CXC-JCM-004

CXC Coordinates

ACIS CC Mode

Jonathan McDowell
Chandra X-ray Center
October 29, 2001

1 Introduction

This document describes the algorithm for calculating sky coordinates for ACIS CC mode.

In CC mode, the rows of the chip are continuously clocked. This means that we know the arrival time of the photon well, but we don't know the CHIPY of the photon. CC mode is used when there is a bright source of known celestial position in the field.

We want to do the best we can in assigning sky coordinates to each photon, even though it's not really meaningful for photons not from the target. With only one dimension known, we have to guess the other. The choice we make is to lay the photons in a line which goes through the position of the target, so that target photons have the correct sky coordinates.

If we want the photons from the source to all have the same sky coordinates, we must assign them a CHIPY which varies with time to match the spacecraft dither. What about photons far from the source? If we want the photons on other chips in an ACIS-S to lie on the same line (mostly for elegance, rather than science, since the positions are meaningless at that point - unless the target is dithered across a chip boundary when it does matter), we must make sure that the same CHIPY is selected even for other chips. This leads to the algorithm described below.

2 The algorithm

First we describe the actual algorithm. Then we describe a simplified calculation which ignores projection effects and works only for the target chip for the case when CHIP and DET are parallel (S3 chip).

2.1 Quantities of interest

To go from DET to CEL (RA and Dec) coordinates we apply

$$\begin{pmatrix} \mathrm{RA} \\ \mathrm{DEC} \end{pmatrix} = \mathrm{P(A)} \begin{pmatrix} x \\ y \end{pmatrix}$$

where P(A) is the tangent projection mapping, and the aspect A is defined by a nominal position (α_0, δ_0), a corresponding reference position (x_0, y_0), a scale Δ, and a roll angle θ.

SKY to CEL is defined by the same equation with zero roll angle (since SKY coordinates are aligned with RA and DEC).

We will consider three different aspects:

- Sky aspect A_S, with the nominal pointing direction and zero roll angle - this determines the SKY-CEL mapping at all times. There is no time dependence in the mapping of SKY to CEL. We have two sets of reference celestial coordinates: (RA_NOM,DEC_NOM) with sky pixels x_N, y_N and (RA_TARG,DEC_TARG) with sky pixels x_T, y_T.
- Nominal aspect A_N, with the nominal pointing direction and the nominal roll angle - this determines a reference DET-CEL mapping which may not correspond to the actual DET-CEL mapping for any time. The corresponding CEL and SKY coordinates of the optical axis are (RA_NOM, DEC_NOM) and (x_N, y_N). The DET coordinates of the axis are always (dx_0, dy_0).
- Instantaneous aspect, A_t, with the pointing direction and roll angle for a given time, the true DET-CEL mapping at a given time. The corresponding CEL and SKY coordinates of the optical axis are (RA_0(t),DEC_0(t)) and $(x_0(t), y_0(t))$.

There is also a small time dependence (SIM fiducial light boresight correction) in the mapping from DET to CHIP, which can be characterized by an offset $(UX(t), UY(t))$. To initialize the algorithm, we take the time of the first photon t_0. The mapping from CHIP to DET is represented by the operator L_j for chip j:

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = L_j(UX, UY) \begin{pmatrix} cx \\ cy \end{pmatrix}$$

2.2 The algorithm

We take the CEL coordinates RA_TARG, DEC_TARG and calculate the corresponding DET coordinates for the target at Nominal Aspect:

$$\begin{pmatrix} dx_T(N) \\ dy_T(N) \end{pmatrix} = P^{-1}(A_N) \begin{pmatrix} \mathrm{RA\_TARG} \\ \mathrm{DEC\_TARG} \end{pmatrix}$$

which can also be expressed in terms of the equivalent sky coordinates x_T, y_T:

$$\begin{pmatrix} dx_T(N) \\ dy_T(N) \end{pmatrix} = P^{-1}(A_N) P(A_S) \begin{pmatrix} x_T \\ y_T \end{pmatrix}$$

This calculation is implemented with routines dmTanPixToWorld and dmTanWorldToPix and gives us the **target detector coordinates**.

We then calculate the **target chip coordinates** (chip coords of the target at nominal aspect), using

$$\begin{pmatrix} cx_T(N) \\ cy_T(N) \end{pmatrix} = L_i^{-1}(UX(t_0), UY(t_0)) \begin{pmatrix} dx_T(N) \\ dy_T(N) \end{pmatrix}$$

(implemented by the pix_fpc_to_chip routine). Here i is the target chip (strictly, the target chip for this choice of aspect, since the target may be dithered across a chip edge). The value of i is found at this stage, rather than assumed beforehand, and saved. The target chip coordinates are not saved, as the variation in the fiducial correction must be applied later. The above calculation is done once for a given observation. Both cx and cy coordinates are used in the calculation that follows for each photon.

We now consider a photon at time t from chip j. We calculate the DET coordinates of the target chip coordinates on chip j at time t, using the fiducial offset values for that time. To be clear, let me be specific: suppose that the target fell on chip 7 at (256,712) in our previous calculation, but our new photon is from chip 6. We're going to use (256,712) as our starting chip coordinates on chip 6 as well, even though that position has no particular meaning on chip 6, so that we mimic the same dither pattern on each chip. We have effectively replicated the target on each chip - we'll call this the 'chip j target'. The way we do this, taking into account the time-dependent fiducial light correction, is to work with the target detector coordinates. Then

$$\begin{pmatrix} cx_{Ti}(N,t) \\ cy_{Ti}(N,t) \end{pmatrix} = L_i^{-1}(UX(t), UY(t)) \begin{pmatrix} dx_T(N) \\ dy_T(N) \end{pmatrix}$$

- the target chip coordinates using the fid correction at time t. Now we replicate, for chip j:

$$\begin{pmatrix} cx_{Tj}(N,t) \\ cy_{Tj}(N,t) \end{pmatrix} = \begin{pmatrix} cx_{Ti}(N,t) \\ cy_{Ti}(N,t) \end{pmatrix}$$

and calculate the corresponding detector coordinates -

$$\begin{pmatrix} dx_{Tj}(N,t) \\ dy_{Tj}(N,t) \end{pmatrix} = L_j(UX(t), UY(t)) \begin{pmatrix} cx_{Tj}(N,t) \\ cy_{Tj}(N,t) \end{pmatrix}$$

these are the DET coords of the chip j target with nominal aspect but fiducal offset for time t.

We next find the SKY coords of the chip j target at **nominal** aspect:

$$\begin{pmatrix} x_{Tj}(N,t) \\ y_{Tj}(N,t) \end{pmatrix} = P^{-1}(A_S)P(A_N) \begin{pmatrix} dx_{Tj}(N,t) \\ dy_{Tj}(N,t) \end{pmatrix}$$

In this equation, the time dependence of the result is only through the fiducial correction. If the change in fiducial correction is negligible and j is the target chip, we have gone round in a circle and recalculated (x_T, y_T) again! This procedure has given us a fake target sky position which is adjusted for the SIM motion and which is replicated on each chip. We are now ready to take out the dither, by applying the instantaneous aspect (for the time of the photon we are working with)

$$\begin{pmatrix} dx_{Tj}(t) \\ dy_{Tj}(t) \end{pmatrix} = P^{-1}(A_t)P(A_S) \begin{pmatrix} x_{Tj}(N,t) \\ y_{Tj}(N,t) \end{pmatrix}$$

And next, back to chip coordinates again:

$$\begin{pmatrix} cx_T(t) \\ cy_T(t) \end{pmatrix} = L_j^{-1}(UX(t), UY(t)) \begin{pmatrix} dx_{Tj}(t) \\ dy_{Tj}(t) \end{pmatrix}$$

Now we have the target j chip position corrected for dither.

So far we have not used any actual data from ACIS except the photon arrival time and which chip it fell on. Now we take the CHIPY for the target j chip position and combine it with the CHIPX for the actual photon, $cx(t)$ (which is the only positional info we have for it). Then we forward calculate using this bogus position,

$$\begin{pmatrix} dx(t) \\ dy(t) \end{pmatrix} = L_j(UX(t), UY(t)) \begin{pmatrix} cx(t) \\ cy_T(t) \end{pmatrix}$$

and

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = P^{-1}(A_S)P(A_t) \begin{pmatrix} dx(t) \\ dy(t) \end{pmatrix}$$

This is, in fact, our final answer. To get it we have gone from SKY to CHIP and back to SKY, and then again from SKY to CHIP and back to SKY, since different effects must be applied with different choices of aspect and fid correction. This is illustrated in the figure.

2.3 Marginally simpler treatment

We can understand the algorithm better by looking at the simple case of a photon and the target position both on CCD 7 (S3), and neglecting small corrections due to projection effects and time variations in the roll. In this case, the operator L is simply an additive constant

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} UX \\ UY \end{pmatrix} + \begin{pmatrix} cx \\ cy \end{pmatrix}$$

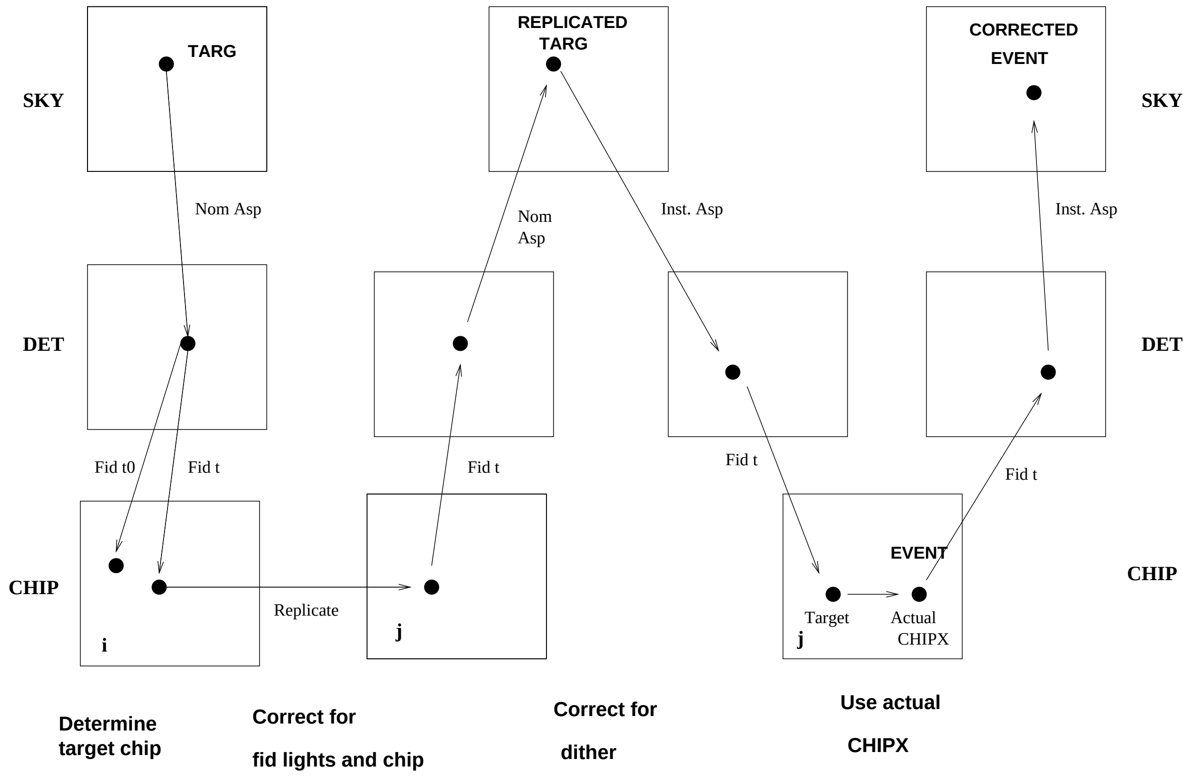

Figure 1: Illustration of the transformations involved in the CC coordinate algorithm.

and the operator $P^{-1}(A_S)P(A_t)$ is

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_0(t) \\ y_0(t) \end{pmatrix} + R \begin{pmatrix} dx - dx_0 \\ dy - dy_0 \end{pmatrix}$$

where

$$R = \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

is a rotation matrix where c and s are the cosine and sine of an angle which is related to the roll angle. Then

$$\begin{pmatrix} dx_T(N) \\ dy_T(N) \end{pmatrix} = \begin{pmatrix} dx_0 \\ dy_0 \end{pmatrix} + R^{-1} \begin{pmatrix} x_T - x_N \\ y_T - y_N \end{pmatrix}$$

and since $i = j$ the same follows for the next step:

$$\begin{pmatrix} dx_{Tj}(N,t) \\ dy_{Tj}(N,t) \end{pmatrix} = \begin{pmatrix} dx_0 \\ dy_0 \end{pmatrix} + R^{-1} \begin{pmatrix} x_T - x_N \\ y_T - y_N \end{pmatrix}$$

so the x_N values cancel out and we are left with the target sky position

$$\begin{pmatrix} x_{Tj}(N,t) \\ y_{Tj}(N,t) \end{pmatrix} = \begin{pmatrix} x_T \\ y_T \end{pmatrix}$$

Now

$$\begin{pmatrix} dx_{Tj}(t) \\ dy_{Tj}(t) \end{pmatrix} = \begin{pmatrix} dx_0 \\ dy_0 \end{pmatrix} + R^{-1} \begin{pmatrix} x_{Tj}(N,t) - x_0(t) \\ y_{Tj}(N,t) - y_0(t) \end{pmatrix}$$

so substituting in and applying the inverse L operator this reduces to

$$\begin{pmatrix} cx_T(t) \\ cy_T(t) \end{pmatrix} = \begin{pmatrix} dx_0 \\ dy_0 \end{pmatrix} + R^{-1} \begin{pmatrix} x_T - x_0(t) \\ y_T - y_0(t) \end{pmatrix} - \begin{pmatrix} UX(t) \\ UY(t) \end{pmatrix}$$

Now working with the actual CHIPX,

$$\begin{pmatrix} dx(t) \\ dy(t) \end{pmatrix} = \begin{pmatrix} cx(t) \\ dy_0 \end{pmatrix} + \begin{pmatrix} 0 \\ c(y_T - y_0(t)) - s(x_T - x_0(t)) \end{pmatrix} + \begin{pmatrix} UX(t) \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} c^2 x_0(t) + c(cx(t) - dx_0) - sc(y_T - y_0(t)) + s^2 x_T + c(UX(t)) \\ s^2 y_0(t) + s(cx(t) - dx_0) + c^2 y_T - cs(x_T - x_0(T)) + s(UX(t)) \end{pmatrix}$$

Let's see what sense we can make of this mess. Multiply the x coordinate by s (sin of roll) and the y coordinate by c and subtract, and we find lots of things cancel and leave

$$sx(t) - cy(t) = sx_T - cy_T$$

which is a line in sky coordinates passing through the target position.