1 1 64 255 0 poly.gif(リターン)
0 0 10 0 5 (リターン)
50 0 0 5 6(リターン)
（コントロールを押しながら d）

起動パラメータの指定：

scale

鳥瞰図の画像（正方形）の画素数を決めるパラメータである。3次元画像の画素の辺長を浮動小数点数で指定する。この場合、鳥瞰図の画像はこの値でスケールした物体像に応じた画素数になる。

γ bias

物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上の陰影の程度 I は以下のようにして計算される。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。通常は γ に 1~2、また、bias に 16~64 程度の値を指定すればよい (γ を大きな値にすると陰影の差が広がり、bias が 0 なら影の部分は真っ黒に、255 なら影がなくなる)

GIF_file

出力される鳥観図のファイル名。

標準入力からの鳥瞰図の指定：

視線方向と出力ファイル名を記述した行（行内の値は空白もしくはタブコードで区切る）を標準入力から指定する。

lon lat

鳥瞰図の視線方向の経度と緯度（数値は度単位で、フリーフォーマットの浮動少数点数)。

lon_base lat_base lon_step lat_step step number

鳥瞰図の視線方向の経度と緯度の初期値、それぞれの 1 ステップでの移動量と、トータルのステップ数（数値は度単位で、フリーフォーマットの浮動少数点数)。

注 1：

stl_bev_GIF_SS などで作成した鳥瞰図の画像は正確には GIF87a 形式でファイルに格納される。このため、再生が自動で始まらない場合などがある。この場合は以下のようにして GIF89a 形式に変換してやると良い

gif_movie . -0_0_96.gif 0 65536 new_0_0_96.gif

この例では 0_0_96.gif を new_0_0_96.gif に変換している。名前ファイルではなく、ファイルを直接を指定するときは 0_0_96.gif の前にはマイナスを付ける。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**, **gif_movie**

stl_bev_C_GIF_SS (stl **b**ird **e**ye's **v**iewmap **C**olor **G**if **S**urface **S**moothing)

書式：stl_bev_C_GIF_SS STL_file scale γ bias bgR bgG bgB fgR fgG fgB scR scG scB GIF_file

機能説明：

指定したファイル STL_files(複数も可) の中の STL データ が表す物体像を任意の視線方向から正射図法のカラー鳥瞰図を描き、それらすべての画像を GIF 形式画像ファイル GIF_file("-"の指定で標準出力に書き込む) にまとめて格納する

パラメータ：

STL_file

STL ファイル

scale

画素の辺の長さ（長さの単位長）を示すスケーリングファクター.

γ bias

物体像の陰影の程度を決める浮動小数点数の数値。γ は 0 より大きい数値で、bias は 0 ～ 254 の整数.

bgR bgG bgB

鳥瞰図の背景の画素の RGB 値.

fgR fgG fgB

鳥瞰図の上に元の 3 次元画像の直方体領域を表す枠を描く際に用いる画素の RGB 値.

scR scG scB

色情報を持っていない三角形に塗 るべき色の RGB 値.

GIF_file

出力される鳥観図（動画 gif）のファイル名。

使用例：

次の例は次の例は poly_C.stl ファイルを用い、stl_bev_C_GIF_SS の直接標準入力に指定することによって鳥観図の動画を作成する。まず (0,0) から経度方向に 10 度ずつ 5 回動かし、その後 経度方向に 5 度ずつ 6 回動かしている

stl_bev_C_GIF_SS poly.stl 1 1 64 255 255 255 0 0 0 0 0 0 poly_C.gif(リターン)
0 0 10 0 5 (リターン)
50 0 0 5 6(リターン)
（コントロールを押しながら d）

起動パラメータの指定：

scale

鳥瞰図の画像（正方形）の画素数を決めるパラメータである。3次元画像の画素の辺長を浮動小数点数で指定する。この場合、鳥瞰図の画像はこの値でスケールした物体像に応じた画素数になる。

γ bias

物体表面の点の法線ベクトルと光源方向のベクトルの内積の値を P とすると鳥瞰図上の陰影の程度 I は以下のようにして計算される。

$$I = bias + (255 - bias) \times (P/255)^{(1/\gamma)}$$

γ はコンピュータグラフィックスでグレースケールの表示輝度の補正に用いられるガンマファクターと同じものと考えればよい。通常は γ に 1~2、また、bias に 16~64 程度の値を指定すればよい (γ を大きな値にすると陰影の差が広がり、bias が 0 なら影の部分は真っ黒に、255 なら影がなくなる)

GIF_file

出力される鳥観図のファイル名。

標準入力からの鳥瞰図の指定：

視線方向と出力ファイル名を記述した行（行内の値は空白もしくはタブコードで区切る）を標準入力から指定する。

lon lat

鳥瞰図の視線方向の経度と緯度（数値は度単位で、フリーフォーマットの浮動少数点数)。

lon_base lat_base lon_step lat_step step number

鳥瞰図の視線方向の経度と緯度の初期値、それぞれの 1 ステップでの移動量と、トータルのステップ数（数値は度単位で、フリーフォーマットの浮動少数点数)。

注 1：

stl_bev_C_GIF_SS などで作成した鳥瞰図の画像は正確には GIF87a 形式でファイルに格納される。このため、再生が自動で始まらない場合などがある。この場合は以下のようにして GIF89a 形式に変換してやると良い

gif_movie . -0_0_96.gif 0 65536 new_0_0_96.gif

この例では 0_0_96.gif を new_0_0_96.gif に変換している。名前ファイルではなく、ファイルを直接を指定するときは 0_0_96.gif の前にはマイナスを付ける。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**, **gif_movie**

of_stl_ih (**o**void **f**it stl **i**cosa**h**edron)

書式：of_stl_ih level [xO yO zO λ ϕ θ A B C] [R G B] STL_FILE

機能説明：

指定された 3 軸不等楕円体の形状を近似する正 20 面体 (icosahedron;ih) を表す STL データを作成し、それを binary もしくは color STL 形式でファイル STL_file(“-”を指定した場合は標準出力) に書き込む.

パラメータ：

level

0 以上の整数値で、楕円体の像を近似する正多面体の表面の正三角形の細分化の繰り返しの回数

xO yO zO

楕円体の中心の座標値.

λ ϕ θ

度単位の角度で、λ と ϕ は楕円体の c 軸の方向を示す経度と緯度、θ は b(もしくは a) 軸の方向を示す角度.

A B C

相互に直交する楕円体の 3 軸 (a、b、c 軸) 方向の半径の値.

R G B

鳥観図にカラーをつけるときの RGB 値.

STL_FILE

ポリゴンデータを格納する STL 形式画像ファイルの名前. “-” が与えられた場合、標準出力に ASCII データが出力される.

使用例：

次の例は sliceOF で出力されたデータを保存した of.txt を用い、xO, yO, zO, λ, ϕ, θ A, B, C を抽出して of_stl_ih の標準入力に送り、stl ファイルを作成する。

tail -n+2 of.txt | cut -f3-5,9-14 | of_stl_ih 7 of.stl

標準出力からのデータの入力形式：

起動時に楕円体の形状パラメータ (xO, yO, zO, λ, ϕ, θ A, B, C) の指定がない場合には標準入力からそれらの読み込みが行われる。

xO yO zO λ ϕ θ A B C
level xO yO zO λ ϕ θ A B C
xO yO zO λ ϕ θ A B C R G B
level xO yO zO λ ϕ θ A B C R G B

このとき、厳密には STL 形式の仕様に反することだが、プログラム of_stl_[i,o,t]h はそれぞれの楕円体を近似した複数の多面体の STL データをひとつのファイルにまとめて書き込むことができる。すなわち、上記のように 4 種類の記述法が許されている標準入力からの指定は異なる種類のものを交えて複数回行ってもよい。その際に上記の 2 番目以降の記述法を使えば楕円体 (多面体) ごとに level の値や色の情報 (R, G,B) を個別に指定することもできる。なお、上記の 1 番目の記述法のように level などの指定を省略すると起動時に指定された値が用いられる

楕円体の多面体近似について：

プログラム of_stl_ih、of_stl_oh および of_stl_th はそれぞれ正 20 面体 (icosahedron;ih)、 正 8 面体 (octahedron;oh) および正 4 面体 (tetrahedron;th) の表面の正三角形のそれぞれ を以下に記す (再帰的な) 手続きで細分化した多面体を用いて楕円体の像を近似している:
(0) まず、球に内接する適当な正多面体 (前記の 3 つの正多面体のいずれか) を用意する。
(1) 多面体の三角形それぞれを 3 辺の中点を新しい頂点として 4 個の三角形に分割し、それらの新しい頂点を球の中心から投影した球面上の点に移動する。このような三角形の細分化処理を球 (球面) の近似に十分な多面体が構築できるまで繰り返す。
(2) 最後に、このような球 (球面) を近似する多面体を長、中、短軸の方向にそれぞれ伸縮して近似楕円体に合わせた形状に変形する。
上の処理 (1) の三角形の細分化の繰り返しの回数を of_stl_ih などの起動パラメータ level で指定する。ただし、level に 0 を指定すると三角形の細分化は行われず、元の正多面体を変形したもので楕円体を表すことになる。元の正多面体にもよるが、level として概ね 7 以上の値を指定すれば楕円体の表面を表すのに十分な多面体 (STL データ) になるハズである。
なお、上記の手順で得られる楕円体を近似する STL データ中の三角形の総数は、処理に用いた元の正多面体の面の数に 4^{level} を乗じた値となる。したがって、of_stl_[i,o,t]h によって複数の楕円体をサイズに応じた面の数を持つ多面体で近似したい場合は、それぞれの楕円体に対して体積 (もしくは代表的な軸半径) の対数に比例するような非負の整数値を level として指定してやればよい。

関連項目：

si_m_bev, **si_s_stl**, **si_stl_A**, **si_stl_B**, **si_stl_C**, **stl_bev**, **stl_bev_C**, **stl_bev_SS**, **stl_bev_C_SS**, **stl_bev_GIF**, **stl_bev_GIF_SS**, **stl_bev_C_GIF**, **stl_bev_C_GIF_SS**, **stl_bev_wf**, **stl_bev_wf_nf**

索 引