# Algebraic Scene-Based Nonuniformity Correction in Focal-Plane Arrays

Bradley M. Ratliff [a,*], Majeed M. Hayat[b,a,**], and Russell C. Hardie[a,b,***]
[a] Department of Electrical and Computer Engineering, University of Dayton, Dayton, OH
[b] Electro-Optics Program, University of Dayton, Dayton, OH

## ABSTRACT

An algorithm is developed to compensate for the spatial fixed-pattern (nonuniformity) noise in focal-plane arrays, which is a pressing problem, particularly for mid- to far-infrared imaging systems. The proposed algorithm uses pairs of frames from an image sequence exhibiting pure horizontal and vertical sub-pixel shifts. The algorithm assumes a linear irradiance-voltage model for which the nonuniformity is attributed only to variation in the offset of various detectors in the array. Using a modified gradient-based shift estimator, pairs of frames exhibiting the above shift requirements can be identified and used to generate a correction matrix, which will compensate for the offset nonuniformity in all frames. The efficacy of this nonuniformity correction technique is demonstrated by applying it to infrared and simulated data. The strength of this technique is in its simplicity, requiring relatively few frames to generate an acceptable correction matrix.

Keywords: nonuniformity correction, focal-plane array, infrared sensors, registration, fixed-pattern noise.

## 1. INTRODUCTION

The performance of focal-plane array (FPA) sensors, particularly those used in infrared imaging systems, is known to be strongly affected by spatial nonuniformity. This spatial nonuniformity is also referred to as fixed-pattern noise, which is attributed to the variation in the photoresponses of individual detectors in the array[1]. Though FPA technology has advanced in recent years, fixed–pattern noise remains a serious issue in that it significantly reduces the quality of acquired images and degrades the temperature resolvability of the imaging system. Moreover, due mainly to external conditions (e.g., surrounding temperature and variation in transistor bias voltage), the spatial nonuniformity tends to drift slowly in time, which demands repeated compensation for the nonuniformity during the course of the sensor's operation.

One commonly used nonuniformity correction (NUC) technique is the two-point calibration, which requires the use of a blackbody radiation source as well as the halting of camera operation for the duration of the calibration procedure. Calibration is necessary in applications where accurate temperature measurement is required. However, some applications do not require such accuracy, but instead require a relative compensation of nonuniformity so that the fixed-pattern noise is removed. As a result, there has recently been a considerable amount of research focused on developing scene-based NUC techniques as an alternative to calibration. These scene-based NUC algorithms normally use an image sequence and rely on global motion between frames as well as diversity in the irradiance seen by each detector. Scene-based NUC algorithms proposed in the literature include Narendra *et al.*[2,3], and more recently Harris *et al.*[4,5] and Chiang *et al.*[6], who developed algorithms that continually compensate for offset and gain nonuniformity by using the concept of constant statistics. O'Neil *et al.*[7,8] and Hardie *et al.*[9], developed NUC techniques which rely on the fact that detectors that record the same scene point at different times should have the same response. Hayat *et al.*[10] and Torres *et al.*[11] developed a statistical algorithm which relies on a key assumption that all detectors in the array are exposed to the same range of collected irradiance within a reasonable number of frames. Torres *et al.*[12,13], recently developed a Kalman-filter technique that also accounts for drift in the nonuniformity parameters.

[*]bratliff@siscom.net; phone 1 937 229-4521; fax 1 937 229-2097; University of Dayton, 300 College Park, Dayton, OH, USA 45469-0226; [**]mhayat@udayton.edu; phone 1 937 229-4521; fax 1 937 229-2097; [***]rhardie@udayton.edu; phone 1 937 229-3178; fax 1 937 229-2097

This paper proposes a new scene-based algorithm that is algebraic in nature. We use an approximate linear irradiance-voltage detector model, where only variations in detector bias nonuniformity are considered. A reliable gradient-based shift estimator is used for estimating motion between image frames. Using this inter-frame motion knowledge, a number of gradient matrices are formed from image frames exhibiting certain subpixel shifts. These gradient matrices can then be used to generate an overall correction matrix that is used to compensate for bias nonuniformity in all frames. Initial findings of this algorithm, using both infrared and simulated data, have shown that it can provide an effective correction matrix using relatively few frames. Due to its algebraic simplicity, the algorithm is easily implemented and is computationally efficient, allowing the correction matrix to be computed quickly, as needed in cases where drift occurs.

This paper is organized as follows. The approximate linear irradiance-voltage sensor model is given in Section 2. In Section 3, development of the algorithm is presented. Performance issues and experimental results obtained from simulated and infrared data are presented in Section 4. Finally, the conclusions are given in Section 5.

## 2. SENSOR MODEL

The coordinate system used in this paper for any image sequence, $\{F_n\}_{n=1}^{P}$, is shown below in Fig. 1.

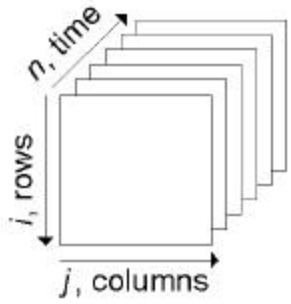


Figure 1: Image sequence coordinate system.

We assume that the *ij*th detector response to irradiance obeys the commonly-used linear model[14],

$$y_n(i, j) = a(i, j)z_n(i, j) + b(i, j), \tag{1}$$

where $y_n(i,j)$ is the observed output, $z_n(i,j)$ is the true scene irradiance, $a(i,j)$ is the detector's gain and $b(i,j)$ is the detector's offset (bias). We assume uniform gain across all detectors, and without loss of generality, we assume a common gain value of unity. Given these assumptions, the above equation becomes,

$$y_n(i, j) = z_n(i, j) + b(i, j). \tag{2}$$

Moreover, we assume that the temperature of the observed objects does not change during the inter-frame time interval. This assumption allows us to model scene values of the $(n+1)$th frame as an interpolation of two adjacent scene values from the $n$th frame. For simplicity and convenience, we use linear interpolation as an approximation. Therefore, for a pair of consecutive frames, $k$ and $k+1$, exhibiting a purely *vertical* sub-pixel motion $a$, the pixel values of the $k$th and the $(k+1)$th frames are related by

$$y_{k+1}(i+1, j) = a\, z_k(i, j) + (1-a)z_k(i+1, j) + b(i+1, j), \quad 0 \le a \le 1. \tag{3}$$

Note that a positive $a$ represents downward motion. Similarly, for a pair of adjacent frames, $m$ and $m+1$, with a purely *horizontal* sub-pixel motion $\hat{a}$, the pixel values of the $m$th and the $(m+1)$th frames are related by

$$y_{m+1}(i, j+1) = \beta\, z_m(i, j) + (1-\beta) z_m(i, j+1) + b(i, j+1), \qquad 0 \le \beta \le 1. \tag{4}$$

In the next Section, we exploit the relationships given by (3) and (4) to algebraically extract a nonuniformity map.

## 3. ALGORITHM DESCRIPTION

### 3.1 Correction Algorithm

In this subsection, we present the development of the proposed algorithm. We begin by assuming that we have two collections, $\mathbf{C}_\alpha$ and $\mathbf{C}_\beta$, of adjacent frame pairs exhibiting vertical and horizontal subpixel motion, respectively. For example, each element in $\mathbf{C}_\alpha$ is a pair of consecutive frames exhibiting a purely vertical shift. For simplicity and without loss of generality, we assume that each shift associated with each frame pair in both $\mathbf{C}_\alpha$ and $\mathbf{C}_\beta$ is positive. Furthermore, we assume that $\mathbf{C}_\alpha$ contains $A$ shift pairs and $\mathbf{C}_\beta$ contains $B$ shift pairs, where $A>B$.

We now start the description of the algorithm. For a $p$th pair of frames in $\mathbf{C}_\alpha$, where $p = 1, 2, \ldots, A$, we compute a partial vertical correction matrix, $\mathbf{V}_p$.

#### 3.1.1 Partial Vertical Correction Matrix

Each $\mathbf{V}_p$ is computed in two stages. The first step involves generating an intermediate matrix $\tilde{\mathbf{V}}_p$, and is computed by performing a scaled difference between elements of the $k$th and $(k+1)$th frames. More precisely, each element of $\tilde{\mathbf{V}}_p$ is computed for each pair of frames in $\mathbf{C}_\alpha$ as follows:

$$\tilde{\mathbf{V}}_p(1, j) = 0, \tag{5}$$

where $j = 1, 2, \ldots, N$, and

$$\tilde{\mathbf{V}}_p(i, j) = \frac{1}{\alpha}\left[\alpha\ y_k(i-1, j) + (1-\alpha) y_k(i, j) - y_{k+1}(i, j)\right], \tag{6}$$

for $i = 2, 3, \ldots, M$, and $j = 1, 2, \ldots, N$. Here, $M$x$N$ is the dimension of each frame. Notice that (6) involves combining two scaled, adjacent pixels from the $k$th frame and subtracting a pixel from the $(k+1)$th frame. This specific choice of scaling constants results in the cancellation of the scene values $z_k(i\text{-}1,j)$ and $z_k(i,j)$, leaving a difference of only the bias terms. In particular, by substituting for $y_k(i\text{-}1,j)$, $y_k(i,j)$ and $y_{k+1}(i,j)$ in (6), we obtain

$$\begin{aligned}\tilde{\mathbf{V}}_p(i, j) &= \frac{1}{\alpha}\left[\alpha\ z_p(i-1, j) + \alpha\ b(i-1, j) + (1-\alpha) z_p(i, j) + (1-\alpha) b(i, j) - \alpha\, z_p(i-1, j) - (1-\alpha) z_p(i, j) - b(i, j)\right], \\ &= b(i-1, j) - b(i, j), \end{aligned} \tag{7}$$

for $i = 2, 3, \ldots, M$, and $j = 1, 2, \ldots, N$. In matrix form, we have

$$\tilde{\mathbf{V}}_p = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ b(1,1) - b(2,1) & b(1,2) - b(2,2) & \cdots & b(1, N) - b(2, N) \\ b(2,1) - b(3,1) & b(2,2) - b(3,2) & \cdots & b(2, N) - b(3, N) \\ \vdots & \vdots & \ddots & \vdots \\ b(M-1,1) - b(M,1) & b(M-1,2) - b(M,2) & \cdots & b(M-1, N) - b(M, N) \end{bmatrix}. \tag{8}$$

The second step, resulting in $\mathbf{V}_p$, is achieved by taking a cumulative sum down each column of $\tilde{\mathbf{V}}_p$. More precisely, for $i = 2, 3, \ldots, M$, and $j = 1, 2, \ldots, N$, define the $ij$th entry of $\mathbf{V}_p$ by

$$\mathbf{V}_p(i, j) = \sum_{c=2}^{i} \tilde{\mathbf{V}}_p(c, j) . \tag{9}$$

In matrix form, the resulting $\mathbf{V}_p$ is

$$\mathbf{V}_p = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ b(1,1) - b(2,1) & b(1,2) - b(2,2) & \cdots & b(1,N) - b(2,N) \\ b(1,1) - b(3,1) & b(1,2) - b(3,2) & \cdots & b(1,N) - b(3,N) \\ \vdots & \vdots & \ddots & \vdots \\ b(1,1) - b(M,1) & b(1,2) - b(M,2) & \cdots & b(1,N) - b(M,N) \end{bmatrix}. \tag{10}$$

Observe now that if $\mathbf{V}_p$ is added to any frame $\mathbf{F}_n$, then

$$\mathbf{F}_n + \mathbf{V}_p = \begin{bmatrix} z(1,1) + b(1,1) & z(1,2) + b(1,2) & \cdots & z(1,N) + b(1,N) \\ z(2,1) + b(1,1) & z(2,2) + b(1,2) & \cdots & z(2,N) + b(1,N) \\ \vdots & \vdots & \ddots & \vdots \\ z(M,1) + b(1,1) & z(M,2) + b(1,2) & \cdots & z(M,N) + b(1,N) \end{bmatrix}. \tag{11}$$

Notice that in the partially corrected frame of (11), each column's bias has effectively been "synchronized" to the bias of the column's first element, or "leader." In particular, the $j$th column of each partially corrected frame will have a bias of $b(1,j)$, regardless of the row value.

Recall that $\mathbf{V}_p$ is computed for each pair of frames in $\mathbf{C}_a$. Moreover, it is clear from (10) that ideally, all $\mathbf{V}_p$'s are identical, but in practice there is an error term associated with each $ij$th value (due to aliasing, shift estimation error, etc.). To minimize this error, all $\mathbf{V}_p$'s are averaged together, resulting in an averaged partial vertical correction matrix given by

$$\overline{\mathbf{V}} = \frac{1}{A} \sum_{p=1}^{A} \mathbf{V}_p . \tag{12}$$

It is this $\grave{\mathbf{V}}$ that will, in part, be used to generate the total correction matrix.

### 3.1.2 Partial Horizontal Correction Matrix

Next, we take $\grave{\mathbf{V}}$ and add it to every frame in $\mathbf{C}_b$. This gives a collection of partially-corrected, horizontally shifted frame pairs, $\mathbf{C}_r$. It is this $\mathbf{C}_r$, *not* $\mathbf{C}_b$, that is used to generate the partial horizontal correction matrix. This partial correction step is crucial, as it will ultimately allow for all bias values to be synchronized to a common value.

Now, a partial horizontal correction matrix, $\mathbf{H}_q$, where $q = 1, 2, \ldots, B$, is computed for each partially-corrected pair of frames in $\mathbf{C}_r$. For a pair of frames $m$ and $m$+1, computing $\mathbf{H}_q$ is very similar to computing $\mathbf{V}_p$. First, an $\tilde{\mathbf{H}}_q$ is computed as follows:

$$\tilde{\mathbf{H}}_q(i,1) = 0, \tag{13}$$

where $i = 1, 2, \ldots, M$, and

$$\tilde{\mathbf{H}}_q(i,j) = \frac{1}{b}\left[b\ y_m(i,j-1) + (1-b)y_m(i,j) - y_{m+1}(i,j)\right], \tag{14}$$

for $i = 1, 2, \ldots, M$, and $j = 2, 3, \ldots, N$. By substituting for $y_m(i,j\text{-}1)$, $y_m(i,j)$ and $y_{m+1}(i,j)$, canceling terms and expanding (14), we obtain

$$\tilde{\mathbf{H}}_q(i,j) = b(i,j-1) - b(i,j), \tag{15}$$

for $i = 1, 2, \ldots, M$, and $j = 2, 3, \ldots, N$. The final result of computing $\tilde{\mathbf{H}}_q$ is given below in matrix form as

$$\tilde{\mathbf{H}}_q = \begin{bmatrix} 0 & b(1,1)-b(1,2) & b(1,2)-b(1,3) & \cdots & b(1,N-1)-b(1,N) \\ 0 & b(1,1)-b(1,2) & b(1,2)-b(1,3) & \cdots & b(1,N-1)-b(1,N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & b(1,1)-b(1,2) & b(1,2)-b(1,3) & \cdots & b(1,N-1)-b(1,N) \end{bmatrix}. \tag{16}$$

$\mathbf{H}_q$ is then computed by taking a cumulative sum across each row of $\tilde{\mathbf{H}}_q$,

$$\mathbf{H}_q(i,j) = \sum_{r=2}^{j} \tilde{\mathbf{H}}_q(i,r), \tag{17}$$

for $i = 1, 2, 3, \ldots, M$, and $j = 2, 3, \ldots, N$. More compactly, we have

$$\mathbf{H}_q = \begin{bmatrix} 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \\ 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \end{bmatrix}. \tag{18}$$

Adding $\mathbf{H}_q$ to any *partially corrected* frame will synchronize the bias values across each column, ultimately unifying all biases to a common value. Again, observing that each $\mathbf{H}_q$ is identical, the *ij*th term of each $\mathbf{H}_q$ is averaged together to reduce error, resulting in the averaged partial horizontal correction matrix,

$$\hat{\mathbf{H}} = \frac{1}{B}\sum_{q=1}^{B}\mathbf{H}_q = \begin{bmatrix} 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \\ 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \end{bmatrix}. \tag{19}$$

Observe that the elements of each column in $\hat{\mathbf{H}}$ are ideally identical (assuming that the shift estimates are accurate and the model of (3) and (4) is valid). As a way to further minimize error, a second average is taken down each column of $\hat{\mathbf{H}}$ to produce an averaged, horizontal partial row vector $\vec{\mathbf{H}}$. The values of $\vec{\mathbf{H}}$ can then be duplicated down each column *M* times to produce the final *M*x*N* averaged partial horizontal correction matrix, $\overline{\mathbf{H}}$.

### 3.1.3 Total Correction Matrix

The final correction matrix, **C**, is then obtained simply by summing $\overline{\mathbf{V}}$ and $\overline{\mathbf{H}}$, giving

$$\mathbf{C} = \overline{\mathbf{V}} + \overline{\mathbf{H}} = \begin{bmatrix} 0 & b(1,1)-b(1,2) & b(1,1)-b(1,3) & \cdots & b(1,1)-b(1,N) \\ b(1,1)-b(2,1) & b(1,1)-b(2,2) & b(1,1)-b(2,3) & \cdots & b(1,1)-b(2,N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b(1,1)-b(M,1) & b(1,1)-b(M,2) & b(1,1)-b(M,3) & \cdots & b(1,1)-b(M,N) \end{bmatrix}. \tag{20}$$

For correction, $\mathbf{C}$ is added to each frame, $\mathbf{F}_n$, of the original image sequence, giving

$$\mathbf{F}_n + \mathbf{C} = \begin{bmatrix} z(1,1)+b(1,1) & z(1,2)+b(1,1) & \cdots & z(1,N)+b(1,1) \\ z(2,1)+b(1,1) & z(2,2)+b(1,1) & \cdots & z(2,N)+b(1,1) \\ \vdots & \vdots & \ddots & \vdots \\ z(M,1)+b(1,1) & z(3,2)+b(1,1) & \cdots & z(M,N)+b(1,1) \end{bmatrix}. \tag{21}$$

Observe that the bias terms in the corrected image frame of (21) have effectively been unified to $b(1,1)$. Hence, we have made the bias terms uniform across all detectors. A block diagram of the presented NUC algorithm is shown below in Fig. 2.

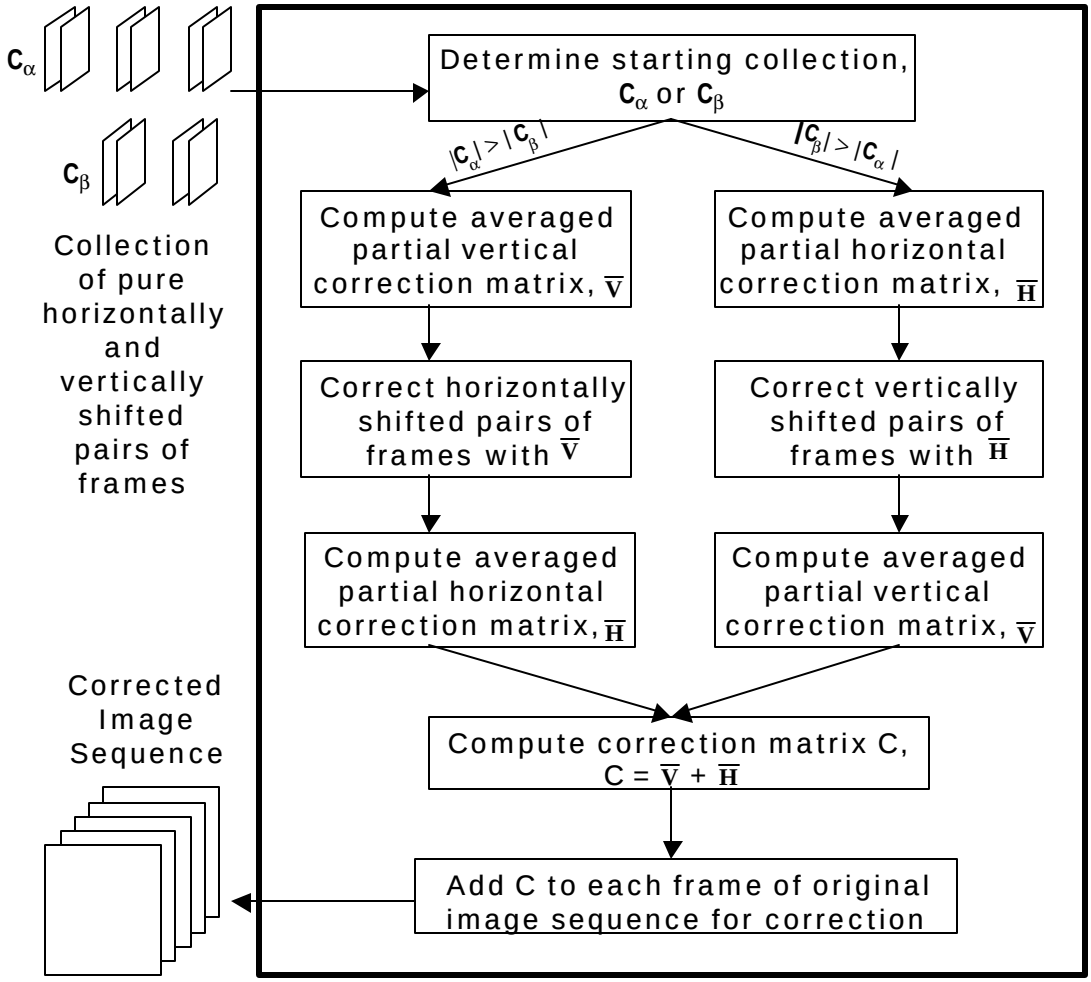

Figure 2: Block diagram of the proposed NUC algorithm.

### 3.2 Shift Estimation

We now describe how the collections $\mathbf{C}_a$ and $\mathbf{C}_b$ are generated by utilizing a shift estimation algorithm. In the shift estimation process, the global shifts between adjacent frames of the original image sequence are estimated. This is accomplished by using the gradient-based shift estimation algorithm reported by Hardie *et al.*[9]. However, Armstrong *et al.*[15] showed that spatial nonuniformity introduces significant error in the shift estimates produced by the gradient-based algorithm. To minimize this error, we first pre-filter the original image sequence with an *R*x*R* smoothing filter. Through smoothing, the nonuniformity is reduced, allowing for improved shift estimation. It is this smoothed image sequence that is used in the gradient-based algorithm. The resulting shift estimates are then analyzed for acceptable sub-pixel, purely horizontal and vertical shifts. We define acceptable sub-pixel shifts as those being in the interval [–1,1] for both pure

horizontal and pure vertical motion. Furthermore, we define a tolerance parameter, $e$, which defines how closely we want the orthogonal motion to be to zero. A block diagram of the shift estimation process is displayed in Fig. 3.

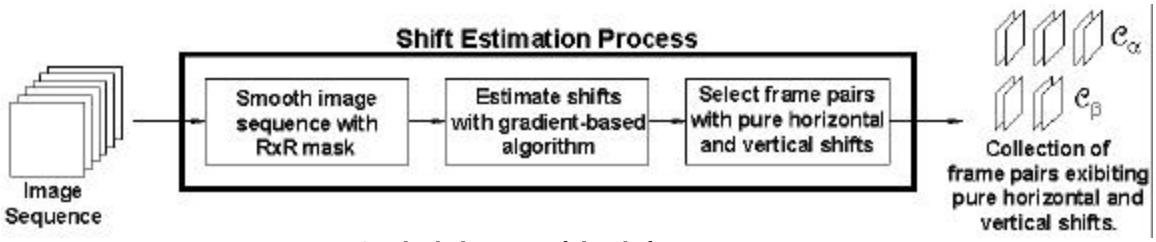


Figure 3: Block diagram of the shift estimation process.

## 4. APPLICATION

### 4.1 Key Performance Issues

In this subsection, the performance of the proposed correction algorithm is studied using created image sequences corrupted by simulated nonuniformity. Three main areas are studied, addressing issues with the correction algorithm, the shift estimation algorithm, and aliasing.

#### 4.1.1 Correction Algorithm Issues

Three important issues that are used to improve robustness of the correction algorithm are: (1) the starting collection (i.e. $\mathbf{C}_a$ and $\mathbf{C}_b$), (2) appropriately handling positive and negative shifts, and (3) orthogonal motion tolerance, $e$.

After shift estimation, two collections of frame pairs are obtained, $\mathbf{C}_a$ and $\mathbf{C}_b$,. The algorithm must determine the best starting collection. To make this decision, the algorithm simply counts the number of frame pairs in each collection and begins using the set having the most pairs. The reason for such a selection is due to the fact that all the spatial nonuniformity is encountered when computing the first averaged partial correction matrix. By taking the larger collection, more partial correction matrices are averaged together, minimizing the error in each pixel value. After the first averaged partial correction matrix is formed, a good estimate of the nonuniformity is obtained, but each column has been synchronized to its respective leader's bias value. The second collection is then used to accurately synchronize the bias of each column to a value common to all pixels.

When each pair of frames in the collections $\mathbf{C}_a$ and $\mathbf{C}_b$ is used, its respective shift may be either positive or negative. For each partial correction matrix, the shift polarity determines the starting row or column. As an example, using the coordinate system of Fig. 1, if an $M$x$N$ frame pair exhibiting positive horizontal motion is used, the algorithm would begin in column 1 and perform its computations in a rightward fashion. If the frame pair exhibits negative horizontal motion, it would instead begin with column $N$ and perform its computations in a leftward fashion. Similar cases occur with vertical motion. By considering these special cases, the algorithm is able to find and use more acceptable pairs of frames in accordance to the sign of the shift.

Another parameter that has an affect on the performance of the proposed algorithm is the orthogonal-motion tolerance, $e$. This parameter determines the amount of motion that will be allowed in the "zero" or no-motion direction. Ideally, we would like this parameter to be zero, but in practice we have to allow a small amount of deviation in order to get a reasonable amount of acceptable shift pairs. For all the data tested, sufficient shift pairs were found when this parameter was set to 0.05. This parameter was increased to see the affect of introducing substantial motion in the orthogonal direction. No major performance decrease in algorithm correction was noticed until the parameter was increased beyond 0.2. Once the tolerance parameter becomes too large, a striping effect across rows and/or columns was observed in the corrected image sequence.

### 4.1.2 Shift Estimation Issues

The gradient-based shift estimation algorithm has demonstrated a high level of accuracy on image sequences with little or no spatial nonuniformity. As the amount of nonuniformity is increased, so does error in the shift estimates. Estimation error, when greater than about five percent (of a pixel), begins to affect performance of the correction algorithm. As a result of this shift error, the corrected image sequence begins to show striping artifacts that are present across all corrected image frames.

To reduce this estimation error, each frame of the input image sequence is filtered with an $R$x$R$ smoothing mask prior to registration. Spatial nonuniformity is high-frequency noise; smoothing effectively removes this nonuniformity since it is a low-pass filtering operation. Shift estimation of the smoothed image sequence with an appropriate mask results in a shift error that is around one percent or less. By an "appropriate" mask, we imply choosing the best value for $R$. For all real and simulated data tested, it was found that $R$ values of 3 and 5 appeared to work best. In practice, however, the best $R$ value will depend on the strength of the nonuniformity.

### 4.1.3 Aliased Data

The performance of the correction algorithm is affected by an aliased image sequence when too few acceptable shift pairs are available. Aliasing occurs as a result of undersampling, and is a common problem due to finite detector size. As aliasing is increased in an image sequence, the validity of our linear interpolation approximation decreases, introducing error into each pixel value. In fact, it is observed that higher aliasing requires more shift pairs to achieve an acceptable correction matrix. Thus, when adequate shift pairs are available, aliasing can be overcome through averaging enough partial correction matrices together. As will be shown in Section 4.2, simulated aliased image sequences are successfully corrected using 20 $\alpha$- and 19 $\beta$-shift pairs.

## 4.2 Results from Simulated Data

Two types of simulated image sequences are created. The first are sequences where each frame is constructed using linear interpolation. This type of data is ideal for the proposed algorithm because our model relies on the assumption that scene values in each ($n$+1)th frame are a linear interpolation of two adjacent scene values from the $n$th frame. Since this data is synthetically generated, the shift between image frames can be precisely controlled.

Indeed, when the ideal image sequence is fed into the algorithm with no added nonuniformity, each element of the correction matrix is equal to zero, as is expected. Moreover, when the same $M$x$N$ matrix of zero-mean Gaussian noise (with standard deviation of 20) is added to each image frame (simulating spatial nonuniformity), the generated $\mathbf{C}$ matrix corrects the image sequence with no error, thereby demonstrating the algorithm's ideal performance. Notably, using ideal data, a perfect correction matrix can be generated using only four images, one horizontally shifted frame pair and one vertically shifted frame pair.

The second type of image sequences are generated by down-sampling a high-resolution image. Through down-sampling we can again precisely control the shifts between image frames. All image sequences created are down-sampled by either a factor of 10 or 20. In either case, the sequences produced are highly aliased. Zero-mean Gaussian noise (with standard deviation of 20) is added to each frame of the down-sampled image sequence. The sequence is blurred with a 5x5 smoothing filter in the shift estimation process. Found and used are 20 acceptable $a$–shift pairs and 19 acceptable $\hat{a}$ –shift pairs. Despite aliasing, the results obtained are very good. Figures 4 and 5 show frame 1 from a 10-factor down-sampled 128x128 by 40-frame data sequence before and after adding the simulated nonuniformity. Figure 6 displays the image frame after correction. The correction matrix generated and used by the algorithm is displayed (scaled to 256 dynamic levels) in Fig. 7.

## 4.3 Results from Real Infrared Data

Real infrared image sequences were used to test the proposed algorithm. The data sets were collected using a 128x128 InSb FPA camera (Amber Model AE-4128) operating in the 3 ~ 5 μm range. For all real data, a 3x3 mask is used to smooth the image sequences during the shift estimation process. Using three different real infrared data sequences, it is found that 75-100 shift pairs in the first direction and 10-20 shift pairs in the second direction produce an effective correction matrix. In the first data set (consisting of 400 frames),

Figure 4: Frame 1 from the down-sampled image sequence prior to the added nonuniformity.

Figure 5: Frame 1 from the down-sampled image sequence after the added nonuniformity.

Figure 6: Frame 1 from the corrected down-sampled image sequence.

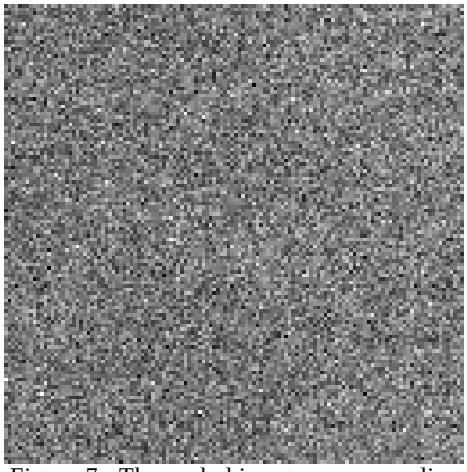

Figure 7: The scaled image corresponding to the correction matrix, **C**.

there were 179 $a$–shift pairs and 15 $\hat{a}$ –shift pairs found and used by the algorithm. A tolerance parameter, $e$, of 0.05 was used, meaning that the orthogonal motion shifts were in the interval of [-0.05, 0.05]. Figure 8 displays frame 1 of the 400-frame infrared image sequence. Frame 1 of the infrared image sequence after correction is shown in Fig. 9. The correction matrix (scaled to 256 dynamic levels) is displayed in Fig. 10. The results obtained from this first infrared data set are generally very good. By examining the original image frame of Fig. 8, we notice the crosshatch pattern of the spatial nonuniformity. This same crosshatch pattern is observed in the correction matrix **C** of Fig. 10; this shows that the algorithm has correctly found the bias nonuniformity. In the corrected image frame of Fig. 9, it is apparent that the spatial nonuniformity has effectively been removed.

Correction results from a second 500-frame 128x128 real infrared image sequence, collected at a later time using the same camera, are also displayed. Figures 11 and 12 show frame 1 of the infrared sequence before and after correction. Figure 13 displays the corresponding correction matrix (scaled to 256 dynamic levels). For this correction, 139 $a$–shift pairs and 47 $\hat{a}$ –shift pairs are found and used by the algorithm. A value of 0.05 was used for $e$. For the second infrared image sequence, satisfactory results are obtained. One difference can be observed by examining the bottom of Fig. 13. Notice the brighter region and the slight amount of striping present within it. In the image sequence, foliage is present at the bottom of some frames. All acceptable horizontally shifted frame pairs contain this foliage. As a result, some foliage information is present in the correction matrix. Therefore, more horizontally shifted frame pairs may be required in order to reduce this scene information. Even so, the corrected frame of Fig. 12 is still good, as the striping is not very profound.

Figure 8: Frame 1 from the infrared data set 1.

Figure 9: Frame 1 from the corrected infrared data set 1.

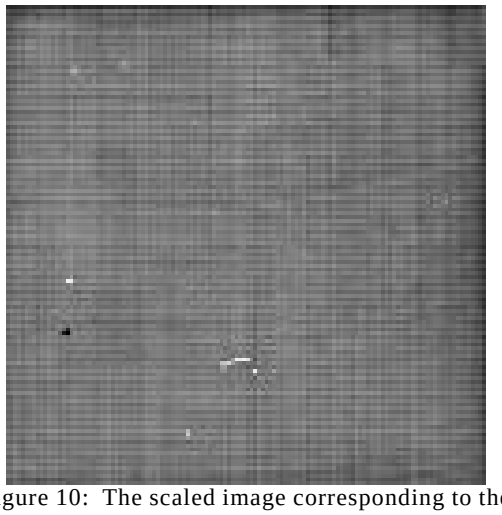

Figure 10: The scaled image corresponding to the correction matrix, **C,** for the infrared data set 1.

## 5. CONCLUSIONS

We have presented an algebraic scene-based technique for nonuniformity compensation in the detector offsets of a focal-plane array. It has been shown that a useful correction map can be obtained using relatively few image frames. The strength of the proposed algorithm is in its simple algebraic nature, allowing the correction map to be computed quickly. The efficacy of this technique was tested using both simulated and real infrared data.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (Career Program MIP-9733308). The authors wish to thank Ernest E. Armstrong at the U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base for assistance in collecting data.

Figure 11: Frame 1 from the infrared data set 2.

Figure 12: Frame 1 from the corrected infrared data set 2.

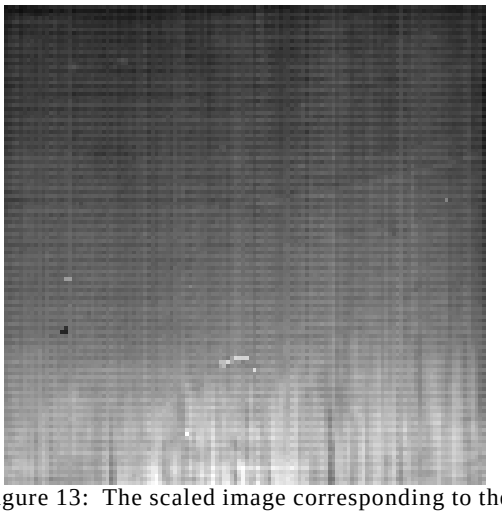

Figure 13: The scaled image corresponding to the correction matrix, **C**, from the infrared data set 2.

## REFERENCES


1. A. F. Milton, F. R. Barone, and M.R. Kruer, "Influence of nonuniformity on infrared focal plane array performance," *Optical Engineering*, **vol. 24**, pp. 855-862, 1985.

2. P. M. Narendra and N.A. Foss, "Shutterless fixed pattern nois e correction for infrared imaging arrays," *Proc. SPIE, Technical issues in focal plane development*, **vol. 282**, pp. 44-51, 1981.

3. P. M. Narendra, "Reference-free nonuniformity compensation for IR imaging arrays," *Proc. SPIE (Smart Sensors II)*, **vol. 252**, pp. 10-17, 1980.

4. J. G. Harris, "Continuous-time calibration of VLSI sensors for gain and offset variations," *SPIE Int. Symp. on Aerospace and Dual-Use Photonics: Smart Focal Plane Arrays and Focal Plane Array Testing*, **vol. 2474**, pp. 23-33, 1995.

5. J. G. Harris and Y. M. Chiang, "Nonuniformity correction using constant average statistics constraint: Analog and digital implementations," *Proc. SPIE*, **vol. 3061**, pp. 895-905, 1997.

6. Y. M. Chiang and J. G. Harris, “An analog integrated circuit for continuous-time gain and offset calibration of sensor arrays,” *J. Analog Integrated Circuits and Signal Processing*, 1996.

7. W. F. O’Neil, “Dithered scan detector compensation,” *Proc. IRIS*, 1993.

8. W. F. O’Neil, “Experimental verification of dithered scan nonuniformity correction,” *Proc. IRIS*, 1997.

9. R. C. Hardie, M. M. Hayat and E. E. Armstrong and B. Yasuda, “Scene-based nonuniformity correction using video sequences and registration,” *OSA Applied Optics*, **vol. 39**, pp. 1241-1250, 2000.

10. M. M. Hayat, S. N. Torres, E. E. Armstrong and B. Yasuda, “Statistical algorithm for non-uniformity correction in focal-plane arrays,” *Applied Optics*, **vol. 38**, pp. 772-780, 1999.

11. S. N. Torres, M. M. Hayat, E. E. Armstrong and B. Yasuda, “On the performance analysis of a recent statistical algorithm for nonuniformity correction in focal-plane arrays,” *Proc. CISST*, **vol. 3703**, 2000.

12. S. N. Torres, M. M. Hayat, E. E. Armstrong, and B. Yasuda, “A Kalman-filtering approach for non-uniformity correction in focal-plane array sensors,” *Proc. SPIE AeroSense 2000, Infrared Imaging Systems: Design, Analysis, Modeling, and Testing*, **vol. 4030**, pp. 196-203, 2000.

13. S. N. Torres and M. M. Hayat, “Compensation for gain and bias nonuniformity and drift in array detectors: A Kalman-filtering approach," *submitted to IEEE Trans. Image Processing*, Jan. 2000, under review.

14. G. C. Holst, *CCD Arrays, Cameras and Displays*, SPIE Optical Engineering Press, Bellingham, 1996.

15. E. E. Armstrong, M. M. Hayat, R. C. Hardie, S. N. Torres, and B. Yasuda, “Non-uniformity correction for improved registration and high-resolution image reconstruction in IR imagery,” *Proc. SPIE, Application of digital image Processing XXII*, **vol**. **3808**, pp. 150-161, 1999.