Design Specification
ScanCath Reconstruction Algorithms

Prepared For: Edward Solomon, CTO
NovaRay, Inc.
39655 Eureka Drive
Newark, CA 94560

Prepared By: Augustus P. Lowell
Triple Ring Technologies, Inc.
39655 Eureka Drive
Newark, CA 94560

Table of Contents

1 Scope

This specification describes the design requirements for algorithms implementing variable focal-plane reconstruction in the ScanCath system. It includes large-scale timing, synchronization, and data formatting requirements; detailed requirements will be documented in the design documentation for the implementing hardware and software components.

2 Introduction

The ScanCath System is a C-Arm based low-dose, real-time x-ray system intended for Cardiac Electrophysiology and Cath Lab imaging applications. Compared to conventional C-Arm fluoroscopic systems, the ScanCath System provides comparable image quality with significant decrease in radiation exposure.

Typical fluoroscopic systems use a conventional x-ray tube which produces a point-source x-ray beam. This beam is collimated and aligned to an image intensifier which, in conjunction with video-imaging electronics, displays the image.

The ScanCath System has the same intended use as conventional fluoroscope but uses a different geometry. A large scanning-beam x-ray source replaces the conventional fluoroscope's stationary point source and one or more small detector arrays replace the large-diameter image-intensifier and camera. The large x-ray source is created by scanning a target with an electron beam; the target emits x-rays which are passed through a collimator to allow only those directed at the detector to emerge (see Figure 1).

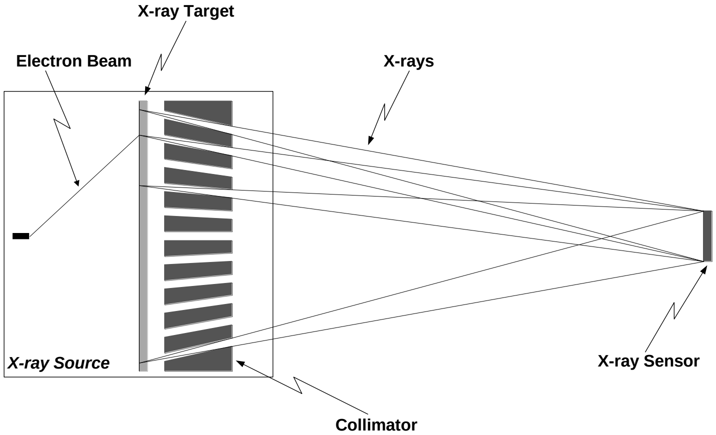

Figure 1: ***ScanCath System Theory of Operation***

The smaller detector and larger detector-patient spacing reduce the scatter seen by the ScanCath System detector. The reduced scatter, as well as higher detector efficiency, results in lower overall x-ray exposure when compared to conventional fluoroscopy systems; patient skin exposure is further reduced by spreading the beam over a larger entrance area than does a conventional system.

Because of the scanning geometry, the x-ray detector is illuminated many times, at least once per collimator hole, for every displayed patient image. Each illumination results in a partial image of the patient, recorded from a viewing perspective determined by the physical position of the collimator hole and the x-ray detector. These multiple images must be combined into a single image for display; the process of combining these images is called *Image Reconstruction*.

Since partial images from many different viewing perspectives are combined, the resulting reconstructed image will represent a focal-plane within the patient: objects within the focal-plane will be in sharp focus, and objects outside the focal plane will be blurred, with blurring increasing as the distance from the focal-plane increases. The effect is similar to that of a camera lens.

To cover the entire patient imaging volume, images from multiple focal-planes must be reconstructed, and image reconstruction algorithms must be suitable for tailoring to achieve the various focal planes required.

3 References

1) Conference Paper, SPIE Medical Imaging Conference 1999, ***Scanning Beam Digital X-ray (SBDX) System for Cardiac Angiography***, February 1999

4 Definitions

4.1 Data Acquisition Sequence

Data is generated by illuminating the imaging volume with x-rays from the x-ray source and detecting photons which have passed through the imaging volume to the sensor. Since objects within the imaging volume will attenuate the x-ray beam by either absorbing or deflecting photons as they pass through the object only photons which have not been absorbed or deflected away from the detector will be observed and the number of photons detected will indicate the relative density of the object along the photon path.

To generate an image of the imaging volume the x-ray source generates a sequence of illuminations from different locations on the large two-dimensional surface of the x-ray tube and the resulting data sets are combined into a composite picture in a process designated *Image Reconstruction*.. The locations of the illuminations on the surface of the tube are defined by the locations (*hole location*) of holes in a collimator covering the tube surface (*collimator holes*); the sampling interval for photon detection corresponds to the illumination of a single collimator hole (a *sample*). The entire sequence of illuminations leading to a single composite image is called the *Scan*, or the *Frame*.

The general sequence of illuminations within the scan is a *raster* which proceeds along each row of collimator holes in turn, completing illumination of one row before proceeding to the next. To control heat-loading within the x-ray source each collimator hole may only be illuminated for a short time before that scan position must be allowed to cool. The total time required to illuminate all collimator holes for their maximum time may be less than that allocated for a scan frame. As a result each row of collimator holes may be illuminated multiple times during the scan, with illumination of other collimator rows intervening to allow sufficient cooling time; this process is designated *Collimator Rescanning*, or *Rescanning*. Nominally every illumination of the same collimator hole generates an identical x-ray path through the imaging volume and the results of multiple scans may be combined by simple summation. However, since the illuminations are separated in time the object in the beam path may move during that interval resulting in motion blurring. Thus rescan sequences and image reconstruction must be designed with motion-blurring consequences in mind.

The x-ray sensor comprises a 2-dimensional array of photon-counting detectors. The entire array is referred to as the *Eye* or the *Sensor*, while each individual detector is referred to as a *Detector* or an *Element*.

In general it is assumed for purposes of reconstruction that detectors within the detector array lie on a fixed grid which allows for ‘separable’ reconstruction (see section 7.2.6). It may be desirable to break up the regularity to improve sampling statistics and this restriction may be relaxed to varying degrees, depending on the implementation; in particular, in certain cases spaces may be inserted between tiles or between individual elements to push elements off the regular grid. In all cases if the spacing in the x-coordinate is constant in the y-dimension and the spacing in the y-coordinate is constant in the x-dimension then the spacing may be accommodated in the detector-pixel mapping (see section 7.2.1) by inclusion of a row-dependent (for column mapping) or column-dependent (for row mapping) offset for each detector and no special implementation support is required.

Other forms of variation in the placement of detector elements with respect to a fixed sampling grid require special accommodation during implementation. These variations will be referred to collectively as *detector dithering*, or simply as *dithering*. Detailed descriptions of various *dithering* schemes, and their implications for reconstruction algorithms and implementation, will be discussed in section 7.2.7.

4.2 m:n reconstruction

The geometry of the reconstruction process lends itself to a compact and convenient method for specifying a focal-plane in terms of the free parameters of the reconstruction algorithm, which we designate the *m:n* reconstruction specification. The definitions of *m* and *n* in this specification are as follows:

> *m* represents the number of pixels spanned in the focal-plane by moving the illumination between neighboring collimator holes (see *Figure 2*).
>
> *n* represents the size of the detector, in pixels, when it is projected along the x-ray illumination onto the focal-plane (see *Figure 3*).

The size of a pixel in the focal-plane is entirely determined by the geometry of the collimator (the distance, or *pitch*, between collimator holes), the distance between the collimator and the detector, and the value of *m* (see Figure 2).

The location of the focal-plane between the collimator and the detector is entirely determined by the geometry of the detector (the distance, or *pitch*, between detectors), the geometry of the collimator, the distance between the collimator and the detector, and the ratio of *m* to *n*, or *m:n* (see Figure 3). Since, in practice, the value of *m* is typically fixed in the system only *n* is varied to move the focal-plane and *n*, by itself, is designated the *Reconstruction Ratio*.

For purposes of specifying the image reconstruction the following system geometry specifications apply:

D_{sd}	Distance from the source (x-ray emitter) to the x-ray detector
D_{sp}	Distance from the source (x-ray emitter) to the focal-plane
L_s	Source Pitch (distance between collimator holes)
L_d	Detector Pitch (distance between detector elements)
L_p	Pixel Pitch (pixel size) in the focal-plane

4.3 α-Noise

As an artifact of the binning nature of the image reconstruction process the resulting reconstructed image contains a regular pattern of pixel gain errors which appear as a grid superimposed on the image. This artifact is designated *α-noise*. The gain adjustment used to correct for this artifact is designated *α-correction*.

4.4 Edge-Gain Artifact

As a result of the reconstruction process the image intensity at the edge of the image increases from 50% to 100% (under flat-illumination conditions) (see section 7.2.8) over some number of pixels; the number of pixels over which this variation occurs is dependent on the focal-plane and can vary by 2:1 or more between the top and bottom of the imaging volume. The variation in width from focal-plane to focal-plane results in an intensity mismatch between focal-planes in this transition region which may be thought of as a mismatch in imaging system gain. This mismatch is called the *Edge-Gain Artifact*.

5 Nomenclature

Requirements are designated within in this document by the words “shall”, “will”, “should”, “is”, and “are”. These terms are interchangeable and usage is driven entirely by context and stylistic considerations.

Information presented in ***<bracketed bold italics>*** is preliminary and must be reviewed and verified prior to product release.

Four levels of requirement may be identified:

- ***Mandatory***: These requirements must be implemented and verified for product release.

- ***Required***: These requirements must be implemented and verified for product release unless specifically exempted by the Project Manager during product development or at the time of product release. Items included in the final product implementation must be fully verified prior verification and validation; items not included in the final product implementation must be exempted in writing as part of the verification and validation process.

- ***Desired***: These requirements comprise design goals, performance targets, and “nice to haves” that are desirable but not necessary to product operation. Items in this list may be included or excluded by the implementing designer or Project Manager as resources and technology permits. Items that are included in the final product implementation must be documented prior to product release; *desired* product features and functions must be verified as part of verification and validation, but *desired* design goals and performance targets need not be formally verified.

- ***Guidance***: These requirements are provided for guidance and clarification to the designer regarding the intent of other specific requirements, expected product usage, external interfaces and constraints, and preferred or potential design approaches, technologies, and practices. They are not intended for formal tracking and verification. These items should be taken into account during design and deviations should be documented within the standard design documentation, but they are not binding either for design or for verification.

Required, *Desired*, and *Guidance* requirements must be specifically designated as such within the text of the specification. Any requirement without such a designation is presumed to be *Mandatory*.

6 System Constraints

6.1 Timing

The imaging system must produce ‘live’ video images based on the sampled data at frame rates up to 30Hz and all image reconstruction and image processing algorithms must be consistent with that. In particular, algorithms for image reconstruction and image processing must operate in real-time on streaming data at the maximum scan frame rate (30Hz). Internal buffering and data reordering is permitted provided the total delay between the final sample of an image pixel at the x-ray detector and the availability of that pixel at the output of the reconstruction process is less than ¼ of a scan frame for the largest scan. Note that the reconstruction process includes any artifact detection/removal steps.

6.2 Implementation

Image reconstruction and artifact detection/removal algorithms must be suitable for implementation in available FPGA or numerical processor architectures; external memories may be used to augment on-board FPGA/processor memory resources.

7 Algorithms

7.1 Pixel Size and Focal-Plane Specification

The relationship between imaging system geometry, the chosen *m* value, focal-plane location, and pixel size is shown in Figure 2 which illustrates illumination of the detector from two neighboring collimator holes.

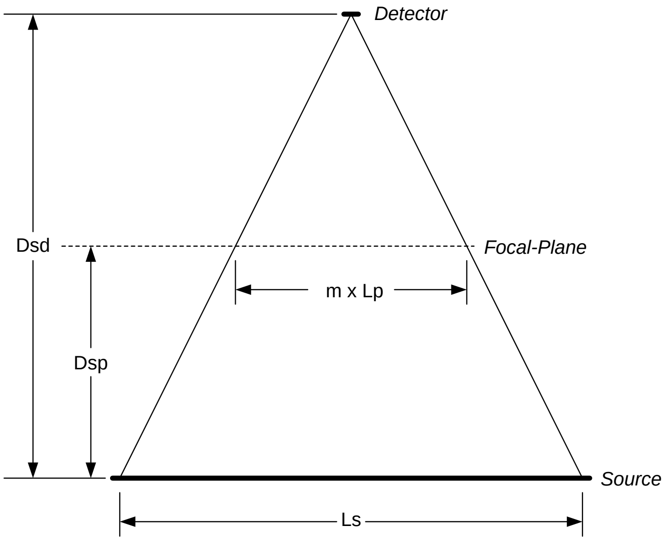

Figure 2: *Pixel-Size Based on* m

From the assessment of similar triangles in the drawing, we can derive the following relationship:

$$L_p = L_s \times \frac{1}{m} \times \frac{D_{sd} - D_{sp}}{D_{sd}} \tag{1}$$

The relationship between imaging system geometry, the chosen *n* value, and pixel size is shown in Figure 3, which illustrates illumination of two neighboring detector elements from a single collimator hole.

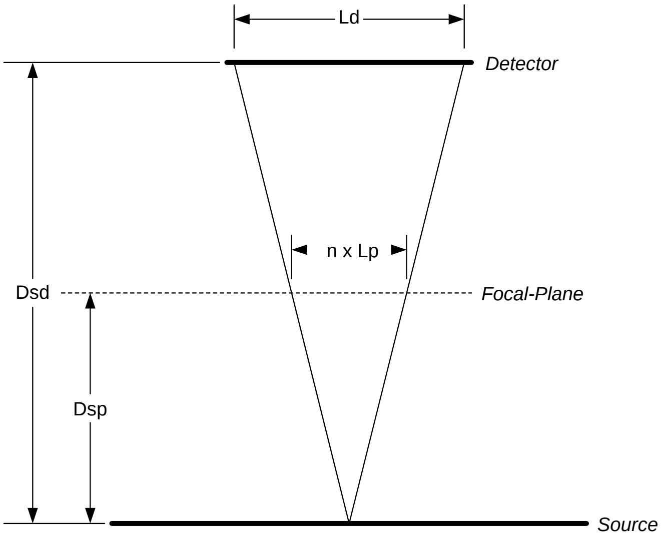

Figure 3: *Pixel-Size Based on* n

Again, based on analysis of similar triangles from the drawing, we can derive the following relationship:

$$L_p = L_d \times \frac{1}{n} \times \frac{D_{sp}}{D_{sd}} \tag{2}$$

Combining these two relationships we can specify the relationship between system geometry, the *m:n* ratio, and the location of the focal-plane:

$$D_{sp} = D_{sd} \times \frac{L_s}{L_s + L_d \times (m:n)} \tag{3}$$

For a fixed value of *m* the focal-plane is uniquely specified by the value of *n* – the *reconstruction ratio* – so we also specify the following relationship for use in implementation:

$$n = m \times \frac{L_d}{L_s} \times \frac{D_{sp}}{D_{sd} - D_{sp}} \tag{4}$$

7.2 Image Reconstruction

7.2.1 Projections and Mappings

From the fact that we span *m* image pixels for each collimator hole we can conclude that the total size of the reconstructed image, in pixels, will be (*m* x Collimator Holes) or, alternatively, that the required value of *m* to achieve the desired image size is (Image Size ÷ Collimator Holes). For ease of implementation we will restrict *m* to integral values.

Image reconstruction is fundamentally a mapping process which may be thought of either as a forward projection in a straight-line from the source through a pixel in the focal-plane onto detectors or as a backward projection in a straight-line back toward the source from the detector onto pixels in the focal-plane. The back-projection technique is optimal for determining into which image pixels a specific detector datum must map and that is the representation we will use for our algorithm discussions.

As a note on what either projection means for the resulting image consider the drawing in Figure 4. The mapping process can be thought of as assigning rays passing between the source and the detector onto a *Virtual Image Plane* – virtual in the sense that it is created by a mathematical mapping process rather than by physically redirecting the rays and therefore exists only logically in the internal memory of the reconstruction engine. Rays passing through the object at position *A*, in the focal plane, land on detectors that are mapped to a single location in the virtual image plane and reinforce to create an image of the object. Rays passing through an object at position *B*, outside the focal plane, get redirected by the same mapping function, but end up diverging in the virtual image plane, thus blurring the image of the object over a wide area. This, in effect, creates a digital lens which may be refocused to the desired focal plane by adjusting the mapping function.

The back-projection of a detector onto pixels in the focal-plane (in this case with $n \approx 2.67$) is shown for one-dimension in Figure 5. The back-projection will not, in general, result in detectors and pixels aligning perfectly with each other, nor is it necessary that the detector projection map uniquely to a single image pixel. In particular, for the ScanCath system geometry the detector projection varies over a wide range of *n* values and will often be larger than an image pixel, overlapping multiple image pixels.

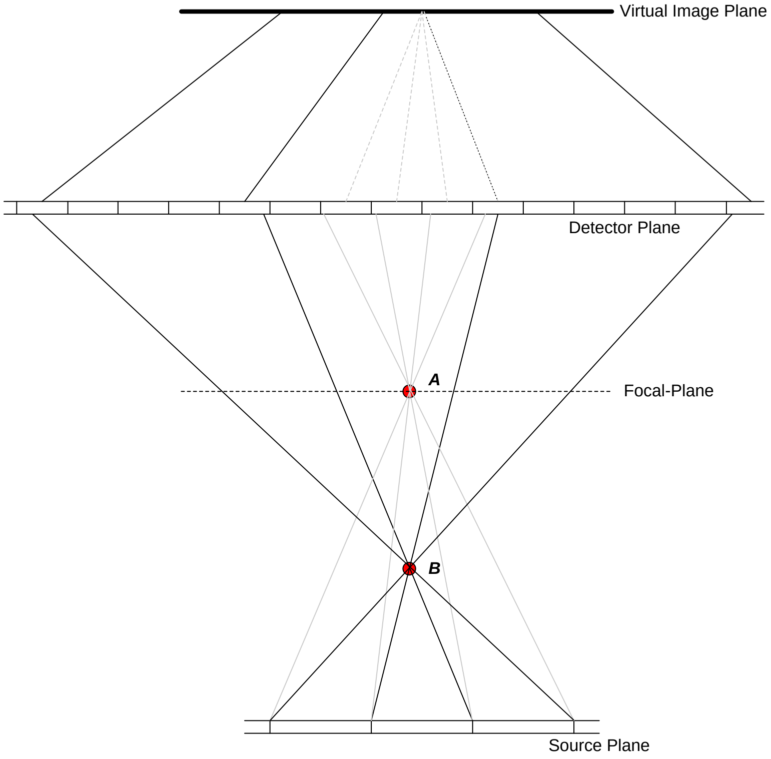

Figure 4: ***Creation of Focal-Planes***

Given that the effective detector size when projected onto the pixel grid is larger than one pixel the ScanCath system is inherently under-sampled and depends on resampling of pixels at multiple phases when illuminated from multiple collimator holes to achieve its specified spatial resolution performance.

Since the progression of phase for sampling depends on the ratio of *m* to *n* there will be some focal planes that are poorly sampled because *m* and *n* are related by common integral factors; in the worst case (for instance, m:n is an integral multiple of 2) the resampling will align precisely on the original grid and no oversampling is possible. See section 7.2.7 for a discussion of methods to improve sampling.

Image reconstruction for a given focal-plane is entirely defined by the mapping of detector elements onto the pixel plane; the image reconstruction implementation must then use that mapping to assign detector data as they are received into the specified image pixels.

The focal plane is defined by locations of the detector element center-lines on the pixel grid; in general we align the center of the x-ray sensor array with some reference point in the image and then move by *n* image pixels in the focal-plane for each detector element in the sensor. Overall the entire sensor array will map onto an area of the image that covers (*n* x Elements) image pixels, which is referred to as the *Reconstruction Window*. The reference point is chosen such that the ray from the center collimator hole to the geometric center of the detector array passes through the geometric center of the image. As defined earlier we move the mapping by *m* image pixels for each collimator hole.

Note that if *m* is restricted to be an integral number of pixels the mapping of detector elements to pixels within the reconstruction window is identical for each collimator hole – the window moves by *m* pixels as you move from collimator hole to collimator hole, but the mapping within the window does not change. This, combined with the fact that the shift itself is an integer, makes implementation of the mapping function much easier and is a requirement of the algorithm for ScanCath implementation.

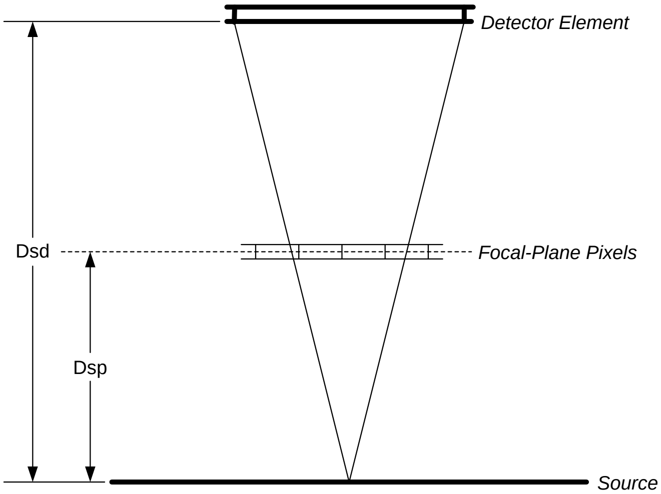

Figure 5: ***Detector Back-Projection Onto the Focal-Plane***

If the detectors do not lie on a regular sampling grid the relative offset between each detector and the sampling grid must be taken into account when doing the detector↔pixel mapping. This is a straightforward matter – each detector position becomes a function both of its location within the array *and* of the offset at that point, rather than just of its position within the array. Provided the offsets along the row are identical for all rows and the offsets along the column are identical for all columns the mapping function generates a single mapping for rows and a single mapping for columns just as it would for regularly-spaced detectors.

Since the detector element overlaps multiple image pixels there is, in principle, no way to know for sure which pixel a given detected photon passed through and there are many ways of assigning the detector datum to the various candidates. The minimum-blurring mapping, known as *nearest-neighbor reconstruction*, assigns the datum exclusively to the pixel along the source↔detector center-line – the pixel geometrically ‘nearest’ to the ray between the source and the detector. This has the added benefit of being the simplest possible mapping; however, the resulting distribution of noise power has some undesirable characteristics and it suffers from the drawback that it tends to exaggerate binning artifacts such as α-noise and leave obvious image artifacts in focal-planes where sampling is less than optimal.

A preferred approach is to split the datum among the various overlapped pixels according to some weighting scheme; this technique is known as *fractional-binning reconstruction*. The most obvious – and intuitively satisfying – weighting scheme is to weight the datum according to the amount of area overlap. In a sense, this assigns data to pixels in proportion to the probability that the photons actually passed through them. This is the general scheme the ScanCath image reconstructor will use. Fractional resolution on the weightings shall be sufficient to obtain the general benefits – improved noise-power distribution, reduced α-noise, and reduced sampling artifact – within the constraints of available FPGA resources.

7.2.2 Noise-Texture Constraints

Implied in the discussion of area overlap is the fact that each detector datum will be split among various neighboring pixels with a weighting determined by the value of *n*, which results in different weightings in different focal-planes even for identical detector↔pixel alignments. This process is roughly equivalent to applying an *n*x*n* filter kernel to the image; **note** that the area overlap scheme described in section 7.2.1 is equivalent to a uniform (boxcar) kernal weighted with a rectangular window of width *n*. The filtering process results in some attenuation of high-frequency image components and a reduction in uncorrelated noise by a factor of about *n*.

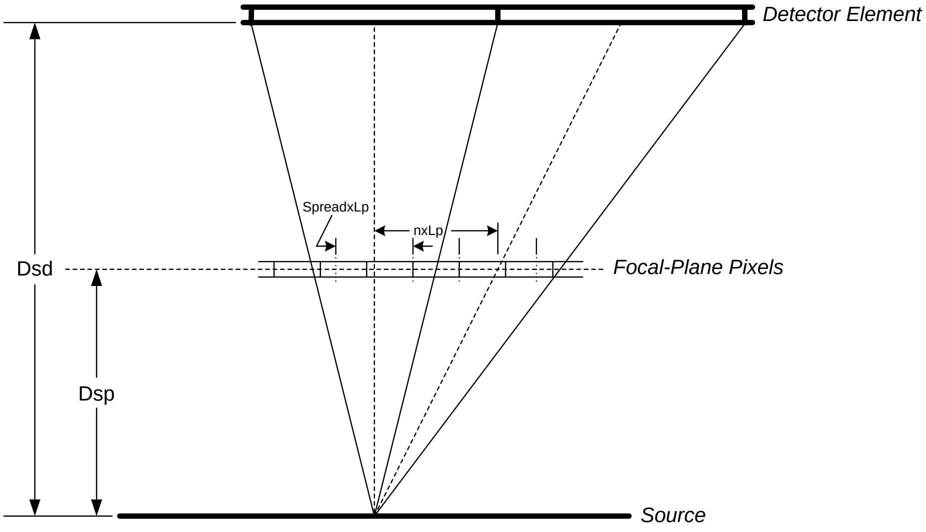

Figure 6: ***Use of*** **Spread** ***for Mapping***

For viewing a single focal-plane this is not problematic – in the worst-case the effect can be undone by judicious equalization filtering – but the ScanCath system must allow for an auto-focus operation in which features of interest from various focal-planes are combined into a single composite image. In this case neighboring regions of the composite image are likely to be from different focal-planes and it has been observed that the differing noise-textures resulting from different effective kernel widths in those regions create visible boundaries between the regions. For this reason we must modify our mapping functions to minimize the differential effects on noise-power between focal-planes.

To this end we define a second mapping variable, the *Pixel Spread* or *Spread*, which is used to determine distribution of data to pixels in the mapping process. Under this modification the *n* value is used to determine the position at which the center of the detector projection intersects the image pixel grid – which determines the focal-plane – and the *Spread* value is used to calculate the area overlap for determining weightings; the new process is shown in one dimension in Figure 6 for $n \approx 2.67$ and $Spread \approx 1.67$.

To eliminate the differential effects on noise-power between focal-planes, we define a standard *Spread* value that is used for all focal-planes.

Note that this modified mapping is also equivalent to a uniform kernel weighted with a rectangular window, this time of width *Spread*. Consistent with the need to equalize effects on noise-power between focal planes the actual mapping kernal may be modified – for instance to use a Gaussian rather than rectangular window function – to improve overall imaging performance.

7.2.3 Synthetic Fractional-Binning

It has been noted that fractional-binning is conceptually similar to performing a nearest-neighbor reconstruction and then filtering the resulting image with a blurring kernel corresponding to the fractional-binning weighting scheme – a technique we will designate as *synthetic fractional-binning reconstruction*. However this process is not identical, because the effective blurring kernel created by fractional-binning reconstruction is non-stationary – its phase with respect to the pixel grid varies from pixel to pixel. There has been great debate over whether the differences between these techniques are significant – whether they would be observable in practice – and there has been no definitive answer to date. One way to reduce the difference – and thereby make the latter technique more attractive as an alternative reconstruction method – is to perform nearest-neighbor reconstruction and blurring on an oversampled pixel grid and then down-sampling to the final pixel resolution. One analysis has indicated that a 2:1 oversample reduces the differences enough to be insignificant even in an analytical sense, and an implementation of synthetic fractional-binning reconstruction using a 2:1 oversample is considered an acceptable alternative to fractional-binning reconstruction.

7.2.4 Hybrid Fractional Binning

Given the assessment in section 7.2.2 an alternative hybrid reconstruction algorithm, designated *hybrid fractional binning reconstruction*, has been proposed in which a fractional binning reconstruction is done on a 2x2 pixel grid – equivalent to specifying a *Spread* of 1.0 (see section 7.2.2) – and then the resulting image is filtered with a blurring kernel as described for *synthetic fractional binning* (section 7.2.3). Because the original 2x2 fractional binning operation provides the non-stationary kernel mapping of true fractional binning this operation does not require that we use an oversampled pixel grid for reconstruction prior to filtering.

The advantage of the hybrid technique is that it reduces the original fractional binning reconstruction process to a simple bi-linear interpolation, which is supported directly as a single-cycle operation by many standard graphics processors. Hence the hybrid method is particularly well-suited to implementation on a graphics processor.

A reconstruction based on *hybrid fractional binning* is considered an acceptable alternative to fractional-binning reconstruction.

7.2.5 Out-of-Plane Blurring

The source of the out-of-plane blurring is illustrated quantitatively in Figure 7 which shows the locations at which rays passing through a single point in the focal plane intersect other planes.

The magnitude of the spread may be calculated from similar triangles and is described by:

$$b_L = d_L \times \frac{D_d}{D_{sd} - D_{sp}} \tag{5}$$

$$b_H = d_H \times \frac{D_d}{D_{sd} - D_{sp}}$$

or more generally as:

$$b = d \times \frac{D_d}{D_{sd} - D_{sp}} \tag{6}$$

where *d* is the distance from the focal-plane and *b* is the span over which the point is blurred. The effects of this blurring on spatial resolution and Modulation Transfer Function (MTF) are described in reference 1.

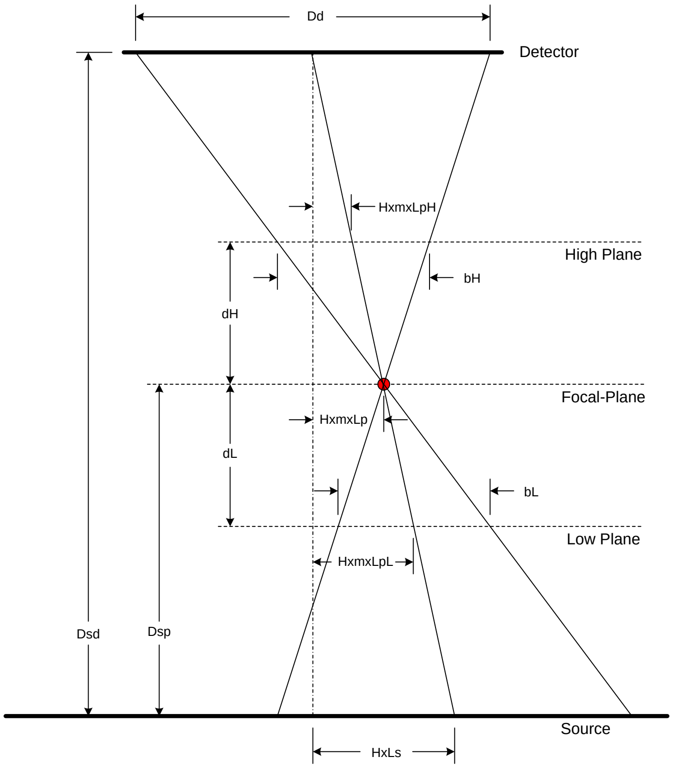

Figure 7: Object *Blurring Into Adjacent Focal Planes*

From Figure 7 we can also see that the distance, in pixels, from the image center-line to the center of the blurred object is (*m* x H) for all focal planes, where *m* is as defined for the reconstruction specification in section 7.1 and H is the distance, in collimator holes, from the source center line. This implies that using a constant value for *m* in specifying reconstruction for focal planes ensures that object will remain centered at the same image pixel location in all focal planes. This is the genesis of the constraint that *m* must be held constant for specifying reconstruction.

7.2.6 Separability

Although for simplicity the figures have shown projections in one dimension the general discussion of detector↔pixel mappings have treated it as a two-dimensional process and that is the ideal. However, hardware resource requirements for general two-dimensional mapping architectures are prohibitive. For this reason the image reconstruction algorithms may be designed to implement a separable mapping function which allows separate reconstructions along the row and along the column.

7.2.7 Detector Dithering

Because the ScanCath system is inherently undersampled (section 7.2.1) we wish to ensure that the resampling of pixels at various collimator holes provides enough phase variation to meet the spatial resolution requirements. The primary way of achieving this is to vary the placement of the detector elements with respect to the sampling grid defined by the detector pitch. Various methods for doing this and their implications for sampling are discussed in the following sections.

As noted in section 7.2.1 if the spacing of detectors along the row is the same for all sensor rows and spacing along the column is the same for all sensor columns then spacing variations may be accommodated by the simple addition of offsets within the mapping process and the entire mapping is naturally separable and can be accommodated in any reconstruction implementation. If, however, detector spacing varies in more complicated ways particular implementation support may be necessary to maintain the ability to perform separable reconstruction. Such variations are known collectively as *detector dithering* and fall into two broad classes:

- *Spatial Dithering*: the detector x-coordinate varies along the column – not all sensor rows are spaced the same – and/or the detector y-coordinate varies long the row – not all sensor columns are spaced the same.

- *Temporal Dithering*: the detector coordinates vary in time – the detector moves in space between samples.

Although the two dither methods seem completely different in concept they are in fact almost identical in implication and the reconstruction mechanisms required to support them are the same. However, the reconstruction mechanisms required to support variations in the X-coordinate are fundamentally different than those required to support variations in the Y-coordinate and the two will be discussed separately.

7.2.7.1 Dithered Reconstruction Techniques

7.2.7.1.1 Variable X-Coordinate

When detector x-coordinates vary from sensor row to sensor row the result is that the detector↔pixel mapping – and the mapping coefficients used for reconstruction – vary from row to row. Since the separable reconstruction process treats each row independently this implies that all that is required to handle this condition is different sets of mapping coefficients for each row – or more specifically, since rows with identical x-coordinate configurations will be mapped identically, a unique set of mapping coefficients for each unique x-coordinate configuration.

From an implementation standpoint this requires merely that we maintain as many coefficient sets as there are x-coordinate configurations and that the appropriate coefficient sets can be applied to the appropriate rows during reconstruction.

7.2.7.1.2 Varable Y-Coordinate

By the time we process columns in a separable reconstruction the sensor data has already been mapped from sensor coordinates into pixel coordinates along the rows and we have lost our sense of sensor columns. Thus when the detector y-coordinates vary from sensor column to sensor column we cannot simply assign different coefficient sets to different sensor columns as we can along the rows. Instead we must make accommodation during row processing for the different column configurations.

To do this we note that what defines a unique sensor 'row' in the first place is the y-coordinate of that row. In general we provide as many row processors to process unique sensor rows as there are unique y-coordinates in the sensor array. In the case of a sensor in which the y-coordinate does not vary along the sensor row each sensor row maps uniquely to one row processor. But if the y-coordinate of detectors *does* vary along a sensor row we could as easily map an additional row processor to each unique y-coordinate along that row and then assign the data associated with that y-coordinate to the appropriate row processor.

Assume, for instance, that half the detectors along the row are shifted by exactly ½ detector in y-. We now have two unique y-coordinates for that row: the original position and a position shifted by ½ detector. Each of those positions corresponds to a unique sensor 'virtual' row and is assigned a 'virtual' row processor to handle the data from detectors corresponding to that 'virtual' row. Thus as a general proposition we can say that a dither which creates *n* unique y-coordinates along each sensor row can be handled by creating *n* 'virtual' row processors to handle each physical sensor row. Of course each 'virtual' row processor need only handle the subset of sensor detectors which correspond to its y-coordinate.

In practice, since data is typically handled in the row processors one column at a time it is not necessary to duplicate the entire row processor to create these 'virtual' rows; we need only duplicate the row buffers which hold the data and switch the row processor on-the-fly to operate on the row buffer appropriate for the particular detector being processed.

Note that the column processor must process all the 'virtual' rows so the result of this process is that the column processor must process *n* times as many rows per sample – that it must run *n* times as fast – and its detector↔pixel mapping must correspond to mapping of the 'virtual' row coordinates not the sensor physical rows.

7.2.7.2 Dithering Schemes

7.2.7.2.1 'Pinwheel' Spatial Dither

The eye configuration shown in Figure 8, called a 'Pinwheel' eye, provides twice the effective sampling resolution in both x- and y- of a standard eye. It does this because each pixel receives data from many different parts of the eye, sampled at different collimator holes, so if different parts of the eye are aligned on different sampling grids sampling is improved. To achieve its improvement it introduces a spatial dither of ½ detector in y- at the horizontal center of the eye and a spatial dither of ½ detector in x- at the vertical center of the eye.

Note that this method does not improve the precession of sampling phases; rather by doubling the effective sampling resolution it guarantees that the size of a detector when projected on the effective sampling grid is always less than one pixel for the defined system geometry.

As described in section 7.2.7.1 the single dither in the x-coordinate requires two sets of mapping coefficients for the row processor; in this case one applies to the top half of the eye and the other applies to the bottom half of the eye.

As described in section 7.2.7.1.2 the single dither of ½ detector in the y-coordinate requires creation of pairs of 'virtual' reconstruction rows on a ½ detector grid (at $\pm{}^{1}/_{4}$ detector relative to the real rows) into which row data can be mapped; in this case one 'virtual' row receives data from the left half of the eye and the other receives data from the right half of the eye.

It should be noted that the sampling advantages obtained from the 'Pinwheel' eye only apply to pixels which receive data from the entire eye. In particular, because of the group-delay effects of reconstruction (see section 7.2.8) pixels near the edge of the image will receive data predominately from one half of the eye, and the sampling advantage degrades or disappears in these regions.

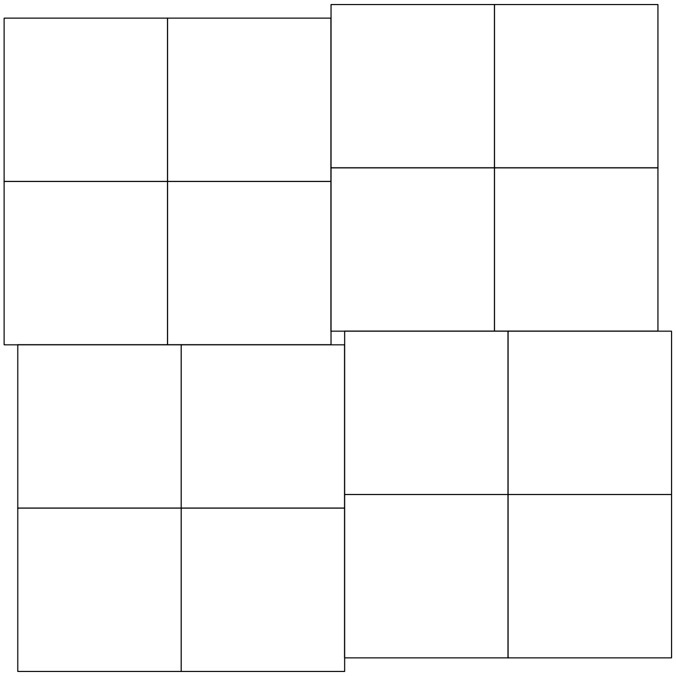

Figure 8: ***'Pinwheel' Eye Configuration***

7.2.7.2.2 Rotated Eye

A particularly favorable way to dither detectors to improve sampling is to rotate the eye relative to the scan coordinates, such that detector positions along a row or column precess continuously with respect to a fixed sampling grid aligned with the scan/pixel coordinate grid. This technique increases the effective sampling resolution of the detector array by approximately R/N_d, where R is the displacement at the edge of the detector caused by the rotation and N_d is the number of individual elements along one dimension of the detector array. While there are techniques for separably reconstructing data from such a rotated eye precisely they have some drawbacks and the resources required for implementation are currently prohibitive. For that reason, although these techniques would offer the best possible reconstruction we will not address such techniques here.

If we compare the rotated eye in Figure 9 with the 'pinwheel' eye in Figure 8, however, an approximate method for reconstructing data from the rotated eye suggests itself: treat it as a pinwheel eye for purposes of reconstruction. Simulations have shown that this method provides better performance than a pinwheel eye (with much simpler eye mechanical construction) and much of the benefit of a true rotated reconstruction. However to obtain maximum benefit from this method the rotation angle must be such that the y-coordinate varies by at least one full detector as you move from center to edge along the x-dimension, rather than by ½ detector as it does in the pinwheel eye. This means that although we can still operate using 'virtual' rows on a ½ detector grid the data from a single row of the rotated eye must be spread over 4 of the 'virtual' rows – at $-^3/_4$, $-^1/_4$, $+^1/_4$, and $+^3/_4$ relative to the real rows – rather than only to the two at $\pm^1/_4$ required for the pinwheel eye. This makes the internal structure of the row reconstructor slightly more complicated but not prohibitively so.

*** DRAFT ***

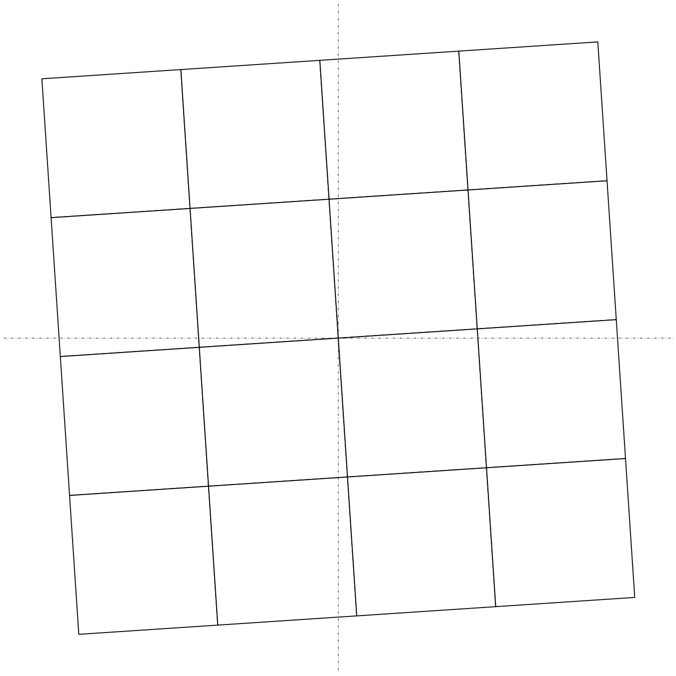

Figure 9: ***Rotated Eye Configuration***

We should note that, because the rotation is effectively twice that of a 'Pinwheel' eye, the rotated eye provides the sampling advantages all the way to the edge of the image and so is superior to the 'Pinwheel' eye in the edge regions.

7.2.7.2.3 'Shimmy' Temporal Dither

Another way to double the effective resolution of the sampling grid is to move the eye back and forth by ½ detector in the x-dimension as we move along the y-dimension in the scan and by ½ detector in the y-dimension as we move along the x-dimension in the scan. Again, since each pixel receives multiple samples taken from different collimator holes the net effect is to combine samples from different sampling grids into the same pixel. And, again, it achieves the effect by creating a spatial dither of ½ detector in both x- and y- but by doing so across the entire eye at different points in time rather than in different parts of the eye at the same time.

Of course physically moving the eye fast enough to accommodate the sampling rates of the ScanCath system is impossible. To achieve temporal dithering we must create an eye which inherently oversamples by a factor of at least two and then electronically combine the oversampled 'meso-elements' into the 'dithered' elements for reconstruction. A 'meso-element' configuration to achieve this, and the pattern of meso-element grouping from sample-to-sample, is shown in Figure 10. Arrays of 2x2 half-sized meso-elements are combined into single elements; the location of the effective center of the resulting element depends on the alignment of the grouping on the half-size grid. **Note** that as shown in the figure the x-location of the element varies as you move along the scan column but not along the scan row, and the y-location of the element varies as you move along the scan row but not along the scan column. This satisfies the separability requirements defined in section 7.2.7.1.

As with any dither method the ½ detector shift in the x-coordinates requires creation of a second set of mapping coefficients corresponding to the shifted configuration. However, in this case the alternate coefficient sets are applied to the entire eye for given collimator holes but are switched between collimator holes rather than being applied separately to different parts of the eye.

*** DRAFT ***

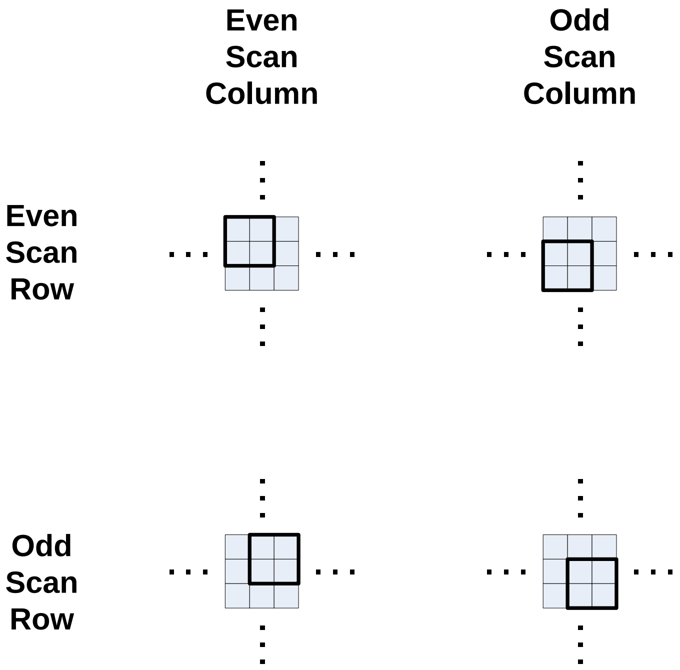

Figure 10: ***'Shimmy' Dither Configuration***

As with any dither method the ½ detector shift in the y-coordinates requires creation of pairs of 'virtual' reconstruction rows on a ½ detector grid (at $\pm{}^{1}/_{4}$ detector relative to the real rows) into which row data can be mapped; again, in this case the assignment of data to the 'virtual' rows is the same for the entire eye for a given collimator hole but varies between collimator holes.

7.2.8 Kernal Width and Group Delay

As a mapping process which assigns input samples to output samples the image reconstruction process may be represented as an FIR filtering process. In general the filter is non-stationary – that is the input→output transformation is different for different input and output samples – but in aggregate it may be considered as a filter with an (*n* x Detector Elements) by (*n* x Detector Elements) rectangular-windowed kernel.

As a result of this representation it becomes clear that the reconstruction process must have both a startup and a shutdown transient as the filter is filled/cleared and an overall group delay between input and output. Since the *n* value – and therefore the effective kernel size – is different for different focal-planes the width of the startup/shutdown transient and the effective group delay will be different for different focal-planes.

Since the auto-focus process must combine different planes into a composite image the resulting misalignment of pixels between planes is a problem. In addition the dark edges at the beginning of the startup transient and the end of the shutdown transient contain too little data to be useful for imaging.

To overcome these limitations the image reconstruction algorithm shall be designed to align the centers of the effective reconstruction kernels with the left and right edges of the displayed image; the effect of this is to align the pixels from focal-plane to focal-plane and to clip image display for regions of the startup/shutdown transients in which the image intensity is less than 50% of its final/initial value. In practical terms this means choosing a standard center location within the image reconstruction buffer at which to center the reconstruction window and delaying image display until the output source location reaches that 'center' position.

7.3 Artifact Correction

As noted in section 4 the image reconstruction process generates several artifacts which reduce the overall quality of the displayed image. These artifacts must be detected and removed prior to image display.

7.3.1 α-Noise

α-noise is a patterned gain error that presents itself as a grid pattern superimposed on the image. It is a result both of data binning in the reconstruction process and of variance in weighting – due to gain differences in the x-ray detector and to intensity variations in the x-ray illumination – of x-ray samples. For a given reconstruction mapping, and for constant x-ray detector gains and x-ray illumination patterns, the α-noise pattern repeats every *m* pixels along both row and column. If both the x-ray detector gains and the illumination pattern are known the α-noise pattern is predictable and a correction may be calculated directly. In this case only (*m* x *m*) corrections must be calculated because the pattern repeats with a period of *m* pixels in both dimensions across the entire image.

In general, however, the x-ray detector gains and illumination pattern are not known a priori with any precision, both tend to drift over time, and the x-ray illumination pattern tends to vary over the physical extent of the x-ray source. For these reasons a single image-wide estimate of α-correction will generally not be sufficient to remove the α-noise.

To accommodate the dynamic nature of the x-ray detector and the x-ray illumination we must estimate α-noise and α-correction on the fly from the live image data. This requires use of a general class of algorithms that average over some number of (*m* x *m*) tiles in the image to extract the pixel-by-pixel variation from the mean; if the tiles are well-chosen, such that the α-noise pattern reinforces during the averaging process and the underlying image structure dissipates, then the ratio of the pixel mean to the tile mean is the α-noise.

In combination, the algorithms must reduce alpha-noise to an undetectable level without leaving significant artifacts in the image. Details of the algorithms will be specified in the design documentation for the implementing hardware blocks. Techniques for detecting and removing artifacts associated with α-correction are described in section 7.3.1.1.

7.3.1.1 Artifact Detection/Removal

The intensity-binning α-correction algorithm is prone to artifacts which resemble horizontal bars when the following conditions apply:

1) There is little intensity variation across the row such that long sequences of pixels along the row all lie within the same intensity bin and there is a significant variation in the α-noise pattern along the row.
2) There is a significant variation in the α-noise pattern along the column and there is a sudden change in intensity along the row such that the binning shifts to a bin that has not been updated recently.

Since both of these conditions occur when there is significant spatial variance in the α-noise pattern the intensity binning algorithm is not appropriate for when that is the case. Under these conditions this algorithm should either be deactivated or configured for a single bin with a very long time-constant which effectively creates a global (across the entire image) *m*x*m* correction; it is roughly equivalent to using the spatial estimation method over a 'neighborhood' which encompasses the entire image.

The spatial-binning algorithm is prone to 'shadow' artifacts along narrow high-contrast edges. These 'shadows' are the result of gain errors in the estimator as the sparse (individual pixel within the tile) filter kernels cross the edge and manifest as large tile-to-tile variations in the α-noise estimate. Since the underlying assumption of the spatial-binning algorithm is that α-noise is similar in adjacent tiles – that there are *not* large tile-to-tile variations in α-noise – the presence of these variations is a direct indicator that the estimate contains an artifact.

To detect the artifact we calculate the tile-to-tile gradient in the α-correction estimate for each pixel. If the gradient exceeds some threshold we assume the estimate is an artifact and we substitute a correction value of 1 (no correction); alternatively we could define a continuous weighting function of the gradient that could be applied either to the individual correction or to all corrections in the neighborhood to synthesize a 'weighted' correction value.

Note that this technique works best if the spatial-binning estimator is a second-pass correction such that the resulting uncorrected pixel has already had a first-cut correction (perhaps a global correction generated from a single intensity bin) applied.

Details of the algorithms for artifact detection/removal will be specified in the design documentation for the implementing hardware blocks.

7.3.1.2 Edge-Gain

As a result of the variable width of the startup/shutdown transient for the reconstruction kernel (see section 7.2.8) the effective imaging system gain varies at the edge of the image and the variation is different in different focal-planes. The resulting focal-plane to focal-plane gain mismatches constitute an edge-gain artifact which must be corrected for proper functioning of the auto-focus algorithms.

Since the edge-gain artifact is the result of filtering with different effective kernel sizes the mismatch may be estimated by comparing the results of different size filters on a 'nominal' (flat-field) test data set. To generate the test data set we create a *Scan Mask* corresponding to the collimator illumination pattern: the scan mask will have ones at image positions corresponding to illuminated collimator holes and zeros at image positions corresponding to unilluminated collimator holes; since collimator holes are separated by *m* pixels in the image the mask pattern will be filled in (*m* x *m*) pixel blocks.

The scan mask is then filtered using two FIR boxcar filters with different kernel widths, one corresponding to the current reconstruction mapping [(*n* x Detector Elements) by (*n* x Detector Elements)] and one corresponding to a *Reference Plane* to which we wish to normalize. The filter results represent the effective gains for the two different reconstructed planes and the gain correction is the ratio of the two.

8 Approvals

Revision: A

Title	Triple Ring Technologies, Inc. Approvals	Effective Date
Author		
Project Manager		
RA/QA		
	Client Approvals	
Project Manager		
Regulatory		

9 Revision History

Revision	Description of Change	Revision Date
A	Initial Release	9 April 2008