

ペアの画像の位置決め (registration)

1. 位置決め処理の概要

立方体画素の辺長と物体像の3次元剛体運動（回転と並進）だけを考慮した affine 変換（線形な座標変換）でペアの画像の配置が同じになると仮定する。つまり、3次元画像それぞれの諸元を

画素の辺長が小さい画像（7 keV で撮影した画像；7 keV 画像）

画素の辺長： α

画素数： $N_x \times N_y \times N_z$

座標値： $(x, y, z) = (0, 0, 0) \sim (N_x-1, N_y-1, N_z-1)$

画素の辺長が大きい画像（8 keV で撮影した画像；8 keV 画像）

画素の辺長： β ($\geq \alpha$)

画素数： $N_h \times N_v \times N_d$

座標値： $(h, v, d) = (0, 0, 0) \sim (N_h-1, N_v-1, N_d-1)$

として、これらの座標値の間に以下の affine 変換の関係式が成り立つものとする：

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_h & x_v & x_d \\ y_h & y_v & y_d \\ z_h & z_v & z_d \end{pmatrix} \begin{pmatrix} h \\ v \\ d \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

ただし、上式の右辺の第一項目は等方な伸長と剛体回転を表している：

$$\begin{pmatrix} x_h & x_v & x_d \\ y_h & y_v & y_d \\ z_h & z_v & z_d \end{pmatrix} = \frac{\beta}{\alpha} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

ここでは指定した「ペアの画像それぞれの画素の辺長の比の値 ($\gamma = \beta / \alpha$)」と「それらの画像の上の物体像の同じ位置を指す点（マーカ） $i = 1 \sim N$ の座標値のペア (x_i, y_i, z_i) と (h_i, v_i, d_i) 」から以下の式で計算した「7 keV 画像上の画素の幅を単位とするマーカの距離の平均残差 (mean residual) R 」が最小になる affine 変換の行列とベクトルの成分値 ($x_h, x_v, x_d, y_h, y_v, y_d, z_h, z_v, z_d, x_0, y_0$ および z_0) を決める：

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} - \left\{ \begin{pmatrix} x_h & x_v & x_d \\ y_h & y_v & y_d \\ z_h & z_v & z_d \end{pmatrix} \begin{pmatrix} h_i \\ v_i \\ d_i \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \right\}$$

$$R_i^2 = X_i^2 + Y_i^2 + Z_i^2$$

$$R^2 = \frac{1}{N} \times \sum_{i=1}^N R_i^2$$

なお、上式の (X_i, Y_i, Z_i) と R_i はそれぞれマーカ $i = 1 \sim N$ の「7 keV 画像の上での（画素の幅を単位とする）座標値と距離の残差 (residual)」である。

2. 位置決め処理の具体的な手順

2-1. マーカのペアの座標値の読み取り

ペアの画像から読み取ったそれぞれのマーカの6個の座標値 $x_i, y_i, z_i, h_i, v_i, d_i$ をこの順で、空白もしくはタブコード区切りで1行にまとめたテキストファイルを作成する。ただし、空白やタブコードで区切ってあれば、これら6個の座標値の後ろに余分な値（文字列）が書き込まれていても良い（次に説明するプログラム `check_marker` はそれを読み飛ばす）。

2-2. affine 変換のパラメータの決定

多数のマーカの6個の座標値を各行に書き込んだテキストファイルを”`marker.txt`”として、Linux や Windows の端末から以下のいずれかを入力する：

```
check_marker  γ  marker.txt
check_marker  γ  marker.txt  residual.txt
```

いずれの場合もプログラム `check_marker` は前記の方法でペアの画像の `affine` 変換に使うパラメータの値（下記の16個の値）を決め、それらをタブコードで区切って並べた6行を標準出力に書き出す（これは適当なファイルにリダイレクトした方が良い）：

| | | | | |
|-----|----|----|----|-----------------------|
| [1] | A | B | C | ← 回転角 (Euler 角; 単位は度) |
| [2] | xh | xv | xd | ← 等方な伸長と剛体回転を表す行列の成分値 |
| [3] | yh | yv | yd | ← ” |
| [4] | zh | zv | zd | ← ” |
| [5] | x0 | y0 | z0 | ← 並進を表すベクトルの成分値 |
| [6] | R | | | ← 平均残差 (単位は画素幅) |

`check_marker` に2個目のファイル名”`residual.txt`”を指定すると、”`marker.txt`”の各行に7個目の値としてマーカの位置の距離に関する残差 R_i の値を追記したテキストファイルを作成する。**この値を参考にしてマーカの座標値の修正や削除を行えば良い。**そして、`check_marker` は各行の7個目以降の値を読み飛ばすので、”`marker.txt`”の代わりに”`residual.txt`”を `check_marker` に指定することもできる。

2-3. 8 keV 画像を7 keV 画像と同じ配置および画素数の画像に変換する処理

プログラム `si_tli` を使えば8 keV 画像を7 keV 画像と同じ空間配置かつ画素数構成の画像に変換できる。そのソースや Windows 用の実行ファイルなどは以下の書庫ファイルに入っている：

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

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

端末から以下のように入力して `si_tli` を起動する：

```
si_tli orgDir - (改行しない)
      xh xv xd yh yv yd zh zv zd (改行しない)
      -x0 -y0 -z0 Nx Ny Nz (改行しない)
      PV (改行しない)
      newDir
```

ただし、

`orgDir`

オリジナルの 8 keV 画像のスライス画像が入っているディレクトリの名前

“_”

ここでは `orgDir` にスライス画像のファイルだけが入っていると仮定している。

`xh`、`xv`、`xd`、`yh`、`yv`、`yd`、`zh`、`zv`、`zd`

`check_marker` が出力した `affine` 変換の行列の成分値そのものを指定する。

`-x0`、`-y0`、`-z0`

`check_marker` が出力した値の符号を反転した値を指定する。

`Nx`、`Ny`、`Nz`

7 keV 画像の画素数。ここではこの画素数の構成の画像を作成する（先に指定した行列の成分値によって、この画像の画素の辺長は 7 keV 画像のものと同じになる）。

`PV`

`affine` 変換によって生じた「余白」を埋める画素値。通常は CT 値 0 に相当する画素値を指定する。8 keV 画像が `byte` 画像なら 0 を指定すれば良いが、`word` 画像の場合には画素値と CT 値の対応関係を考慮してその値を算出する必要がある。

`newDir`

8 keV 画像を空間補間しながら `affine` 変換した結果の 3 次元画像のスライス画像のファイルを入れるディレクトリの名前。事前に自分で作成しておく必要がある。

3. 位置決めしたペアの画像の応用（2次元ヒストグラムの作成）

同じ空間配置かつ画素数構成にしたペアの画像からそれらの上の同じ位置の画素の値（画素値）のペアの出現頻度分布（2次元ヒストグラム）を抽出する。そのためにプログラム `si_pair_hg` を書いた。ただし、これは以下の書庫ファイルに入っているプログラム `si_dmp` を呼び出して処理を行うため、自立した汎用プログラムではない。

<http://www-bl20.spring8.or.jp/~sp8et/tmp/rar.zip>

<http://www-bl20.spring8.or.jp/~sp8et/tmp/rar.taz>

`si_pair_hg` の起動法は以下の通りである：

```
si_pair_hg xDir xBase xStep xPels yDir yBase yStep yPels > hg.txt
```

ここで、

xDir もしくは **yDir**

2次元ヒストグラムの **x** もしくは **y** 座標値とする画素値を取り出す「3次元画像のスライス画像のファイル」が入っているディレクトリの名前。**si_pair_hg** はこれらのディレクトリにスライス画像のファイルだけが入っていると仮定している。

xBase もしくは **yBase**

2次元ヒストグラムの **x** もしくは **y** 座標値 (== 各画像の画素値) の最小値

xStep もしくは **yStep**

ヒストグラムの **x** もしくは **y** 方向の **bin** の幅 (== ひとまとめにする画素値の階調数)

xPels もしくは **yPels**

ヒストグラムの **x** もしくは **y** 方向の **bin** の数 (== ヒストグラム画像の画素数)

hg.txt

以下の形式の2次元ヒストグラムのテキストデータを書き込むファイルの名前：

1行目

xPels、**yPels** と「ヒストグラムの2次元 **bin** に落ちた画素のペアの個数の最大値」の合計の3個の値がタブコード区切りで並んでいる。

2～**xPels**×**yPels** + 1行

それぞれの **bin** に落ちた画素のペアの個数がその2次元配列のスキャンライン順 (**x** 方向優先) で各行に書き込まれている。

hg.txt に入れた2次元ヒストグラムのテキストデータを **TIFF** 画像に変換する際には **bin** の値を適切な画素値に変換する必要がある。例えば、その値を対数変換し、出現した最大の値で正規化した画素値に変換するには **Linux** で以下のように入力すれば良い：

```
awk -f log_8bit.awk hg.txt | pnmtotiff > hg.tif
```

ここで、”**log_8bit.awk**” は **bin** の値の対数変換と8ビット画素値化を行うプログラム言語 **awk** のスクリプト (テキストファイル) で、具体的には以下の2行からなる：

```
NR==1 { print "P2",$1,$2,255; scale=255/log($3) }
```

```
NR>=2 { print ($1<=1)?0:int(scale*log($1)+0.5) }
```

また、以下の **E-mail** で説明しているプログラム **tcm_rgb** を使えば、この2次元ヒストグラムの **TIFF** 画像 **hg.tif** を擬似カラーの **TIFF** 画像 **rgb.tif** に変換できる。

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

さらに、以下の **E-mail** で説明しているプログラム **sr_map** を使えば、画像 **hg.tif** (および **rgb.tif**) から2次元ヒストグラムの陰影図の **TIFF** 画像 **sr.tif** を作成することもできる。

http://www-bl20.spring8.or.jp/~sp8ct/tmp/sr_map.pdf

上記の E-mails に記されているように、プログラム `tcm_rgb` と `sr_map` のソースや Windows 用の実行ファイルはすべて以下の書庫ファイルに入っている：

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

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

これらのプログラムを用いた 2 次元ヒストグラムの（表示用の）画像の作成手順は以下の通り。

擬似カラー画像の作成法

```
tcm_rgb hg.tif cm_rgb_b.txt rgb.tif
```

ただし、”`cm_rgb_b.txt`”は 8 ビットの画素値 0～255 に対して「青、シアン、緑、黄、赤、マゼンタの順に徐々に変化する表示色」を割り当てた「カラーマップ」のデータファイルで、以下の書庫ファイルを展開したディレクトリ “`ecs/etc/`” に入っている：

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

陰影図画像の作成法

グレースケールの陰影図画像

```
sr_map hg.tif 1 -30 60 0.5 sr.tif
```

カラーの陰影図画像

```
sr_map hg.tif 1 rgb.tif -30 60 0.5 sr.tif
```

ただし、`sr_map` に指定したパラメータの数値の意味は以下の通り。

1 : 画素値を「高度」に変換する際のファクタ

-30 と 60 : 陰影付けのための光源の方向の度単位の「方位角」と「仰角」

0.5 : 陰影付けの効果を定める 0～1 の値。0 を指定すると陰影付けをしない。

捕逸

4. プログラム `check_marker` の改造

画素の 3 つの辺長が違っている画像のペアの位置決めにも使えるように `check_marker` を改造した。従来の起動法（下記の [a]）に加えて新しい起動法（[b] と [c]）を追加した。

```
[a] check_marker ratio marker.txt {residual.txt}
```

```
[b] check_marker Uxyz Uh Uv Ud marker.txt {residual.txt}
```

```
[c] check_marker Ux Uy Uz Uh Uv Ud marker.txt {residual.txt}
```

ただし、新しい起動法で指定する必要があるパラメータの意味は以下の通り。

U_{xyz} : 7 keV 画像の画素の辺長が同じ値なら、その 1 個の値だけを指定すれば良い。

U_x, U_y, U_z : 7 keV 画像の x、y、z 方向の画素の辺長

U_h, U_v, U_d : 8 keV 画像の h、v、d 方向の画素の辺長

このような起動法の違いに伴う `check_marker` の処理内容の違いはない。つまり、**[b]**や**[c]**で起動した場合も `check_marker` は**[a]**の場合と同等な式から得た同内容の計算結果(後述)を出力する。

5. affine 変換のパラメータの決定法

5-1. 平均残差が停留値になる並進ベクトルの成分値

$$\begin{pmatrix} \langle x \rangle \\ \langle y \rangle \\ \langle z \rangle \end{pmatrix} = \begin{pmatrix} \langle x \rangle \\ \langle y \rangle \\ \langle z \rangle \end{pmatrix} - \begin{pmatrix} x_h & x_v & x_d \\ y_h & y_v & y_d \\ z_h & z_v & z_d \end{pmatrix} \begin{pmatrix} \langle h \rangle \\ \langle v \rangle \\ \langle d \rangle \end{pmatrix}$$

$$\begin{pmatrix} \langle x \rangle \\ \langle y \rangle \\ \langle z \rangle \end{pmatrix} = \frac{1}{N} \times \sum_{i=1}^N \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$$

$$\begin{pmatrix} \langle h \rangle \\ \langle v \rangle \\ \langle d \rangle \end{pmatrix} = \frac{1}{N} \times \sum_{i=1}^N \begin{pmatrix} h_i \\ v_i \\ d_i \end{pmatrix}$$

5-2. それぞれのマーカの座標値の残差

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_i - \langle x \rangle \\ y_i - \langle y \rangle \\ z_i - \langle z \rangle \end{pmatrix} - \begin{pmatrix} x_h & x_v & x_d \\ y_h & y_v & y_d \\ z_h & z_v & z_d \end{pmatrix} \begin{pmatrix} h_i - \langle h \rangle \\ v_i - \langle v \rangle \\ d_i - \langle d \rangle \end{pmatrix}$$

5-3. 画素の 3 つの辺長が違っているペアの画像に対する affine 変換の行列

$$\begin{pmatrix} x_h & x_v & x_d \\ y_h & y_v & y_d \\ z_h & z_v & z_d \end{pmatrix} = \begin{pmatrix} XH \times U_h/U_x & XV \times U_v/U_x & XD \times U_d/U_x \\ YH \times U_h/U_y & YV \times U_v/U_y & YD \times U_d/U_y \\ ZH \times U_h/U_z & ZV \times U_v/U_z & ZD \times U_d/U_z \end{pmatrix}$$

$$\begin{pmatrix} XH & XV & XD \\ YH & YV & YD \\ ZH & ZV & ZD \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos A & -\sin A \\ 0 & \sin A & \cos A \end{pmatrix} \begin{pmatrix} \cos B & 0 & \sin B \\ 0 & 1 & 0 \\ -\sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos C & -\sin C & 0 \\ \sin C & \cos C & 0 \\ 0 & 0 & 1 \end{pmatrix}$$