1 MOSAIC CAMERA GEOMETRY FITTING

1.1 The Problem

We assume that the telescope produces a distorted image of the sky on the detector plane, with some simple parameterised form for distortion, and that the CCD chips are roughly positioned on a regular grid, but with small offsets and rotations. Given a set of dithered observations we want to determine

- parameters of the distortion model
- shifts and rotations of the chips relative to asumed grid
- global shift and linear transformations for dithered fields

The method of solution described here is

1. Convert pixel positions to 'nominal' detector frame coords (i.e. assuming chips are perfectly positioned on the grid).
2. Guess parameters for chip rotations, offsets and distortion (e.g all zero).
3. For each exposure, from the lists of star coordinates from the individual chips generate a 'superlist' containing approximate sky positions determined using guessed parameters and chip ID.
4. For a pair of exposures, solve approximately for relative shift, rotation of 2nd exposure.
5. Transform 2nd superlist; merge superlists (with some coarse tolerance); and untransform coordinates from 2nd exposure.
6. Un-apply any non trivial transformation in step 3 so we're back to nominal chip coords.
7. Using pairs of these coordinates from merged superlists solve for all parameters in a linearised manner.

Now using these improved parameters we can

1. Apply transformation from nominal coords to sky coords using current parameters (including shift, rotation to bring second exposure coords onto first).
2. Merge lists using fine tolerance
3. Apply inverse of tranformation in step 1.
4. Solve for all parameters using improved merged list.

and iterate if necessary.

In fact we have more than one pair of pointings so we really want to solve for everything simultaneously using all pairings of objects....

The goal is that with these tranformations we can map the individual images from each exposure onto a single image so we can then process a stack of such images to do cosmic ray removal and averaging.

1.2 Mathematical formulation

The starting point is a set of catalogues for $\sim$ 50 or so stars per chip per exposure giving integer pixel coords i_x, i_y. Let chips have nominal size $N_x \times N_y$ (2048×4096).

Our first task is to generate a first approximation to rectilinear sky coords based on assumtions re size, spacing and orientation of chips. Call these coordinates 'nominal detector coords'. We generate these as follows:

First we make transformation for upper row of chips (which are 'upside down':

$$\begin{array}{l} i_x \rightarrow N_x - i_x - 1 \\ i_y \rightarrow N_y - i_y - 1 \end{array} \tag{1}$$

We then model chips as mounted on blocks of size $(N_x + 2M_x) \times (N_y + 2M_y)$ so M_x, M_y are the margin widths. We assign chips an integer position (I_x, I_y) where $I_x = (-2, ..., 1)$, $I_y = (-1, 0))$. (Bottom left chip is (-2,-1); chip-5 has readout pixel at centre of grid and has position $(0, 0)$.

Generate 'nominal detector coords' for cth chip and eth exposure

$$\begin{bmatrix} x_{ce} \\ y_{ce} \end{bmatrix} = \begin{bmatrix} I_x * (N_x + 2M_x) + M_x + i_x \\ I_y * (N_y + 2M_y) + M_y + i_y \end{bmatrix} \tag{2}$$

from the pixel positions i, j and chip position. We assume these are related to perfect rectilinear detector coords (x, y) by

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} 1 & \phi_c \\ -\phi_c & 1 \end{bmatrix} \begin{bmatrix} x_{ce} \\ y_{ce} \end{bmatrix} + \begin{bmatrix} dx_c \\ dy_c \end{bmatrix} \tag{3}$$

We assume that perfect detector coords are related to locally rectilinear sky coords by

$$\begin{bmatrix} X_e \\ Y_e \end{bmatrix} = (1 + \alpha(x_e^2 + y_e^2)) \begin{bmatrix} x_e \\ y_e \end{bmatrix} \tag{4}$$

for some fiducial telescope pointing and orientation, and that absolute sky coordinates (X, Y) are given by

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X_e \\ Y_e \end{bmatrix} + \begin{bmatrix} \Phi_e^{00} & \Phi_e^{01} \\ \Phi_e^{10} & \Phi_e^{11} \end{bmatrix} \begin{bmatrix} X_e \\ Y_e \end{bmatrix} + \begin{bmatrix} dX_e \\ dY_e \end{bmatrix} \tag{5}$$

Let chip number c run from 0 to $N_c - 1$ and let exposure number e run from 0 to $N_e - 1$ We set $\phi_0 = dx_0 = dx_0 = \Phi_0 = dX_0 = dY_0 = 0$, and linearise in the $1+3(N_e-1)+3(N_c-1)$ assumed very small parameters α; Φ_m, dX_m, dY_m, $m=1\ldots N_e-1$; and ϕ_n, dx_n, dy_n, $n=1\ldots N_c-1$ to obtain

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} x_{ce} \\ y_{ce} \end{bmatrix} + \alpha(x_{ce}^2+y_{ce}^2)\begin{bmatrix} x_{ce} \\ y_{ce} \end{bmatrix} + \begin{bmatrix} \Phi_e^{00} & \Phi_e^{01} \\ \Phi_e^{10} & \Phi_e^{11} \end{bmatrix}\begin{bmatrix} x_{ce} \\ y_{ce} \end{bmatrix} + \begin{bmatrix} dX_e \\ dY_e \end{bmatrix} + \phi_c \begin{bmatrix} y_{ce} \\ -x_{ce} \end{bmatrix} + \begin{bmatrix} dx_c \\ dy_c \end{bmatrix} \tag{6}$$

To determine coefficients we need to generate lists of pairs of positions (X,Y) (X',Y') for same object and minimise

$$\chi^2 = \sum(\Delta X^2 + \Delta Y^2) \tag{7}$$

where $\Delta X \equiv X - X'$ etc.

Packaging the variables in a vector V_I we can write

$$\begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = \begin{bmatrix} x_{ce} - x_{c'e'} \\ y_{ce} - y_{c'e'} \end{bmatrix} + \begin{bmatrix} C_I^x V_I \\ C_I^y V_I \end{bmatrix} \tag{8}$$

where

$$V_I = \begin{bmatrix} \alpha \\ \vdots \\ \Phi_m^{00} \\ \vdots \\ \Phi_m^{01} \\ \vdots \\ \Phi_m^{10} \\ \vdots \\ \Phi_m^{11} \\ \vdots \\ dX_m \\ \vdots \\ dY_m \\ \vdots \\ \phi_n \\ \vdots \\ dx_n \\ \vdots \\ dy_n \\ \vdots \end{bmatrix} \qquad C_I^x = \begin{bmatrix} r^2x - r'^2x' \\ \vdots \\ x\delta_{me} - x'\delta_{me'} \\ \vdots \\ y\delta_{me} - y'\delta_{me'} \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ \delta_{me} - \delta_{me'} \\ \vdots \\ 0 \\ \vdots \\ y\delta_{nc} - y'\delta_{nc'} \\ \vdots \\ \delta_{nc} - \delta_{nc'} \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \qquad C_I^y = \begin{bmatrix} r^2y - r'^2y' \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ x\delta_{me} - x'\delta_{me'} \\ \vdots \\ y\delta_{me} - y'\delta_{me'} \\ \vdots \\ 0 \\ \vdots \\ \delta_{me} - \delta_{me'} \\ \vdots \\ -x\delta_{nc} + x'\delta_{nc'} \\ \vdots \\ 0 \\ \vdots \\ \delta_{nc} - \delta_{nc'} \\ \vdots \end{bmatrix} \tag{9}$$

and we now have

$$\chi^2 = \sum(\Delta x + C_I^x V_I)^2 + (\Delta y + C_I^y V_I)^2 \tag{10}$$

where $\Delta x \equiv x_{ce} - x_{c'e'}$ and so minimising wrt the variable V_I yields the set of linear equations:

$$A_{IJ} V_J = B_J \tag{11}$$

where

$$A_{IJ} = \sum C_I^x C_J^x + C_I^y C_J^y \qquad\qquad B_J = -\sum C_J^x \Delta x + C_J^y \Delta y \tag{12}$$

1.3 Databases and software

The parameters for the nominal chip size, margin etc are kept in `nominal.db`. (All these databases are in perl format as they get read by perl scripts).

Chip names, integer positions and orientation of chips are stored as associative arrays in `chips.db`. The key is an integer which runs from $0 \ldots N_c - 1$. Orientation is 1 for lower row of chips, -1 for upper.

Field names (for a1413 run these were exposure numbers '031', '032' etc.) are stored as associative array in `fields.db`.

Assume that one has generated `basedir/chip$chipname/$fieldname.stars` cats of moderately bright stars (I used `select -m 16 19 -rx 0.8 1.2` to select them) then you first want to run `superlists.pl` which generates a set of N_e files (1 per exposure) containing nominal detector coords x, y; chip number c, and exposure number e.

The next step is to run `makemergelist.pl` which figures out an approximate rotation and translation to register pairs of superlists (we use all pairings of exposures) and generate a file `mergelists.out` containing $x, y, c, e, x', y', c', e'$.

We now feed `mergelists.out` to `mosaicfit -c 7 -e 11` which solves for α; Φ_e, dX_e, dY_e; ϕ_c, dx_c, dy_c and redirect output into `mosaicfit.par`.

We now run `makemergelist2.pl` which uses 1st approximation to paramaters in `mosaicfit.par` make a refined merged list: `mergelists2.out` which we feed back to `mosaicfit -c 7 -e 11` (perhaps after filtering on column 9 which contains residual displacement) to give a refined set of transformation parameters.

For the 11 exposures and 7 chips for a1413 data the result was

```
-1.92821e-10
           0            0            0
-6.01328e-05       296.92      10.5084
   0.0004332      271.737       1202.7
 0.000364594     -262.278      1218.76
 0.000282352     -380.615      1090.64
 0.000300921      1105.92      1159.14
 0.000401059      1265.68      1321.57
  0.00897166     -748.315      1151.47
  0.00893781       -161.3      1188.75
  0.00881633     -147.714       897.66
```

0.00940282	731.594	935.408
0	0	0
0.000571406	9.25005	8.26747
0.00255388	9.46212	7.78085
0.00169522	2.99181	10.856
0.00245214	6.8493	0.62552
-0.000119338	-5.15558	-12.0497
0.000498863	-1.47967	-10.6429

The first line contains distortion coefficient α.

The next 11 lines contain Φ_e, dX_e, dY_e (zero for 1st exposure).

The next 7 lines contain ϕ_c, dx_c, dy_c (zero for first chip).

Now we can generate a stack of images for a chosen section of X, Y space using `makestack.pl` which figures out which chips lie under the chosen section for each exposure and warps them using `mosaicmap`.

The final step is to combine the images (median, avsigclip or whatever) with `pastiche`.