MIT Kavli Institute **Chandra X-Ray Center**

MEMORANDUM
December 19, 2005

To:	Jonathan McDowell, SDS Group Leader
From:	Glenn E. Allen, SDS
Subject:	Adjusting ACIS Event Data to Compensate for CTI
Revision:	5.4
URL:	http://space.mit.edu/CXC/docs/docs.html#cti
File:	/nfs/cxc/h2/gea/sds/docs/memos/memo_cti_correction_5.4.tex

The ACIS instrument teams at PSU and MIT have shown that a significant improvement in the energy resolution of existing ACIS event data can be obtained by compensating for some of the effects of the parallel and serial charge-transfer inefficiencies (CTIs) of the CCDs. To achieve this improvement, charge is added to each 3 pixel $\times$ 3 pixel event island. While some pixels in the island gain charge, others may lose charge. Yet, the net change for an event is alway positive. The amount of charge added to each event is based on an estimate of the average amount of charge that is lost as charge packets are clocked across the charge traps on the ACIS detectors. The average amount of charge lost depends on the density of charge traps on the detector, the location of the event on the CCD, and the number of traps that have already been filled by "precursor events" in the same CCD column (i.e. on how many empty traps an event must cross to get to the read out). The algorithm described in section 1.4 is used to compute the CTI-adjusted values of PHAS. These adjusted values are contained in a new column called PHAS_ADJ. The original, unadjusted values are retained in the column PHAS. Note that only the column PHAS is written to the output file by default. The column PHAS_ADJ can be included if the parameter eventdef includes "f:phas_adj." Once the values of PHAS_ADJ are computed for an event, these values are used to compute the values of FLTGRADE, GRADE, PHA, ENERGY and PI for the event. At the present, the parallel CTI is calibrated for all but ACIS-S1 and the serial CTI is calibrated for only ACIS-S3. These calibration data are appropriate for observations with frame times of about 3.2 s.

1 Changes to acis_process_events

1.1 Additional Parameters

1. apply_cti,b,h,"yes","no","yes","Apply CTI adjustment?"
2. ctifile,s,h,"CALDB",,,"ACIS CTI ARD file (NONE — none — CALDB — <filename>)"
3. max_cti_iter,i,h,15,1,20,"Maximum number of iterations for the CTI adjustment of each event"
4. cti_converge,r,h,0.1,0.1,1.0,"The convergence criterion for each CTI-adjusted pixel in adu"

When Catherine Grant tested the PSU CTI-adjustment tool, she found that the median number of iterations required to satisfy a convergence criterion of 0.1 adu is four. No event required more than ten iterations. Therefore, a default maximum of fifteen iterations should be sufficient to determine the values of PHAS_ADJ. The default convergence criterion is 0.1 adu because this is the default value used for the PSU CTI-adjustment tool.

1.2 Additional Input

The tool acis_process_events must read the CTI ARD file in addition to the other files it already reads. The format of the CTI ARD file is summarized in section 2.

1.3 Additional Output

If apply_cti = "yes," the output event file has the same format as a Level 1 or Level 2 ACIS event file except that the file includes the the additional keywords "CTIFILE" and "CTI_CORR" and may also include the additional column "PHAS_ADJ." The keyword CTIFILE contains the name of the CTI ARD file used to process the data. The keyword CTI_CORR = T. If the parameter event includes, for example, "f:phas_adj," then the values of PHAS_ADJ are written to the output file. The column named PHAS contains the original, unadjusted values of the charge distributions of the event islands.

If apply_cti = "no," no CTI adjustment is applied, CTIFILE = "NONE," and CTI_CORR = F".

1.4 Processing

The tool should produce a warning if TIMEDEL is not in the range specified by the corresponding calibration-boundary ("CBD") keyword.

If apply_cti = "yes," then the algorithm described hereafter is used to compute the real-valued, CTI-adjusted values of a pulse-height event island PHAS_ADJ. The column PHAS contains the unadjusted pulse-height information. The values of FLTGRADE, GRADE, PHA, ENERGY, and PI are based on the values of PHAS_ADJ, not PHAS (see Tables 1 and 2). The column PHA has integer values whether the CTI-adjustment is applied or not.

By default, only the values of PHAS are written to the output file. This information is retained to ensure that users can rerun acis_process_events to recompute the pulse-height information using an updated CTI ARD file or to remove the CTI adjustments.

The columns PHAS_ADJ and PHAS are arrays. For TIMED FAINT-mode observations, the relative CHIPX and CHIPY coordinates associated with $\mathrm{PHAS\_ADJ}_{i,j}$ are distributed as shown in figure 1 with i and $j \in [1,3]$. For example, if a TIMED FAINT-mode event occurs at $(\mathrm{CHIPX}, \mathrm{CHIPY}) = (960, 571)$, then $\mathrm{PHAS\_ADJ}_{1,2}$ corresponds to the CCD coordinates (959,571) and $\mathrm{PHAS\_ADJ}_{2,3}$ corresponds to the CCD coordinates (960,572). The algorithm described below applies only to a 3 pixel × 3 pixel event island. For TIMED VFAINT-mode observations, i and $j \in [1,5]$ and the appropriate 3 pixel × 3 pixel region to use is the central nine pixels of the 5 pixel × 5 pixel event island (i.e. the region associated with $\mathrm{PHAS}_{i,j}$, where i and $j \in [2,4]$ instead of i and $j \in [1,5]$). The outer sixteen pixels of the 5×5 island are not modified by the CTI-adjustment algorithm. The values of PHAS_ADJ are identically the same as the values of PHAS for these sixteen pixels.

If apply_cti = "yes," the following steps describe how to apply the CTI adjustment. For an event detected on CCD_ID = n at a location $(\mathrm{CHIPX}, \mathrm{CHIPY}) = (x_{\mathrm{CCD}}, y_{\mathrm{CCD}})$[1],

1. Copy the contents of PHAS to PHAS_ADJ.
2. Set the arrays DIFF_X = 0 and DIFF_Y = 0.
3. Copy the values of DIFF_X to DIFF_X_PREV and the values of DIFF_Y to DIFF_Y_PREV.
4. Copy the values of PHAS_ADJ (not PHAS) to PHAS_TMP.

[1] If the values of x_{CCD} and y_{CCD} are real numbers instead of integers, they should be rounded to the nearest integer before proceeding.

5. For the column of a 3 pixel $\times$ 3 pixel event island that is closest to the serial read-out node, compute the effects of serial CTI. Steps 5 and 6 should be performed only if the appropriate serial-CTI trap-density map exists for CCD_ID $= n$. The CTI ARD may not contain some serial-CTI trap-density maps if the effects of serial CTI are not fully calibrated.

   i. For NODE_ID $= 0$, $i = 1$ and $j \in [1, 3]$. If $(\mathrm{PHAS}_{1,j} + \mathrm{DIFF\_X\_PREV}_{1,j} + \mathrm{DIFF\_Y\_PREV}_{1,j}) \geq$ the split threshold, then
$$\mathrm{DIFF\_X}_{1,j} = \mathrm{DELTPHAX}_{1,j}. \quad (1)$$
   ii. For NODE_ID $= 1$, $i = 3$ and $j \in [1, 3]$. If $(\mathrm{PHAS}_{3,j} + \mathrm{DIFF\_X\_PREV}_{3,j} + \mathrm{DIFF\_Y\_PREV}_{3,j}) \geq$ the split threshold, then
$$\mathrm{DIFF\_X}_{3,j} = \mathrm{DELTPHAX}_{3,j}. \quad (2)$$
   iii. For NODE_ID $= 2$, $i = 1$ and $j \in [1, 3]$. If $(\mathrm{PHAS}_{1,j} + \mathrm{DIFF\_X\_PREV}_{1,j} + \mathrm{DIFF\_Y\_PREV}_{1,j}) \geq$ the split threshold, then
$$\mathrm{DIFF\_X}_{1,j} = \mathrm{DELTPHAX}_{1,j}. \quad (3)$$
   iv. For NODE_ID $= 3$, $i = 3$ and $j \in [1, 3]$. If $(\mathrm{PHAS}_{3,j} + \mathrm{DIFF\_X\_PREV}_{3,j} + \mathrm{DIFF\_Y\_PREV}_{3,j}) \geq$ the split threshold, then
$$\mathrm{DIFF\_X}_{3,j} = \mathrm{DELTPHAX}_{3,j}. \quad (4)$$

   The quantity $\mathrm{DIFF\_X}_{i,j}$ is an estimate of the amount of charge that should be added to pixel (i, j). (DIFF_X_PREV is the same thing except from the previous iteration.) The quantity $\mathrm{DELTPHAX}_{i,j}$ represents the amount of charge lost from pixel (i, j) due to the effects of serial CTI and is a function of the CCD used, the location of an event on the CCD, and the charge deposited on the CCD.

   The value of $\mathrm{DELTPHAX}_{i,j}$ used in equations 1, 2, 3 and 4 is computed as follows.

   i. Find the row m in HDU 1 of the appropriate CTI ARD file that satisfies the conditions
$$\begin{aligned} \mathrm{CCD\_ID}_m &= n, & (5)\\ \mathrm{CHIPX\_LO}_m \leq x_{\mathrm{CCD}} &\leq \mathrm{CHIPX\_HI}_m, \text{ and} & (6)\\ \mathrm{CHIPY\_LO}_m \leq y_{\mathrm{CCD}} &\leq \mathrm{CHIPY\_HI}_m, & (7) \end{aligned}$$
   where CCD_ID, CHIPX_LO, CHIPX_HI, CHIPY_LO, and CHIPY_HI are the names of columns in the CTI ARD file (see sec. 2). Note that the values of x_{CCD} and y_{CCD} should be rounded to the nearest integer if they are not integers.

   ii. For row m, find the two non-zero (real!) values PHA_k and PHA_{k+1} such that
$$0 < \mathrm{PHA}_k \leq (\mathrm{PHAS}_{i,j} + \mathrm{DIFF\_X\_PREV}_{i,j} + \mathrm{DIFF\_Y\_PREV}_{i,j}) < \mathrm{PHA}_{k+1}, \quad (8)$$
   where PHA_k and PHA_{k+1} are elements of the column PHA in the CTI ARD file.

   iii. Compute the effective "VOLUME" occupied by the charge in pixel (i, j) using the linear interpolation
$$\begin{aligned} \mathrm{VOLUME}_{i,j} = & \left(\frac{(\mathrm{PHAS}_{i,j} + \mathrm{DIFF\_X\_PREV}_{i,j} + \mathrm{DIFF\_Y\_PREV}_{i,j}) - \mathrm{PHA}_k}{\mathrm{PHA}_{k+1} - \mathrm{PHA}_k} \right) \times \\ & (\mathrm{VOLUME\_X}_{k+1} - \mathrm{VOLUME\_X}_k) + \mathrm{VOLUME\_X}_k, \end{aligned} \quad (9)$$
   where $\mathrm{VOLUME\_X}_k$ and $\mathrm{VOLUME\_X}_{k+1}$ are the k^{th} and $(k+1)^{\mathrm{th}}$ elements of the column VOLUME_X in the CTI ARD file (see sec. 2). This formula is valid if and only if $1 \leq k <$ NPOINTS (i.e. if $\mathrm{PHA}_1 \leq (\mathrm{PHAS}_{i,j} + \mathrm{DIFF\_X\_PREV}_{i,j} + \mathrm{DIFF\_Y\_PREV}_{i,j}) < \mathrm{PHA}_{\mathrm{NPOINTS}}$, where PHA_1 and $\mathrm{PHA}_{\mathrm{NPOINTS}}$ are the smallest and largest values of the vector PHA for row m, respectively). If $0 < (\mathrm{PHAS}_{i,j} + \mathrm{DIFF\_X\_PREV}_{i,j} + \mathrm{DIFF\_Y\_PREV}_{i,j}) < \mathrm{PHA}_1$, then use the linear extrapolation
$$\begin{aligned} \mathrm{VOLUME}_{i,j} = & \left(\frac{(\mathrm{PHAS}_{i,j} + \mathrm{DIFF\_X\_PREV}_{i,j} + \mathrm{DIFF\_Y\_PREV}_{i,j}) - \mathrm{PHA}_1}{\mathrm{PHA}_2 - \mathrm{PHA}_1} \right) \times \\ & (\mathrm{VOLUME\_X}_2 - \mathrm{VOLUME\_X}_1) + \mathrm{VOLUME\_X}_1. \end{aligned} \quad (10)$$

If $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq \text{PHA}_{\text{NPOINTS}}$, then use the linear extrapolation

$$\begin{aligned}\text{VOLUME}_{i,j} \quad = \quad & \left(\frac{(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) - \text{PHA}_{\text{NPOINTS}-1}}{\text{PHA}_{\text{NPOINTS}} - \text{PHA}_{\text{NPOINTS}-1}} \right) \times \\ & (\text{VOLUME\_X}_{\text{NPOINTS}} - \text{VOLUME\_X}_{\text{NPOINTS}-1}) + \\ & \text{VOLUME\_X}_{\text{NPOINTS}-1}. \end{aligned} \tag{11}$$

If $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \leq 0$, then $\text{VOLUME}_{i,j} = 0$. However, this situation should not arise since an adjustment should not be performed if the pulse height is less than the split threshold.

iv. Find the HDU in the CTI ARD file that has the serial-CTI trap-density map for CCD_ID $= n$ (see sec. 2). Set $\text{TRAPDENS}_{i,j}$ equal to the value of the map at the position $(\text{CHIPX}, \text{CHIPY}) = (x_i, y_i)$, where $x_i = x_{\text{CCD}} - 1$, x_{CCD}, and $x_{\text{CCD}} + 1$ for $i = 1$, 2, and 3 and $y_i = y_{\text{CCD}} - 1$, y_{CCD}, and $y_{\text{CCD}} + 1$ for $j = 1$, 2, and 3.

v. Compute the value of $\text{DELTPHAX}_{i,j}$:

$$\text{DELTPHAX}_{i,j} \quad = \quad \text{TRAPDENS}_{i,j} \times \text{VOLUME}_{i,j}. \tag{12}$$

6. For the two columns of a 3×3 pixel event island that are farthest from the serial read-out node, compute the effects of serial CTI.

i. For NODE_ID $= 0$, $i \in [2, 3]$ and $j \in [1, 3]$. If $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq$ the split threshold and the split threshold $> (\text{PHAS}_{i-1,j} + \text{DIFF\_X\_PREV}_{i-1,j} + \text{DIFF\_Y\_PREV}_{i-1,j})$, then

$$\text{DIFF\_X}_{i,j} = \text{DELTPHAX}_{i,j}. \tag{13}$$

For NODE_ID $= 0$, $i \in [2, 3]$ and $j \in [1, 3]$. If $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq (\text{PHAS}_{i-1,j} + \text{DIFF\_X\_PREV}_{i-1,j} + \text{DIFF\_Y\_PREV}_{i-1,j})$ and $(\text{PHAS}_{i-1,j} + \text{DIFF\_X\_PREV}_{i-1,j} + \text{DIFF\_Y\_PREV}_{i-1,j}) \geq$ the split threshold, then

$$\text{DIFF\_X}_{i,j} = \text{DELTPHAX}_{i,j} - \text{DELTPHAX}_{i-1,j}. \tag{14}$$

For NODE_ID $= 0$, $i \in [2, 3]$ and $j \in [1, 3]$. If $(\text{PHAS}_{i-1,j} + \text{DIFF\_X\_PREV}_{i-1,j} + \text{DIFF\_Y\_PREV}_{i-1,j}) > (\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j})$ and $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq$ the split threshold, then

$$\text{DIFF\_X}_{i,j} = \text{FRCTRLX}n \times (\text{DELTPHAX}_{i,j} - \text{DELTPHAX}_{i-1,j}). \tag{15}$$

ii. For NODE_ID $= 1$, $i \in [1, 2]$ and $j \in [1, 3]$. If $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq$ the split threshold and the split threshold $> (\text{PHAS}_{i+1,j} + \text{DIFF\_X\_PREV}_{i+1,j} + \text{DIFF\_Y\_PREV}_{i+1,j})$, then

$$\text{DIFF\_X}_{i,j} = \text{DELTPHAX}_{i,j}. \tag{16}$$

For NODE_ID $= 1$, $i \in [1, 2]$ and $j \in [1, 3]$. If $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq (\text{PHAS}_{i+1,j} + \text{DIFF\_X\_PREV}_{i+1,j} + \text{DIFF\_Y\_PREV}_{i+1,j})$ and $(\text{PHAS}_{i+1,j} + \text{DIFF\_X\_PREV}_{i+1,j} + \text{DIFF\_Y\_PREV}_{i+1,j}) \geq$ the split threshold, then

$$\text{DIFF\_X}_{i,j} = \text{DELTPHAX}_{i,j} - \text{DELTPHAX}_{i+1,j}. \tag{17}$$

For NODE_ID $= 1$, $i \in [1, 2]$ and $j \in [1, 3]$. If $(\text{PHAS}_{i+1,j} + \text{DIFF\_X\_PREV}_{i+1,j} + \text{DIFF\_Y\_PREV}_{i+1,j}) > (\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j})$ and $(\text{PHAS}_{i,j} + \text{DIFF\_X\_PREV}_{i,j} + \text{DIFF\_Y\_PREV}_{i,j}) \geq$ the split threshold, then

$$\text{DIFF\_X}_{i,j} = \text{FRCTRLX}n \times (\text{DELTPHAX}_{i,j} - \text{DELTPHAX}_{i+1,j}). \tag{18}$$

iii. For NODE_ID = 2, $i \in [2,3]$ and $j \in [1,3]$. If (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) $\geq$ the split threshold and the split threshold > (PHAS$_{i-1,j}$+DIFF_X_PREV$_{i-1,j}$+DIFF_Y_PREV$_{i-1,j}$), then

$$\mathrm{DIFF\_X}_{i,j} = \mathrm{DELTPHAX}_{i,j}. \tag{19}$$

For NODE_ID = 2, $i \in [2,3]$ and $j \in [1,3]$. If (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) $\geq$ (PHAS$_{i-1,j}$+DIFF_X_PREV$_{i-1,j}$+DIFF_Y_PREV$_{i-1,j}$) and (PHAS$_{i-1,j}$+DIFF_X_PREV$_{i-1,j}$+ DIFF_Y_PREV$_{i-1,j}$) $\geq$ the split threshold, then

$$\mathrm{DIFF\_X}_{i,j} = \mathrm{DELTPHAX}_{i,j} - \mathrm{DELTPHAX}_{i-1,j}. \tag{20}$$

For NODE_ID = 2, $i \in [2,3]$ and $j \in [1,3]$. If (PHAS$_{i-1,j}$+DIFF_X_PREV$_{i-1,j}$+DIFF_Y_PREV$_{i-1,j}$) > (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) and (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) $\geq$ the split threshold, then

$$\mathrm{DIFF\_X}_{i,j} = \mathrm{FRCTRLX}n \times \left(\mathrm{DELTPHAX}_{i,j} - \mathrm{DELTPHAX}_{i-1,j}\right). \tag{21}$$

iv. For NODE_ID = 3, $i \in [1,2]$ and $j \in [1,3]$. If (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) $\geq$ the split threshold and the split threshold > (PHAS$_{i+1,j}$+DIFF_X_PREV$_{i+1,j}$+DIFF_Y_PREV$_{i+1,j}$), then

$$\mathrm{DIFF\_X}_{i,j} = \mathrm{DELTPHAX}_{i,j}. \tag{22}$$

For NODE_ID = 3, $i \in [1,2]$ and $j \in [1,3]$. If (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) $\geq$ (PHAS$_{i+1,j}$+DIFF_X_PREV$_{i+1,j}$+DIFF_Y_PREV$_{i+1,j}$) and (PHAS$_{i+1,j}$+DIFF_X_PREV$_{i+1,j}$+ DIFF_Y_PREV$_{i+1,j}$) $\geq$ the split threshold, then

$$\mathrm{DIFF\_X}_{i,j} = \mathrm{DELTPHAX}_{i,j} - \mathrm{DELTPHAX}_{i+1,j}. \tag{23}$$

For NODE_ID = 3, $i \in [1,2]$ and $j \in [1,3]$. If (PHAS$_{i+1,j}$+DIFF_X_PREV$_{i+1,j}$+DIFF_Y_PREV$_{i+1,j}$) > (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) and (PHAS$_{i,j}$+DIFF_X_PREV$_{i,j}$+DIFF_Y_PREV$_{i,j}$) $\geq$ the split threshold, then

$$\mathrm{DIFF\_X}_{i,j} = \mathrm{FRCTRLX}n \times \left(\mathrm{DELTPHAX}_{i,j} - \mathrm{DELTPHAX}_{i+1,j}\right). \tag{24}$$

where FRCTRLXn is the name of a keyword in the CTI ARD file (see sec. 2) and represents the fraction of the trapped charge that is released in the first pixel following the pixel which lost the charge.

7. For the bottom row of the 3 pixel × 3 pixel event island, compute the effects of parallel CTI. Steps 7 and 8 should be performed only if the appropriate parallel-CTI trap-density map exists for CCD_ID = n. The CTI ARD may not contain some parallel-CTI trap-density maps if the effects of parallel CTI are not fully calibrated. The bottom row of PHAS$_{i,j}$ is given by $i \in [1,3], j = 1$ (fig. 1). If (PHAS$_{i,1}$ + DIFF_X$_{i,1}$ + DIFF_Y_PREV$_{i,1}$) $\geq$ the split threshold, then

$$\mathrm{DIFF\_Y}_{i,1} = \mathrm{DELTPHAY}_{i,1}. \tag{25}$$

The quantity DIFF_Y$_{i,j}$ is an estimate of the amount of charge that should be added to pixel (i,j) of the event island. The quantity DELTPHAY$_{i,j}$ represents the amount of charge lost from pixel (i,j) due to parallel CTI and is a function of the CCD used, the location of an event on the CCD, and the charge deposited on the CCD. This quantity is computed using the same linear interpolation (and extrapolation) method used to compute DELTPHAX$_{i,j}$, where (PHAS$_{i,j}$ + DIFF_X_PREV$_{i,j}$ + DIFF_Y_PREV$_{i,j}$) is replaced by (PHAS$_{i,j}$ + DIFF_X$_{i,j}$ + DIFF_Y_PREV$_{i,j}$), VOLUME_X is replaced with VOLUME_Y and the serial-CTI trap-density map for CCD_ID = n is replaced with the parallel-CTI trap-density map for the CCD.

8. For the middle and top rows of the 3 pixel × 3 pixel event island, compute the effects of parallel CTI. The middle and top rows of PHAS$_{i,j}$ are given by $i \in [1,3]$, $j \in [2,3]$ (fig. 1). If (PHAS$_{i,j}$+DIFF_X$_{i,j}$+

DIFF_Y_PREV$_{i,j}$) $\geq$ the split threshold and the split threshold > (PHAS$_{i,j-1}$+DIFF_X$_{i,j-1}$+DIFF_Y_PREV$_{i,j-1}$), then

$$\text{DIFF\_Y}_{i,j} = \text{DELTPHAY}_{i,j}. \tag{26}$$

If (PHAS$_{i,j}$ + DIFF_X$_{i,j}$ + DIFF_Y_PREV$_{i,j}$) $\geq$ (PHAS$_{i,j-1}$ + DIFF_X$_{i,j-1}$ + DIFF_Y_PREV$_{i,j-1}$) and (PHAS$_{i,j-1}$ + DIFF_X$_{i,j-1}$ + DIFF_Y_PREV$_{i,j-1}$) $\geq$ the split threshold, then

$$\text{DIFF\_Y}_{i,j} = \text{DELTPHAY}_{i,j} - \text{DELTPHAY}_{i,j-1}. \tag{27}$$

If (PHAS$_{i,j-1}$ + DIFF_X$_{i,j-1}$ + DIFF_Y_PREV$_{i,j-1}$) > (PHAS$_{i,j}$ + DIFF_X$_{i,j}$ + DIFF_Y_PREV$_{i,j}$) and (PHAS$_{i,j}$ + DIFF_X$_{i,j}$ + DIFF_Y_PREV$_{i,j}$) $\geq$ the split threshold, then

$$\text{DIFF\_Y}_{i,j} = \text{FRCTRLY}n \times (\text{DELTPHAY}_{i,j} - \text{DELTPHAY}_{i,j-1}). \tag{28}$$

where FRCTRLYn is the name of a keyword in the CTI ARD file (see sec. 2) and represents the fraction of the trapped charge that is released in the first pixel following the pixel which lost the charge.

9. Use the estimate of the effects of serial and parallel CTI to adjust the 3 pixel $\times$ 3 pixel event island PHAS. For i and $j \in [1, 3]$,

$$\text{PHAS\_ADJ}_{i,j} = \text{PHAS}_{i,j} + \text{DIFF\_X}_{i,j} + \text{DIFF\_Y}_{i,j}. \tag{29}$$

   Pixels whose amount of charge < the split threshold are not modified.

10. Repeat steps 3–9 until the absolute values of PHAS_ADJ$_{i,j}$ − PHAS_TMP$_{i,j}$ are less than the value of the parameter cti_converge for all pixels in the 3 pixel $\times$ 3 pixel event island (i.e. until the changes in each of the values of PHAS_ADJ$_{i,j}$ from one iteration to the next are less than the value of the parameter cti_converge). The maximum number of times that steps 3–9 should be performed is specified by the parameter max_cti_iter. If the CTI-adjustment process does not converge after max_cti_iter iterations, set the values of PHAS_ADJ to be the values obtained during the last iteration and set STATUS bit 20 (of bits 0–31) equal to one.

11. Based on the input conditions shown in Table 1, compute the values of PHA, ENERGY, PI, FLTGRADE, GRADE, etc. as shown in Table 2.

12. Write out the results.

2 New CTI ARD

Since the effects of CTI are temperature dependent, a different CTI ARD file is required for each of the different focal-plane temperatures. The CTI ARD files have the following structure.

2.1 HDU 1 Keywords

- FRCTRLYn
- FRCTRLXn

The keywords FRCTRLYn and FRCTRLXn represent the fraction of the trapped charge that is released in the first pixel following the pixel which lost the charge.

Table 1. Input Conditions

Case	Parameter apply_cti[a]	Parameter doevtgrade	Parameter calculate_pi	Keyword CTI_CORR
1[b]	yes	yes	yes	F
2	yes	yes	yes	T
3	yes	yes	no	F
4	yes	yes	no	T
5	yes	no	yes	F
6	yes	no	yes	T
7	yes	no	no	F
8	yes	no	no	T
9	no	yes	yes	F
10	no	yes	yes	T
11	no	yes	no	F
12	no	yes	no	T
13	no	no	yes	F
14	no	no	yes	T
15	no	no	no	F
16	no	no	no	T

[a] If DATAMODE = GRADED, GRADED_HISTO, CC_GRADED, or CC33_GRADED, then follow the instructions for apply_cti=no because it is not possible to perform a CTI adjustment.
[b] This case is appropriate for standard pipeline processing.

2.2 HDU 1 Columns

- CCD_ID
- CHIPX_LO
- CHIPX_HI
- CHIPY_LO
- CHIPY_HI
- NPOINTS
- PHA (a vector with NPOINTS elements)
- VOLUME_X (a vector with NPOINTS elements)
- VOLUME_Y (a vector with NPOINTS elements)

The columns CCD_ID, CHIPX_LO, CHIPX_HI, CHIPY_LO, and CHIPY_HI are used to define a complete set of spatially-separate regions on the ACIS CCDs. Each row of HDU 1 corresponds to one region of an ACIS CCD and includes the vectors PHA, VOLUME_X, and VOLUME_Y. Each vector has NPOINTS elements. The vectors VOLUME_X and VOLUME_Y are used for serial and parallel CTI, respectively.

2.3 HDUs ≥ 2

Each one of these extensions (up to a maximum of twenty) contains either a parallel or serial trap-density map for one of the ten ACIS CCDs. The values in the trap-density maps represent the number of traps integrated along the read-out direction. For example, the value in the parallel trap-density map for ACIS-I3 at location (i, j) is the mean number of traps across which an event at $(\mathrm{CHIPX}, \mathrm{CHIPY}) = (i, j)$ is clocked

Table 2. Output[a]

Case	Column PHA[b]	Column ENERGY	Column PI	Column FLTGRADE[b]	Column GRADE	STATUS bit 20[c]	Keyword CTI_CORR	Keyword CTIFILE	Keyword GAINFILE
1[d]	C[e]	C	C	C	C	set	T	⟨cti⟩[f]	⟨gain_cti⟩[g]
2	C	C	C	C	C	set	T	⟨cti⟩	⟨gain_cti⟩
3	C	DNC[h]	DNC	C	C	set	T	⟨cti⟩	copy
4	C	DNC	DNC	C	C	set	T	⟨cti⟩	copy
5	DNC	C	C	DNC	DNC	unset	F	"NONE"	⟨gain⟩[i]
6	DNC	C	C	DNC	DNC	copy	T	copy	⟨gain_cti⟩
7	DNC	DNC	DNC	DNC	DNC	unset	F	"NONE"	copy
8	DNC	DNC	DNC	DNC	DNC	copy	T	copy	copy
9	C	C	C	C	C	unset	F	"NONE"	⟨gain⟩
10[j]	C	C	C	C	C	unset	F	"NONE"	⟨gain⟩
11	C	DNC	DNC	C	C	unset	F	"NONE"	copy
12	C	DNC	DNC	C	C	unset	F	"NONE"	copy
13	DNC	C	C	DNC	DNC	unset	F	"NONE"	⟨gain⟩
14	DNC	C	C	DNC	DNC	copy	T	copy	⟨gain_cti⟩
15	DNC	DNC	DNC	DNC	DNC	unset	F	"NONE"	copy
16	DNC	DNC	DNC	DNC	DNC	copy	T	copy	copy

[a] In all cases, the PHAS values in the output file are the unadjusted values of the pulse heights in the 3 pixel × 3 pixel (or 5 pixel × 5 pixel) event islands. The values of PHAS_ADJ are only (re)computed if apply_cti=yes.
[b] If apply_cti=yes, then the values of PHAS_ADJ are used to compute the values of PHA and FLTGRADE. If apply_cti=no, then the values of PHAS are used to compute the values of PHA and FLTGRADE. The values of PHA are integers in both cases.
[c] If apply_cti=yes, then STATUS bit 20 (of 0–31) is set to one for an event only if the PHAS adjustments for the event have not converged after performing the specified maximum number of trials. STATUS bit 20 should be (re)set to zero first before performing the CTI adjustment.
[d] This case is used for standard pipeline processing.
[e] (Re)computed.
[f] The name of the CTI ARD file used to perform the CTI adjustments.
[g] The name of the gain ARD file used to compute ENERGY from PHA. This file is appropriate for CTI-adjusted data.
[h] Copied, not (re)computed.
[i] The name of the gain ARD file used to compute ENERGY from PHA. This file is appropriate for data that has not had the CTI adjustment applied.
[j] The effects of the CTI adjustment are removed.

when it is read-out in the parallel direction. The scaled values of the integrated trap densities are stored as two-byte integers. The scaling parameters are specified in each trap-density extension using the keywords BZERO and BSCALE: the trap density = BZERO + BSCALE × the value in the trap-density image. The use of two-byte integers instead of four-byte real numbers helps reduce the size of the ARD file.

2.4 Size of File

The CTI ARD files are relatively large. Each row of HDU 1 has six two-byte integers and three four-byte real vectors with NPOINTS elements each. If the CTI ARD contains information for one region (the entire CCD) on each of the ten ACIS CCDs and if NPOINTS = 100, the binary table of HDU 1 comprises $10 \times 1 \times (6 \times 2 + 3 \times 4 \times 100)$ bytes = 12.1 kb of information. The size of each of the twenty trap-density maps is 2.1 Mb (i.e. 1024 × 1024 pixels × 2 bytes/pixel). Unless the number of regions per CCD becomes much larger than one, the size of HDU 1 is much smaller than the size of the trap-density maps and the size of one CTI ARD file can be as large as 42 Mb (i.e. 20 hdus × 2.1 Mb/hdu).

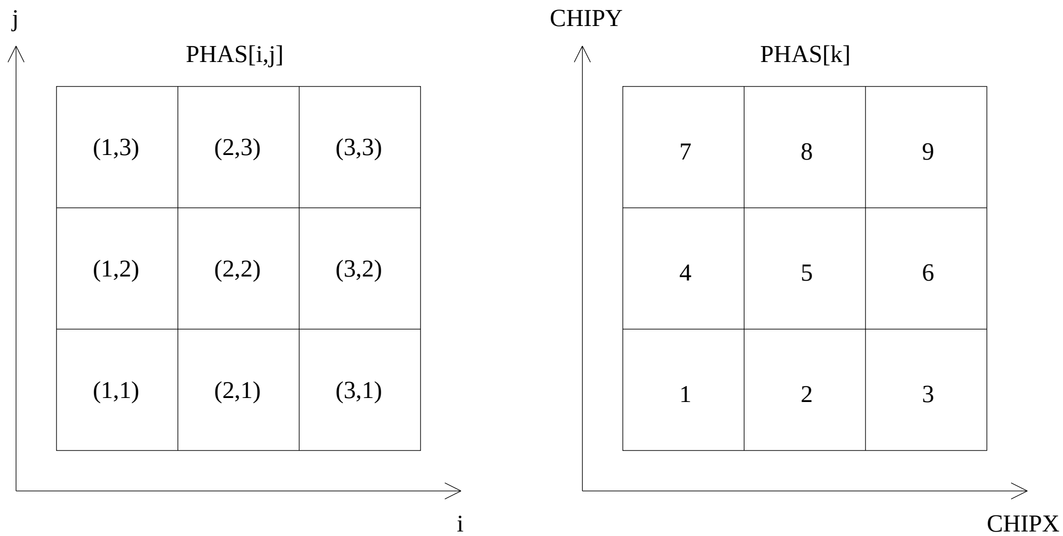

Figure 1: The relative CHIPX and CHIPY coordinates of the nine elements of a 3×3 ACIS event island (i.e. the nine elements of PHAS and PHAS_ADJ for TIMED FAINT-mode events).