Exposure Map Specifications

1 Generalities

Making an exposure map requires warping an image in detector coordinates to sky (or other coordinates) for each unique aim (r.a., dec., roll), scaling by the dwell time, and accumulating into an image. Such image transformation requires three fundamental operations:[1]

- coordinate transformation,
- image reconstruction (interpolation), and
- image resampling.

Coordinate transformation is the mapping from position in the input image (e.g., detector pixel) to the output position (e.g., tangent-plane pixel). It is generally a non-linear mapping. Also, it is usually not sufficient to map pixel center to pixel center, but quadrilateral boundaries to quadrilateral boundaries.

Image reconstruction (or interpolation) is the process of defining image values at non-integral pixel positions. This implies having some model for the image and how discretely sampled data might vary if spatial resolution were infinite. Some typical techniques are nearest-neighbor interpolation, bilinear interpolation, or cubic-sinc convolution (purported to be a theoretically optimal method).

Image resampling is the process of defining output bins, mapping them to the input bins, and interpolating the input into the output.

There are two directions one may take to image resampling. Forward resampling starts with the source image and maps data to the destination image. The entire destination image buffer is required for the forward transformation. The reverse method takes the destination grid, transforms it to the source grid, interpolates its quantities into the cells. The reverse method only needs one output cell at a time, not the whole image. In each approach, it is important to the fidelity of the result to map *pixel boundaries* from one space to the other, and not simply sample pixel centers (point-sampling).

Each approach has certain artifacts to be avoided. Undersampling can introduce aliasing, which are spatial frequency components absent from the source image. Interpolation can introduce trends in the output not present in the input, such as ringing, or "blockiness" of the image.

[1] A decent reference for fundamentals of image processing is Wolberg: "Digital Image Warping"

2 Prototype, overview, assumptions, simplifications

I have minimally prototyped the image transformations; the aim was to verify a transformation method. Further prototyping will apply other calibration data (vignetting, mirror area, different energies), and define data input and output formats. I have chosen the forward transformation direction, primarily because tools were available in IDL: the `poly_2d` function[2] performs image resampling, allowing rotation, oversampling (magnification), and interpolation by several methods. The prototype reads a CCD quantum efficiency uniformity map and an aspect history (dither pattern). It then transforms the image to a destination grid, and sums these over time.

The prototype *assumes* that the image is already in linear tangent-plane coordinates. This must be relaxed: initial transformation must go from detector space to tangent-plane via full AXAF coordinate system transformations.

I am *assuming* that translations of the aim (tangent point) by the dither amplitude does not appreciably affect the non-linearities in the grid; hence, the image may be dithered in the tangent-plane without undergoing detector-to-tangent-plane coordinate transformations at every time-step. This needs to be verified in detail for typical dither amplitudes of 20 arcsec. (This assumption precludes use of the same code in processing slew data.)

> Question on terminology: — does tangent-plane implicitly mean sky orientation and scale? I'm using it to mean detector side of mirror, but linear (sky is different scale, possibly, and different rotation).

At the completion of detector tangent-plane map computation, that map can be scaled and rotated to the sky tangent-plane. Working in the detector-side tangent plane allows images to be approximately the same size and orientation as the calibration maps.

> Note: grating maps require different or further processing to get to grating coordinates.

Heirarchical resolution will be necessary: QE maps come on coarse grids (32×32 pixels for ACIS; about 16x16 pixels for HRC). These can be processed with modest oversampling (5 to 10 times) relatively efficiently. Bad pixels and detector edges must be processed at high resolution — a single-pixel scale oversamped (by 5 or 10 times). Since these are small regions, they can be done fairly efficiently. The bad pixel maps also don't depend on energy (TBR; ACIS doc gives indication of some high/low energy dependence without specifying what constitutes high or low). After both low and high resolution regions have been processed into "maplets", they can be rebinned and translated to a common grid and merged.

My tests were done with nearest-neighbor interpolation, bilinear interpolation, or with optimal cubic sinc convolution. The latter, though theoretically optimal, introduces some edge affects: for a square hole in a uniform map, edges

[2] The IDL documentation is copied in Appendix **??**

are rounded and the amplitude "rings" around it. Nearest-neighbor seems to produce an adequate result if sufficiently oversampled (5-10 times magnification), and is also much faster. I suggest that:

1. Nearest-neighbor sampling be used on the low-resolution QE uniformity map, under the presumption that variations are discrete, and do not represent gradual changes on the detector single-pixel scale from one cell to the next. Tests on 1D arrays showed small differences between bilinear and cubic sinc interpolations (1%, except for edge effect in cubic sinc), but relatively large differences (20%) between bilinear and nearest-neighbor.

2. Nearest-neighbor sampling be used on the bad-pixel and edge regions, to preserve sharp boundaries, under the presumption that the discrete model is appropriate (on or off, no smooth transition).

3. Bilinear interpolation be available as an alternate method. There is a suggestion that ACIS QE non-uniformity becomes gradual at high energy, and a linear interpolation is more appropriate, since cell-to-cell variance represents spatially correlated variations.

3 Detailed Specifications

Key

I: Inputs

R: Parameters

O: Outputs [(*) indicates output to be archived]

S: Status

T: Tool name (page) [from ASC SDS Draft Tool List, 4/26/96]

X: Tool cross-reference from old list (if needed)

P: Prototype

D: Description

C: Comments

3.1 Imaging Mode Processing Steps

1. Create Tangent-Plane QE Uniformity Image

```
I: QE uniformity map, in detector-chip coordinates
   QE map uncertainty (map or scalar TBD)
   aspect solution
R: coordinate transformation params
   rebinning parameters
   interpolation parameters
   energy
O: QE uniformity image in tangent-plane coords.
   QE uncertainty image in tangent-plane coords.
   mean aspect (position and roll)
S: code: nil
   data: have one sample ACIS QE uniformity image at one energy;
         data-format needs definition.
T: exp_make_tp_qe_umap
P: pixlib does coord xforms;
   IDL has poly_2d for image warping, but need to define
    coefficiencts for transformation.
```

D: The input QE uniformity map is a binned image of detector relative sensitivity vs energy. There will be an image for each ACIS CCD, and for the HRC MCPs, each in detector pixel coordinates. Binning will be by approximately 32×32 (ACIS) or 16×16 (current HRC, TBR) detector pixels. Furthermore, there will be images at different energies.

Determine the mean aim-point and mean roll from the aspect solution.

For the mean aim-point and detector chip position in the focal plane, transform the QE map to a tangent-plane coordinate image. The image should be about 5 to 10 times oversampled (specified by rebinning parameter), with no rotation (rotation will be applied later, relative to the mean roll). Nearest-neighbor interpolation (a.k.a. block-replication) should be adequate (another possible standard interpolation method: bilinear).

Do the same for the uncertainty image.

> Sample data:ACIS QE uniformity:
> `dph/ASC/Calib/Data/w163c1_si_rqe_32x32.fits` (minimal header; some bad-values - truncate row 33, column 33);

2. Scale Image By Mirror Vignetting Function

```
I: tangent-plane QE uniformity image
   mirror vignetting data (including uncertainty)
   QE uncertainty image in tangent-plane coords.
   mean aim point
R: energy
   mirror shell n
   interpolation parameters
O: tangent-plane (QE uniformity) * (vignetting) for shell n at E.
   tangent-plane uncertainty image for shell n at E.
S: code: nil
   data: have low-energy-resolution table of area vs off-axis angle.
         Needs to be turned into vignetting only by dividing out
         on-axis area vs E.
T: exp_multiply_image
X:
P: nil
```

D: Resample vignetting image for specified mirror shell and energy to same grid as tangent-plane QE uniformity map, for specified aim-point. Mirror vignetting is a function of off-axis-angle (aim-point) and energy.

Multiply by tangent-plane QE uniformity map

Resample vignetting uncertainty image similarly.

Root-sum-square QE uncertainty tangent-plane image and vignetting tangent-plane uncertainty image to make combined uncertainty image (i.e., each cell, $_{ij}$, of uncertainty image) is:

$$\sigma_{ij} = \sqrt{\sigma^2_{QEij} + \sigma^2_{Vij}}.$$

(Note: propagating uncertainties in products requires use of relative standard deviations: σ_x/x.)

C: Vignetting data need to be specified. I suggest a FITS image stack vs E per shell (4 files, one for each shell, 1, 3, 4, and 6); n_energy extensions, binning TBD (but probably coarse). This may also be parameterized as a function (or a table), in which case the scaling would be determined by the energy and off-axis angle of each pixel in the QE uniformity map.

3. Form Aspect Histogram

```
I: aspect solution
R: binning parameters (alpha_bin, delta_bin, rho_bin)
   quality parameters (ranges)
O: aspect histogram(s) (3D)
S: code: nil
T: exp_make_asp_histogram
X:
P: nil
```

D: Select aspect data within quality ranges specified as $n\sigma$ on each of $\Delta\alpha$, $\Delta\delta$, $\Delta\rho$, and bin each of these regions into a 3D aspect histogram. Each roll-bin ($\Delta\rho$) plane of the result is an image whose value is the dwell time within that bin's $(\Delta\alpha, \Delta\delta)$.

If there are more than one quality range, then the result is a 3D histogram for each quality range.

C: Aspect histograms will be needed at different resolutions. For dither amplitude of 20 arcsec, and full aspect resolution of about 0.2 arcsec, the map will be 200 by 200 bins. Roll is predicted to have an amplitude of about 3 arcminutes (rms) in 48 hours. It may be binned more coarsly, since it amounts to about 6 ACIS or 40 HRC pixels on the -S arrays at maximum dispersion. Hence, 0.2 arcsec corresponds to about 15 roll bins for ACIS, or 25 roll bins for HRC.

4. Form Dead-time Aspect Histogram

```
I: aspect solution
   dead-time (per chip, if appropriate)
R: binning parameters (alpha_bin, delta_bin, rho_bin)
   quality parameters (ranges)
O: dead-time aspect histogram(s) (3D)
S: code: nil
T: exp_make_dead_asp_histogram
X:
P: nil
```

D: Select aspect data for times when detector dead-time was non-zero. Within these ranges, select aspect data within quality ranges specified as $n\sigma$ on each of $\Delta\alpha$, $\Delta\delta$, $\Delta\rho$. Bin each of these regions into a 3D dead-time-aspect histogram. Each roll-bin ($\Delta\rho$) plane of the result is an image whose value is the dead time within that bin's $(\Delta\alpha, \Delta\delta)$.

If there are more than one quality range, then the result is a 3D histogram for each quality range.

C: This product is to be subtracted from the dwell-time aspect histogram. It should be made at the same resolution as the former. This was not incorporated directly into the aspect-histogram because there may be several of these, such as for chip-dependent dead-times with ACIS.

Dead-time-aspect histograms will be needed at different spatial resolutions.

5. MAKE TANGENT-PLANE UNIFORMITY MAP

```
I: QE uniformity * vignetting image, for energy, E and shell, n.
   tangent-plane uncertainty image for shell n at E.
   aspect histogram
   dead-time-aspect histograms
R:
O: tangent-plane effective uniformity map
   tangent-plane effective uniformity quality map
S: IDL prototype
T: exp_convolve_images
X:
P:
```

D: Determine the size of the destination array from the dither amplitude and roll amplitude.

Subtract the dead-time-aspect histogram from the dwell-time aspect histogram.

Convolve uniformity*vignetting image by the dead-time corrected aspect histogram for each roll-bin. Sum resultant maps over roll-bins. The result is an effective uniformity exposure map, with units of time.

C: If done piecewise (i.e., sub-regions of detector, or sub-elements of detector array, such as a single ACIS-S chip), an effective axis of rotation must be computed, and a translation, to allow proper relative placement of "maplets."

Aspect-solution is a table comprised of:

$$t_i \quad \Delta\alpha_i \quad \Delta\delta_i \quad \Delta\rho_i$$

$$\sigma_{\Delta\alpha\, i} \quad \sigma_{\Delta\delta\, i} \quad \sigma_{\Delta\rho_i}$$

Aspect-stable-intervals is time-history of pointing defined in time-bins for which change in pointing has been defined as "stable"; e.g.:

$$t_{\mathrm{start}\, i} \quad t_{\mathrm{stop}_i} \quad < \Delta\alpha >_i \quad < \Delta\delta >_i \quad < \Delta\rho >_i \; .$$

An Aspect-histogram is an image (or volume, including roll) of form:

$$\Delta\alpha_i \quad \Delta\delta_j \quad \Delta\rho_k \quad \Delta t_{ijk}.$$

The fundamental question is: Which is sparser, for a given resolution (accuracy)? Sparseness is proportional to computational speed. Convolution was chosen for computational efficiency. A time-serial method would also work, where for each time-bin, the output image is accumulated by summing a map appropriately rotated and translated.

Issue: quality/uncertainty specifications are placeholder; they need review, detailed specification of algorithm.

6. Transform Tangent-Plane Effective Uniformity Map To Sky

```
I: tangent-plane effective uniformity map
   tangent-plane effective uniformity uncertainty map
   aspect solution
   mean aspect
R: binning parameters (sampling grid, interpolation, magnification)
O: sky-plane effective uniformity map
   sky-plane effective uniformity uncertainty map
S: -
T: exp_coord2sky_image
X: -
P: -
```

D: Rotate, translate, and possibly resample the tangent-plane effective uniformity map (TEUM) to a sky coordinate grid. The input map was computed relative to the mean roll of the observation, with the fiducial axess parallel to the detector axes. Hence, the image must be rotated from the mean roll to the angle of the detector axis on the sky. Nearest-neighbor interpolation should be adequate, if oversampling is 5 to 10 times (this preserves the "sharpness" of the pixel borders).

C: (The transformation of a point from detector tangent-plane to sky tangent-plane should be found in the `pixlib` routines.)

7. Scale Sky-Plane Effective Uniformity Map

```
I: sky-plane effective uniformity map
   sky-plane effective uniformity uncertainty map
   mirror on-axis effective area vs E
   detector mean QE(E)
R: energy, E
   mirror shell, n
O: sky-plane effective exposure map (*)
   sky-plane effective exposure uncertainty map (*)
S: -
T: exp_multiply_image
X: -
P: -
```

D: The sky-plane effective uniformity map is the part of the effective exposure map which depends upon position. It must be scaled by other efficiency factors. Those factors are the mirror effective area (on-axis) for a specified shell at the specified energy (same as Sky-plane Effective Uniformity Map (SEUM), and the mean detector quantum efficiency at the specified energy. The result is an Effective Exposure Map (EEM) *(at one energy)*.

Multiply the sky-plane effective uniformity map by the product of two scalars: mirror on-axis effective area at energy, E, times the mean detector QE at E.

Propagate the mean QE and effective area uncertainties into the sky-plane effective uniformity uncertainty map.

C: This is part of the full exposure map. This part is for one energy, and without the high-resolution structure (bad pixels, high-resolution detector edges).

8. Compute Edge Exposure Map

```
I: edge regions
   bad pixels
   aspect histogram
   mean aspect
   dead-time
R: binning parameters (magnification)
O: sky-plane edge exposure maplets
   sky-plane edge exposure uncertainty maplet
S: -
T: exp_make_edge_map
X: -
P: -
```

D: Compute high-resolution maps of exposure for bad-regions, detector edges, detector gaps. This is similar to the steps from detector uniformity to sky-plane effective exposure map, except that the input images are for small regions, the regions are uniformly unity, the sampling resolution is sub-pixel, and the interpolation is nearest-neighbor. The regions to be processed are edges: single pixel rectangles, defining the perimeter of the devices; bad-pixel regions; and gaps between devices (similar to perimeters, but thicker).

Since the region values are set to unity, the result is time to be be subtracted from the total effective exposure map.

No multiplication by vignetting or other efficiencies (mirror effective area, detector QE) is necessary.

The uncertainty maplets are similar to those in the effective exposure maps, except that the only term is that of the aspect uncertainty.

C: To do the piecewise computation, effective roll centers and translations must be calculated for each maplet so that they may be properly registered with the large-field effective exposure map.

9. MERGE EDGE AND EFFECTIVE EXPOSURE MAPS

```
I: sky-plane effective exposure map
   sky-plane effective exposure uncertainty map
   sky-plane edge exposure maps factors
   sky-plane edge exposure uncertainty map
   binning params
   region
R:
O: Effective exposure map (*)
S:
T: exp_merge_maps
X:
P:
```

D: Resample effective exposure map in region to resolution of edge exposure map. From the exposure map in each edge region, subtract the edge exposure map.

Merge uncertainty maps.

10. ENERGY INTERPOLATE EFFECTIVE EXPOSURE MAPS

```
I: sky-plane effective exposure maps vs E
   sky-plane effective exposure uncertainty map vs E
   efficiency edges
   interpolating function
R: Energy
   interpolation method
O: Effective Exposure Map
```

```
S: nil
T: exp_interp_image
X:
P: nil
```

D: Interpolate the exposure maps at discrete energies to any arbitrary energy within the AXAF bandpass. The interpolation has to know where edges in effective area are. Interpolation method is TBD, but will probably use high-resolution effective area vs energy models.

3.2 Spectroscopy Mode Processing Steps

There are two fundamental differences in spectroscopy-mode maps. First, the vignetting function is not an image, but a *scalar* (at any energy), whose value is the vignetting at the source's zero-order off-axis angle. Second, the final coordinates are diffraction coordinates (angular or wavelength). The map in these coordinates can be derived from the stack of imaging-mode maps by transforming their coordinates given a zero-order centroid, and taking a "slice" through the stack to form a 2D map.

Making the 2D maps just as for imaging-mode will allow transformation to any zero-order coordinate in the field, as opposed to making the map for one-specific zero-order location at a time.

A minor difference is that the grating efficiency is a factor in the final map.

Many steps are exactly identical to Imaging-Mode; they detailed description is not duplicated here - just the heading (and possibly comments on differences in <>).

1. Create Tangent-Plane QE Uniformity Image

   <same as imaging-mode>

2. Form Aspect Histogram

   <same as imaging-mode>

3. Form Dead-time Aspect Histogram

   <same as imaging-mode>

4. Make Tangent-Plane Uniformity Map

```
I: QE uniformity for energy, E and shell, n.
   tangent-plane uncertainty image for shell n at E.
   aspect histogram
   dead-time-aspect histograms
R:
O: detector tangent-plane effective uniformity map
   detector tangent-plane effective uniformity uncertainty map
S: IDL prototype
T: exp_convolve_images
X:
P:
```

   <no vignetting term; same convolution over time as in imaging-mode>

5. Scale Detector-Plane Effective Uniformity Map

```
I: detector tangent-plane effective uniformity map
   detector tangent-plane effective uniformity uncertainty map
   mirror on-axis effective area vs E
   detector mean QE(E)
R: energy, E
   mirror shell, n
   diffraction order, m
O: detector tangent-plane effective exposure map
   detector tangent-plane effective exposure uncertainty map
S: -
T: exp_multiply_image
X: -
P: -
```

D: The detector-plane effective uniformity map is the part of the effective exposure map which depends upon position. It must be scaled by other efficiency factors. Those factors are the mirror effective area (on-axis) for a specified shell at the specified energy (same as Detector-plane Effective Uniformity Map (DEUM), and the mean detector quantum efficiency at the specified energy (the grating efficiency for shell n, energy E, and order, m, is not yet included, since the order need not be specified yet). The result is an Detector-plane Effective Exposure Map (DEEM).

Multiply the detector-plane effective uniformity map by the product of scalars: mirror on-axis effective area at energy, E, for shell n, times the mean detector QE at E.

Propagate the mean QE and effective area uncertainties into the sky-plane effective uniformity uncertainty map.

C: This is one plane of the full exposure map, without a vignetting factor, since zero-rder location hasn't been specified yet, and without the grating efficiency, since an order has not yet been specified. This part is for one energy, and without the high-resolution structure (bad pixels, high-resolution detector edges). It is also still in detector tangent-plane coordinates, since zero-order location has not been specified.

6. COMPUTE EDGE EXPOSURE MAP

```
I: edge regions
   bad pixels
   aspect histogram
   mean aspect
   dead-time
R: binning parameters (magnification)
O: detector tangent-plane edge exposure maplets
   detector tangent-plane edge exposure uncertainty maplet
S: -
```

```
T: exp_make_edge_map
X: -
P: -
```

D: Compute high-resolution maps of exposure for bad-regions, detector edges, detector gaps. This is similar to the steps from detector uniformity to detector tangent-plane effective exposure map, except that the input images are for small regions, the regions are uniformly unity, and the sampling resolution is sub-pixel. The regions to be processed are edges: single pixel rectangles, defining the perimeter of the devices; bad-pixel regions; and gaps between devices (similar to perimeters, but thicker).

<For more details, see imaging-mode description.>

7. MERGE EDGE AND EFFECTIVE EXPOSURE MAPS

```
I: detector tangent-plane effective exposure map
   detector tangent-plane effective exposure uncertainty map
   detector tangent-plane edge exposure maps factors
   detector tangent-plane edge exposure uncertainty map
   binning params
   region
R:
O: detector tangent-plane effective exposure map
S:
T: exp_merge_maps
X:
P:
```

D: Resample effective exposure map in region to resolution of edge exposure map. From exposure map in each edge region, subtract the edge exposure map.

Merge uncertainty maps.

8. TRANSFORM DETECTOR EFFECTIVE EFFIENCY MAP TO DIFFRACTION ANGLES

```
I: detector tangent-plane effective exposure map
   mean aim point
   zero-order centroid
R: spectrum part (HEG | MEG | LEG | LEG-Drake-1 | LEG-Drake-2)
O: Diffraction angle $(r,d)$ effective exposure map (1 plane)
S: code: pixlib?
T: exp_coord2tg_image
X:
P: nil
```

D: Given a zero-order location (from sky coordinate centroid, to detector tangent plane at fiducial mean aspect), transform from the detector tangent-plane to the diffraction angle for the specified part of the spectrum. This is essentially a rotation about the zero-order (e.g., to go from $5°$ oblique angle for HETG, to (r, d) coordinates. There is also a change in scale, due to the angle being measured from the grating node, rather than the mirror node.

> Note: should detector tangent-plane be relative to grating node from the start? what is this coord sys called?

9. Multiply Image By Mirror Vignetting Factor

```
I: Image (diffraction angle, or detector tangent-plane)
   mirror vignetting data (including uncertainty)
   QE uncertainty image in tangent-plane coords.
   mean aim point
   zero-order position
R: energy
   mirror shell n
   interpolation parameters
O: tangent-plane (QE uniformity) * (vignetting) for shell n at E.
   tangent-plane uncertainty image for shell n at E.
S: code: nil
   data: have low-E-res table of area vs off-axis angle.
         Needs to be turned into vignetting only by dividing out
         on-axis area vs E.
T: exp_multiply_image
X:
P: nil
```

D: Mirror vignetting is a function of off-axis-angle (aim-point) and energy. Calculate the vignetting factor at energy E for the mean aim point. Multiply map by the factor.

Propagate vignetting uncertainty into uncertainty map.

10. Extract Spectral Effective Exposure Maps

```
I: diffraction angle effective exposure maps vs E
   diffraction angle effective exposure uncertainty map vs E
   interpolating function
   efficiency edges
   grating efficiency vs energy and order
R: interpolation method
   diffraction order
   spectral part (HEG, MEG, LEG, LEG-Drake-1, LEG-Drake-2)
```

```
O: Spectral Effective Exposure Map
S: nil
T: exp_extract_tg_map
X:
P: nil
```

D: Interpolate the exposure maps at discrete energies to any arbitrary energy within the AXAF bandpass. The interpolation has to know where edges in effective area are. Interpolation method is TBD, but will probably use high-resolution effective area vs energy models.

The spectral map requires transformation from diffraction angle to wavelength, given the order and grating type. From each plane (energy) of the input stack of maps, we are picking a small Δr range corresponding to the energy at that r for the specified order, m. We are effectively taking a two-dimensional slice out of the 3D stack, interpolating from the effective area curve for intermediate energies (defined by r and the dispersion relation, which is incorporated into `pixlib` routines).