Gain Fixed Pattern Noise Correction via Optical Flow

SukHwan Lim, *Member, IEEE,* and Abbas El Gamal, *Fellow, IEEE*

***Abstract*—The paper describes a method that uses a video sequence to correct gain fixed pattern noise (FPN) in an image sensor. The captured sequence and its optical flow are used to estimate gain FPN. Assuming brightness constancy along the motion trajectories, the pixels are grouped in blocks and the gains of the pixels in each block are estimated by iteratively minimizing the sum of the squared brightness variations along the motion trajectories. Significant reductions in gain FPN are demonstrated using both real and synthetically generated video sequences with modest computations.**

***Index Terms*—CMOS image sensor, correlated double sampling (CDS), digital camera, fixed pattern noise (FPN), gain FPN, optical flow estimation.**

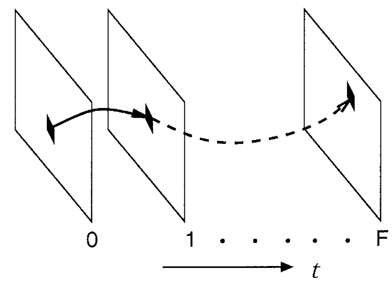


Fig. 1. Sequence of frames with a moving patch of constant brightness.

I. INTRODUCTION

FIXED pattern noise (FPN) or nonuniformity is the spatial variation in output pixel values under uniform illumination due to device mismatches and process parameter variations across an image sensor. It is a major source of image quality degradation especially in CMOS image sensors [1], [2]. In a charged couple devices (CCD) sensor, since all pixels share the same output amplifier, the FPN is mainly due to variations in photodetector area and dark current. In a CMOS image sensor, however, pixels are read out over different chains of buffers and amplifiers each with different gain and offset, resulting in relatively high FPN. While offset FPN can be significantly reduced using correlated double sampling (CDS), no method exists for effectively reducing the gain FPN. In [3], a method is proposed for reducing gain FPN by characterizing the sensor's pixel gains after manufacture, and storing the gains in a lookup table that is subsequently used to perform the correction. A problem with this method is that gain FPN changes with temperature and aging, making a "static" gain lookup table approach inaccurate. Another method would be to characterize the sensor's pixel gains before each capture. This is not feasible since characterizing gain FPN requires many captures at different uniform illuminations.

In this paper, we describe a method to estimate and correct gain FPN using a video sequence and its optical flow. The method can be used in a digital video or still camera by capturing a video sequence with motion prior to capturing the desired images and using it to estimate gain FPN. The needed motion can be caused by panning the camera during capture.

In the following section, we describe the image and FPN model used throughout the paper. In Section III, we describe our algorithm for estimating and correcting gain FPN and illustrate its operation via simple one-dimensional (1-D) examples. In Section IV, we show simulation results using a synthetically generated sequence and its optical flow. We then show experimental results using a real video sequence taken with our experimental imaging system [4].

II. IMAGE AND FPN MODEL

Assuming a linear sensor transfer function, the pixel intensity value i as a function of its input signal s, e.g., photocurrent density [5], can be expressed as

$$i = hs + i_{\mathrm{os}} \tag{1}$$

where h is the gain factor and i_{os} is the offset, which includes the dark signal as well as the offset due to the amplifiers and buffers. Offset FPN is due to pixel to pixel variations in i_{os} and can be significantly reduced by CDS. Gain FPN is caused by variations in the gain factor h and there is no existing method for effectively reducing it.

In this paper, we only consider gain FPN and assume that offset FPN has been canceled with CDS. After eliminating the offset term in (1) and including gain variations, we obtain

$$\begin{aligned} i(x,y,t) &= i_0(x,y,t) + \Delta i(x,y,t) \\ &= (h_0 + \Delta h(x,y))s(x,y,t) \\ &= \left(1 + \frac{\Delta h(x,y)}{h_0}\right) i_0(x,y,t) \\ &= a(x,y) i_0(x,y,t) \end{aligned}$$

where $i_0(x,y,t)$ is the ideal intensity value at pixel (x,y) and time (frame) t, h_0 is the nominal gain factor, and $\Delta h(x,y)$ is the deviation in gain for pixel (x,y). Gain FPN can be represented as the pixel to pixel variation of $a(x,y)$ and its magnitude is $\sigma_{\Delta h}/h_0$. Although gain FPN can slowly vary with temperature

Manuscript received June 9, 2002; revised April 8, 2003. This work was supported by Agilent, Canon, HP, Interval Research, and Kodak under the Programmable Digital Camera Program. This paper was recommended by Associate Editor O. Au.

S. Lim was with Stanford University, Stanford, CA 94305 USA. He is now with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: sukhwan@hpl.hp.com).

A. El Gamal is with Stanford University, Stanford, CA 94305 USA (e-mail: abbas@isl.stanford.edu).

Digital Object Identifier 10.1109/TCSI.2004.823666

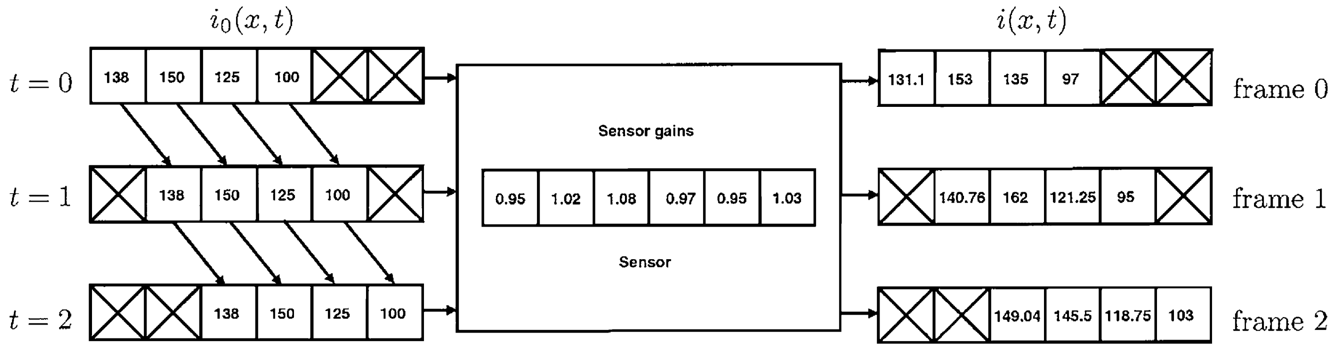


Fig. 2. One-dimensional case simple example when the displacements are integers.

and aging, we assume here that $a(x,y)$ is constant while capturing several frames with the imager. Note that $a(x,y)=1$ for all (x,y) in an ideal sensor having no gain FPN.

We assume brightness constancy, i.e., brightness is constant along each motion trajectory, as is commonly assumed in the development of many video processing and computer vision algorithms [6], [7]. Thus, if $F+1$ frames are captured using an $M \times N$ pixel image sensor, the ideal pixel intensity value at $t, i_0(x,y,t)$, can be expressed in terms of the ideal pixel intensity at $t=0, j(x,y)=i_0(x,y,0)$, as

$$i_0(x+d_x(x,y,t), y+d_y(x,y,t), t) = j(x,y) \tag{2}$$

for $x=1,\ldots,M, y=1,\ldots,N$, and $t=0,\ldots,F$, where $d_x(x,y,t)$ and $d_y(x,y,t)$ are the displacements (optical flow) between frame 0 and t for pixel (x,y) in frame 0. Note that by definition $d_x(x,y,0)=d_y(x,y,0)=0$. This model is illustrated in Fig. 1, which depicts the pixel locations of a moving patch of constant intensity in each frame. Under this ideal model, the pixel output values along the motion trajectory (within the patch) in all frames are equal.

When gain FPN and temporal noise are added to the ideal model, the pixel intensity value $i(x,y,t)$ becomes

$$\begin{aligned} i(x+d_x, y+d_y, t) = {} & a(x+d_x, y+d_y) j(x,y) \\ & + N(x+d_x, y+d_y, t) \end{aligned} \tag{3}$$

where $N(x,y,t)$ is the additive temporal noise for pixel (x,y) at time t. For notational simplicity, we omitted the index (x,y,t) in d_x and d_y. Note that although the gain FPN component $a(x,y)$ is constant over time $t, a(x+d_x, y+d_y)$ varies with time t. Thus, in the example in Fig. 1, the pixel values along the motion trajectory (within the patch) would be different. However, note that if we ignore temporal noise, the ratio of the pixel output values along the motion trajectory (within the patch) equals the ratio of the gains at those tracked pixel locations. These ratios can then be used to correct for gain FPN.

III. Gain FPN Correction Algorithm

In this section, we describe the gain FPN correction algorithm for integer and noninteger displacement case. The goal here is to estimate $j(x,y)$ from $i(x,y,0),\ldots,i(x,y,F)$ in the presence of temporal noise and gain FPN. To do so, we formulate the problem as follows. Let $\hat{j}(x,y\,|\,t)$ be a linear estimate of $j(x,y)$ obtained from $i(x,y,t)$ of the form

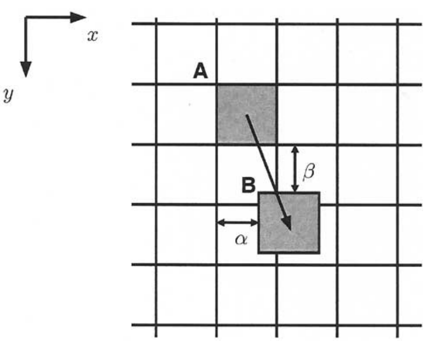


Fig. 3. Example of a pixel-size patch with noninteger displacements.

$$\hat{j}(x,y\,|\,t) = k(x+d_x, y+d_y) i(x+d_x, y+d_y, t) \tag{4}$$

where $k(x,y)$ is a coefficient function that we need to estimate. Because of the brightness constancy assumption, $j(x,y)$ is constant over time, and hence $\hat{j}(x,y\,|\,t)$ does not depend on time. Using this fact, we find $k(x,y)$ that minimizes the mean squared error (MSE) between $\hat{j}(x,y\,|\,0)$ and $\hat{j}(x,y\,|\,t)$. To reduce the computational complexity of estimating $k(x,y)$, we divide the image into nonoverlapping blocks and independently estimate $k(x,y)$ for each block. Thus, to estimate $k(x,y)$ for pixels in block B, we minimize the MSE function

$$\begin{aligned} E_B &= \sum_{t=1}^{F} \sum_{(x,y)\in B} \left(\hat{j}(x,y\,|\,0) - \hat{j}(x,y\,|\,t)\right)^2 \\ &= \sum_{t=1}^{F} \sum_{(x,y)\in B} (k(x,y) i(x,y,0) \\ &\quad - k(x+d_x, y+d_y) i(x+d_x, y+d_y, t))^2. \end{aligned} \tag{5}$$

In the following section, we describe how the estimate is found for the case when the displacements are integer valued. In Section III.B, we extend the discussion to the noninteger case.

A. Integer Displacements

Let R be the set of pixel locations $(x+d_x, y+d_y)$ along the motion trajectories for $(x,y) \in B$, and n_B and n_R be the number of pixels in B and R, respectively. We define the

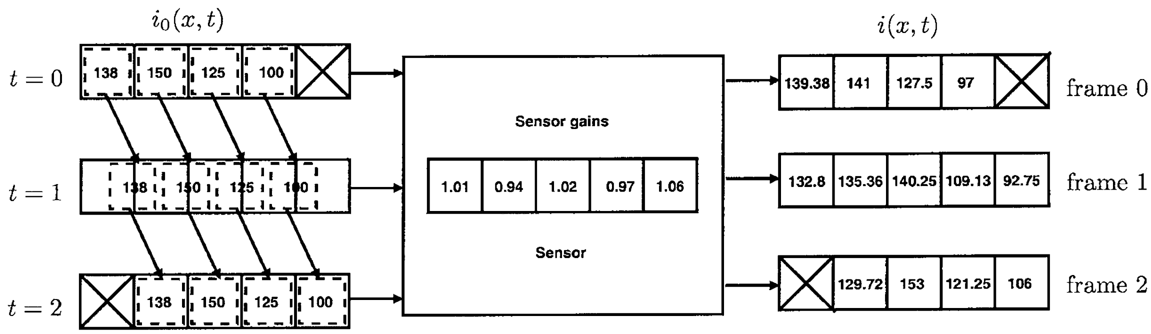


Fig. 4. One-dimensional case simple example when the displacements are nonintegers.

n_R-vector $\mathbf{k}$ to consist of the elements $k(x,y)$ in R beginning with the elements in the block B. Warping $k(x,y)$ to form $k(x+d_x, y+d_y)$ can be represented by multiplying the vector $\mathbf{k}$ with an $n_B \times n_R$ matrix $T(t)$, which is formed as follows. When brightness at the pixel location m in frame 0 moves to pixel location n in frame t, the mth row of $T(t)$ is assigned a 1 to its nth element and 0 to all its other elements. Let $I(t)$ be the $n_B \times n_B$ diagonal matrix whose diagonal elements are $i(x+d_x, y+d_y, t)$ for $(x,y) \in B$. We can now rewrite (5) in a matrix form as

$$E_B = \sum_{t=1}^{F} \left\| \left[I(0) \quad 0_{n_B \times (n_R - n_B)} \right] \mathbf{k} - I(t)T(t)\mathbf{k} \right\|^2$$

where

$$T(t)_{mn} = \begin{cases} 1, & \text{when } m\text{th pixel moves to } n\text{th pixel} \\ & (1 \le m \le n_B \text{ and } 1 \le n \le n_R) \\ 0, & \text{otherwise.} \end{cases}$$

To obtain an unbiased estimate of $\hat{j}(x,y\,|\,t)$, we require that $\mathbf{1}^T\mathbf{k} = n_R$, where $\mathbf{1} = [1 \;\; 1 \cdots 1]^T$. Thus, we wish to minimize

$$E_B = \sum_{t=1}^{F} \left\| \left(\left[I(0) \quad 0_{n_B \times (n_R - n_B)} \right] - I(t)T(t) \right) \mathbf{k} \right\|^2 \qquad (6)$$

subject to $\mathbf{1}^T\mathbf{k} = n_R$.

This is a quadratic optimization problem with a linear constraint and thus has a unique global optimum, which can be estimated using standard methods, e.g., steepest descent or conjugate gradient [8]. Optionally, to make use of the fact that the elements of $\mathbf{k}$ are close to 1, a regularization term $\lambda\|\mathbf{k} - \mathbf{1}\|^2$ may be added to E_B. This becomes useful when temporal noise is high.

After estimating $k(x,y)$ for the entire image, one block at a time, the gain FPN corrected value for each pixel (x,y) can be computed as

$$\begin{aligned} \hat{j}(x,y,0) &= k(x,y)i(x,y,0) \\ &= a(x,y)k(x,y)i_0(x,y,0) + k(x,y)N(x,y,0). \end{aligned} \qquad (7)$$

We use $\hat{j}(x,y\,|\,0)$ over other $\hat{j}(x,y\,|\,t)$s because it does not suffer from interpolation error when the displacements are nonintegers (as will be discussed in the following subsection). Note also that this method does not result in higher temporal noise since the average value of the $k(x,y)s$ is 1.

We illustrate our method via the simple 1-D example in Fig. 2. In this example, we track brightness at four pixel locations in frame 0 and correct gain FPN for six pixel locations. Thus, we have $n_B = 4, F = 2$, and $n_R = 6$. The numbers on the left-hand side are the ideal pixel intensities and the ones on the right-hand side are the pixel intensities when gain FPN is included. Each arrow represents the motion for each pixel between consecutive frames. We assume pixel gains of 0.95, 1.02, 1.08, 0.97, 0.95, 1.03 and ignore temporal noise. Thus

$$I(0) = \begin{bmatrix} 131.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 153 & 0 & 0 & 0 & 0 \\ 0 & 0 & 135 & 0 & 0 & 0 \\ 0 & 0 & 0 & 97 & 0 & 0 \end{bmatrix}$$

$$I(1) = \begin{bmatrix} 140.76 & 0 & 0 & 0 \\ 0 & 162 & 0 & 0 \\ 0 & 0 & 121.25 & 0 \\ 0 & 0 & 0 & 95 \end{bmatrix}$$

$$I(2) = \begin{bmatrix} 149.04 & 0 & 0 & 0 \\ 0 & 145.5 & 0 & 0 \\ 0 & 0 & 118.75 & 0 \\ 0 & 0 & 0 & 103 \end{bmatrix}$$

$$T(1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$T(2) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Solving (6), we obtain

$$\mathbf{k}^* = [1.0503 \quad 0.9782 \quad 0.9239 \quad 1.0286 \quad 1.0503 \quad 0.9687]^T.$$

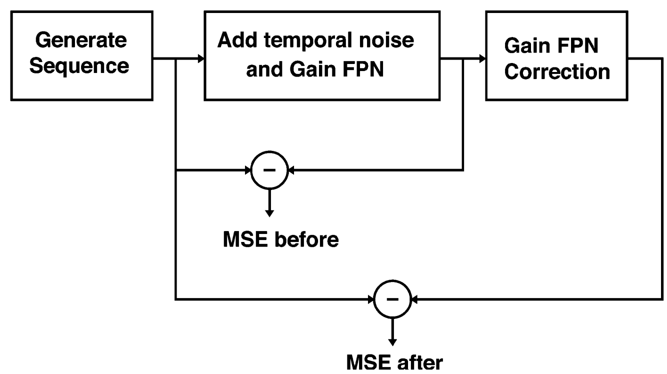
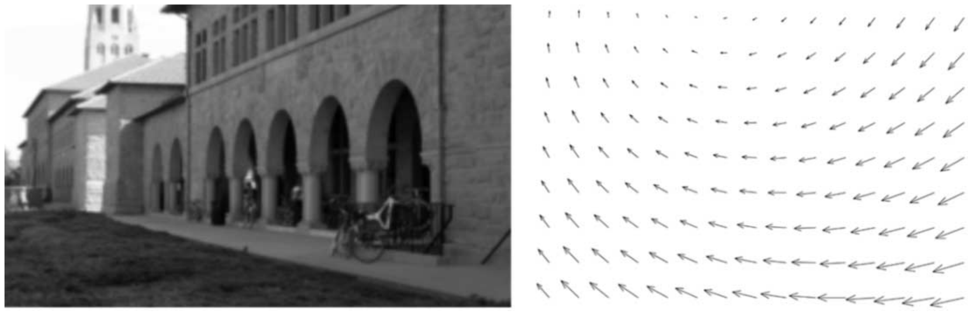


Fig. 5. Simulation setup.



(a) (b)

Fig. 6. (a) Original scene.(b) Its optical flow.

We can now correct gain FPN using (7) and we obtain

$$\begin{aligned} I(0)\mathbf{k}^* &= [137.68 \quad 149.66 \quad 124.72 \quad 99.77]^T \\ &= 0.998[138 \quad 150 \quad 125 \quad 100]^T \end{aligned}$$

which is equal to the ideal intensity multiplied by a factor of 0.998.

B. Noninteger Displacements

The method described in the Section III-A can be extended to the more realistic noninteger displacement case depicted in Fig. 3. The solid lines in the figure represent the pixel grid. The shaded area A is a brightness patch that covers a pixel in frame 0 and moves to location B in frame t. Equation (4) cannot be directly used in this case since location B overlaps with several pixels due to the noninteger displacement of A and $k(x+d_x, y+d_y)$ and $i(x+d_x, y+d_y, t)$ were only defined for integer valued $x+d_x$ and $y+d_y$. To extend our method to noninteger displacements we use spatial interpolation to estimate the intensity values at the noninteger patch locations and define $I(t)$ matrices using the interpolated values $\tilde{i}(x+d_x, y+d_y, t)$ instead of pixel intensities.

To define the gain FPN correction coefficient $k(x+d_x, y+d_y)$ at the noninteger location B note that B partially overlaps with four pixels with possibly different gain FPN values. We define the gain FPN of B as the weighted average of the four pixel gain FPN values, where the weight for a pixel is the fraction of its overlap area with B. Thus, the gain FPN correction coefficient $k(x+d_x, y+d_y)$ can be similarly defined as

$$k(x+d_x, y+d_y) = \sum_{m=0}^{1}\sum_{n=0}^{1} C_{m+1,n+1} k(x+\lfloor d_x \rfloor + m, y+\lfloor d_y \rfloor + n)$$

where

$$C = \begin{bmatrix} (1-\alpha)(1-\beta) & (1-\alpha)\beta \\ \alpha(1-\beta) & \alpha\beta \end{bmatrix}$$
$$\alpha = d_x - \lfloor d_x \rfloor, \beta = d_y - \lfloor d_y \rfloor.$$

The index (x, y, t) in C, d_x, d_y, α, and β are omitted for notational simplicity.

With these modifications, we use (4) to obtain the estimate of the ideal pixel intensity. The MSE function to be minimized is expressed in the matrix form of (6) using the modified definitions of $\tilde{i}(x+d_x, y+d_y, t)$ and $k(x+d_x, y+d_y)$. In this case, $I(t)$ is the $n_B \times n_B$ diagonal matrix whose diagonal elements are $\tilde{i}(x+d_x, y+d_y, t)$. When brightness at pixel location (x, y) in frame 0 moves to the location $(x+d_x, y+d_y)$ in frame t, the row of the $T(t)$ matrix that corresponds to pixel location (x, y), is formed by assigning $C_{m+1,n+1}$s to the elements corresponding to pixel locations $(x+\lfloor d_x \rfloor + m, y+\lfloor d_y \rfloor + n)$ and 0 s otherwise.

We illustrate the noninteger displacement case with the simple 1-D example in Fig. 4. In this example, we track brightness at four pixel locations and correct gain FPN for five pixels. Thus, $n_B = 4, F = 2$, and $n_R = 5$. Note that the magnitude of the displacements between frames 0 and 1 is 0.5 pixel and thus $T(1)$ is no longer a permutation matrix. Since half of the brightness patch at the first (left most) pixel moves to the first pixel and the other half moves to the second pixel, we set $T(1)_{11} = T(1)_{12} = 0.5$. Other rows of $T(1)$ can be defined similarly. Also, since $i(x+d_x, y+d_y, 1)$ is not defined for noninteger d_x and d_y, the diagonal elements of $I(1)$ can be obtained by interpolating $i(x, y, 1)$ to find $\tilde{i}(x+d_x, y+d_y, 1)$. In this example, $I(1)$ is obtained by performing bilinear interpolation and temporal noise is ignored

$$I(0) = \begin{bmatrix} 139.38 & 0 & 0 & 0 & 0 \\ 0 & 141 & 0 & 0 & 0 \\ 0 & 0 & 127.5 & 0 & 0 \\ 0 & 0 & 0 & 97 & 0 \end{bmatrix}$$
$$I(1) = \begin{bmatrix} 134.1 & 0 & 0 & 0 \\ 0 & 137.8 & 0 & 0 \\ 0 & 0 & 124.7 & 0 \\ 0 & 0 & 0 & 100.9 \end{bmatrix}$$
$$I(2) = \begin{bmatrix} 129.72 & 0 & 0 & 0 \\ 0 & 153 & 0 & 0 \\ 0 & 0 & 121.25 & 0 \\ 0 & 0 & 0 & 106 \end{bmatrix}$$
$$T(1) = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$
$$T(2) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Solving (6), we obtain

$$\mathbf{k}^* = [0.9737 \quad 1.0465 \quad 0.9879 \quad 1.0389 \quad 0.9530]^T.$$

(a) (b)

Fig. 7. Images before and after correction with 5% of pixel gain variation. (a) Before correction (MSE=32.38). (b) After correction (MSE=11.54).

(a) (b)

Fig. 8. Images before and after correction with 3% of pixel and 4% of column gain variation. (a) Before correction (MSE=31.17). (b) After correction (MSE=13.494).

We can now correct gain FPN using (7) and we obtain

$$I(0)\mathbf{k}^* = [135.73 \quad 147.56 \quad 125.96 \quad 100.77]^T.$$

Note that unlike in the integer displacement example, where gain FPN was completely corrected for, in this example gain FPN cannot be completely corrected due to interpolation errors.

IV. Results

To test our method, we applied it to synthetically generated video sequences so that the amount of gain FPN and displacement between frames can be controlled, and the performance of gain FPN correction can be measured. We generated the sequences using a realistic image sensor model, which included motion blur, read noise, shot noise, and gain FPN.[1] Detailed description of how these sequences are generated is provided in Lim *et al.* [9]. We measure the performance by computing the MSE between the noise-free image and the gain FPN corrected image and comparing it to the MSE between the noise-free image and the input image before gain FPN correction (see Fig. 5).

The original image with no FPN or temporal noise is shown in Fig. 6 together with its optical flow. Fig. 7 shows one frame of

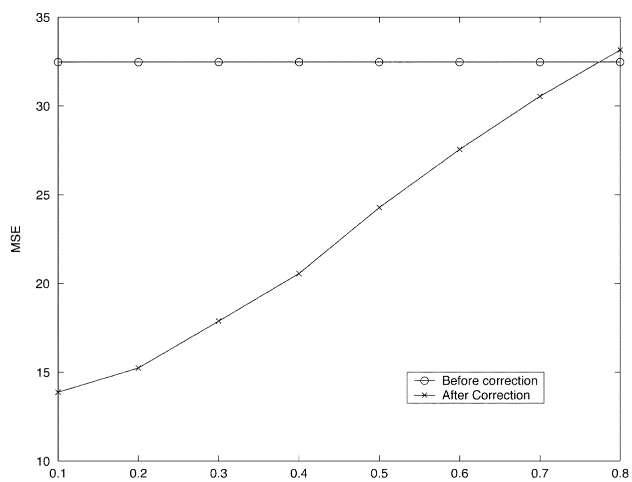
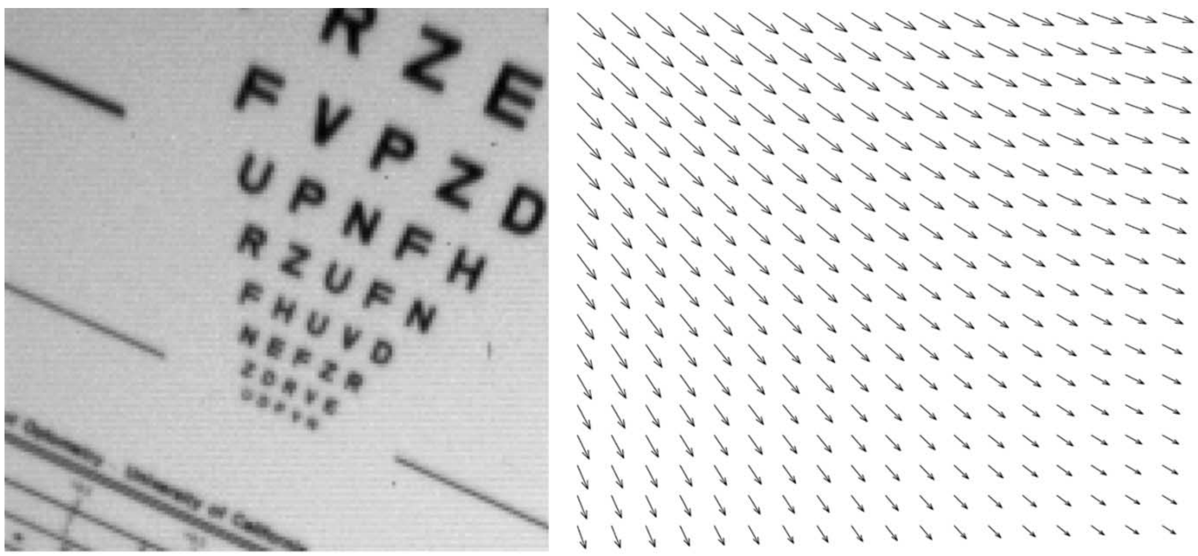


Fig. 9. Performance of the proposed method versus the standard deviation of optical flow estimation error in both x and y directions.

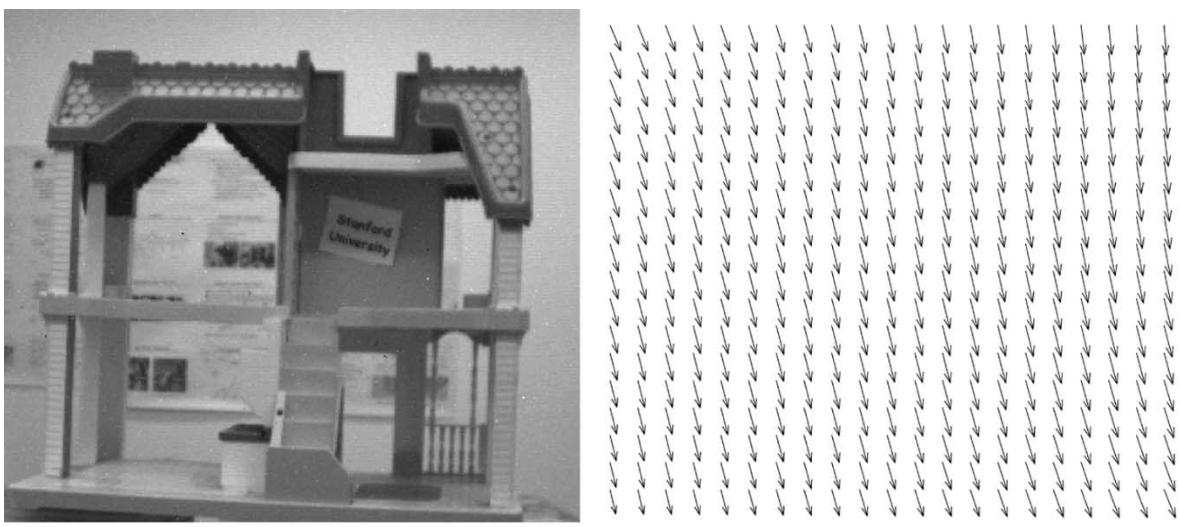
each sequence before and after gain FPN correction using block size of 5×5. After gain FPN correction, the MSE is reduced from 32.38 to 11.54. Although gain FPN is not totally removed due to temporal noise, interpolation error and motion blur, the effect of gain FPN is far less visible as can be seen from the figure. The slight blockiness in the gain FPN corrected image can be reduced by using larger block size at the cost of higher computational complexity.

[1]The following image sensor parameters are used: conversion gain of 32.5μV/e$^-$, read noise of 50 electrons, voltage swing of a 1-V, 8-bit analog–digital converter, pixel gain variation of 5%, i.e., $(\sigma_{H_{\rm pix}}/h_0) = 0.05$.



(a) (b)

Fig. 10. (a) First frame of the eye chart sequence. (b) Its optical flow.



(a) (b)

Fig. 11. (a) First frame of the doll house sequence. (b) Its optical flow.

Most CMOS image sensors suffer from column FPN due to column-wise readout circuits [5] in addition to pixel FPN. We tested our method assuming both are present with pixel gain variations of 3%, i.e., $(\sigma_{H_{\mathrm{pix}}}/h_0) = 0.03$, and column gain variations of 4%, i.e., $(\sigma_{H_{\mathrm{col}}}/h_0) = 0.04$. Fig. 8 shows one frame of each sequence before and after gain FPN correction again using block size of 5×5. After gain FPN correction, the MSE is reduced from 31.17 to 13.49.

We performed another simulation to investigate the effect of optical flow estimation error on the performance of the proposed method. We added Gaussian random noise to the true optical flow and used this for gain FPN correction. The standard deviation of optical flow error in both the x and y directions were kept the same. The performance of the proposed method versus the standard deviation of the displacement error in both x and y are plotted in Fig. 9. Note that the resulting MSE after gain FPN correction becomes larger than MSE before gain FPN correction when the standard deviation of optical flow error is larger than 0.8 pixel. Thus, the proposed method requires subpixel accuracy in the optical flow estimates. Since in this application, the user provides global motion by panning the camera, we can obtain high accuracy optical flow by fitting it to global motion parameters.

We also applied our method to real video sequences captured using an experimental high speed imaging system [4]. The system is based on the digital pixel sensor chip described in [10] and can operate at frame rates of up to 1400 frames/s. The first sequence was obtained by shaking an eye chart in front of the camera and capturing 5 frames at 200 frames/s. The second sequence was obtained by shaking the camera while capturing a dollhouse scene at 200 frames/s.

Figs. 10 and 11 show one frame of each sequence together with its estimated optical flow. We estimated optical flow of the sequence using the method described in [9]. Since the resulting

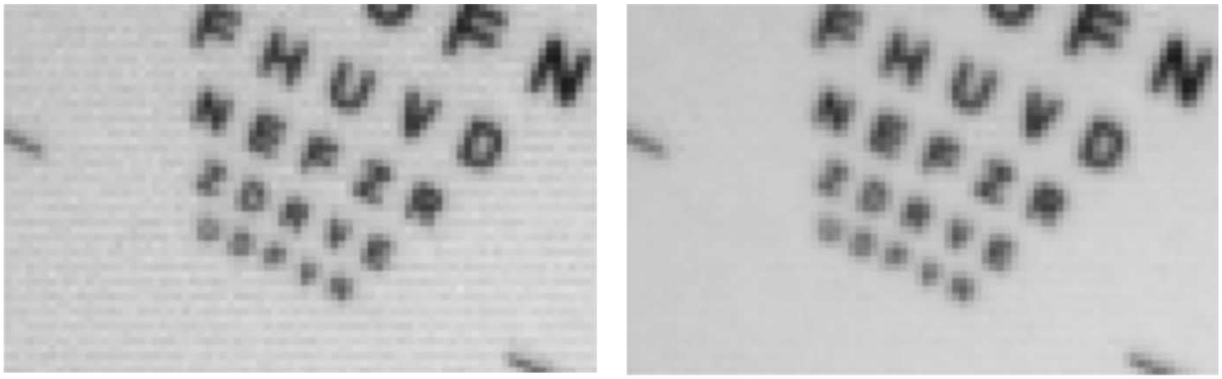


Fig. 12. Zoomed-in images before and after correction for the eye chart sequence.

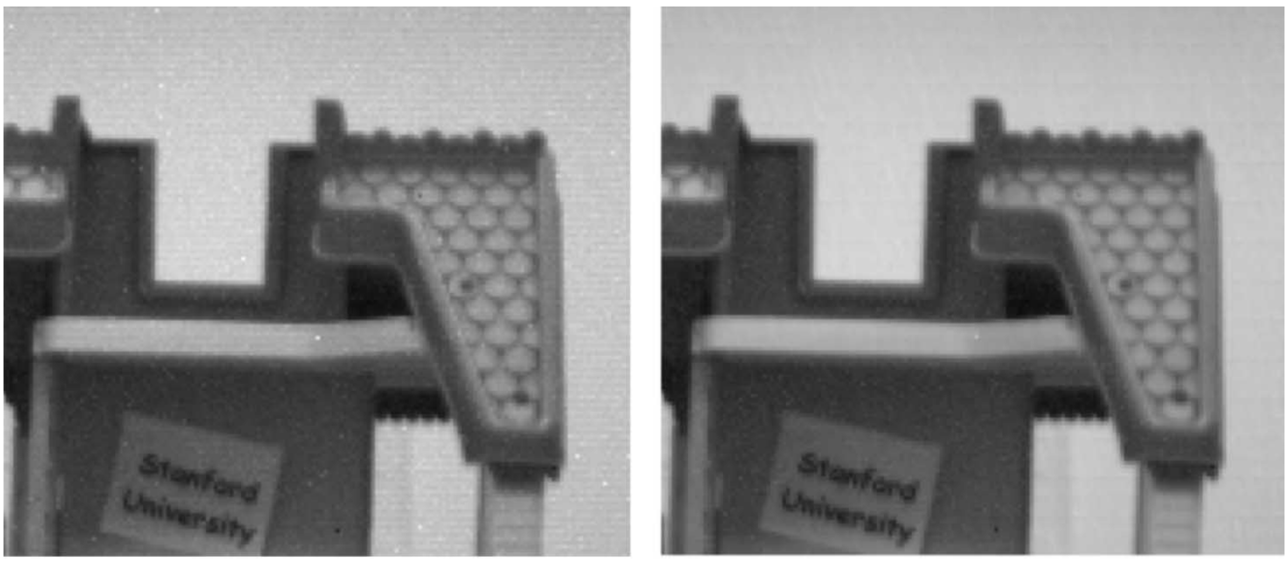


Fig. 13. Zoomed-in images before and after correction for the doll house sequence.

density of the optical flow estimates were not 100%, we fitted the optical flow estimates to global perspective warping parameters and obtained optical flow for all the pixels.

Figs. 12 and 13 show zoomed-in parts of one frame from each sequence before and after applying our gain FPN correction method. Gain FPN correction was again performed with block size of 5×5 for both scenes. Note that the horizontal lines in the image before correction, which is caused by a systematic difference between the gain of pixels in even and odd lines due to vertical mirroring in pixel layout [10], are removed after FPN correction. Also note that FPN correction did not result in any image distortion or blurring.

V. Discussion

The paper described a method for gain FPN correction using a video sequence and its estimated optical flow. Conceptually, the method can be thought of as digital CDS that cancels gain FPN rather than offset FPN. It can be used in a digital video or still camera by taking a video sequence with motion prior to capture and using it to estimate gain FPN. The sequence is then used to estimate pixel gains by iteratively minimizing the sum of the squared brightness variations along the motion trajectories. To reduce complexity we divide the pixel array into blocks, as is commonly done in video coding, and perform the estimation separately for each block. We found that a block size of 5×5 provides a reasonable tradeoff between complexity and performance. Using iterative minimization methods also allows us to lower computational requirements and reduce gain FPN incrementally by utilizing previously computed $\mathbf{k}^*$s as initial estimates instead of starting from $\mathbf{k} = \mathbf{1}$. We tested our gain FPN correction method on real and synthetic video sequences and demonstrated significant gain FPN reduction even in the presence of motion blur and temporal noise.

Acknowledgment

The authors wish to thank A.O. Ercan for his help in capturing real video sequences and K. Salama, T. Chen, H. Eltoukhy, and X. Liu for helpful discussions.

References

[1] A. Blanksby and M. J. Loinaz, "Performance analysis of a color CMOS photogate image sensor," *IEEE Trans. Electron Devices*, vol. 47, pp. 55–64, Jan. 2000.

[2] A. El Gamal, B. Fowler, H. Min, and X. Liu, "Modeling and estimation of FPN components in CMOS image sensors," in *Proc. SPIE Electronic Imaging Conf.*, vol. 3301, Apr. 1998, pp. 168–177.

[3] H. S. Bloss, J. D. Ernst, H. Firla, S. C. Schmoelz, S. K. Gick, and S. Lauxtermann, "High-speed camera based on a CMOS active pixel sensor," in *Proc. SPIE Electronic Imaging Conf.*, vol. 3968, Feb. 2000, pp. 31–38.

[4] A. Ercan, F. Xiao, X. Liu, S.H. Lim, A. El Gamal, and B. Wandell, "Experimental high speed CMOS image sensor system and applications," in *Proc. 1st IEEE Int. Conf. Sensors*, vol. 1, June 12–14, 2002, pp. 15–20.

[5] EE392B Classnotes: Introduction to image sensors and digital cameras, A. El Gamal. (2001). [Online]. Available: http://www.stanford.edu/class/ee392b

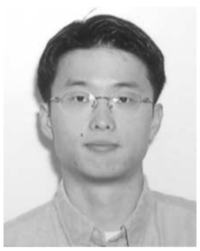
[6] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[7] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vision*, vol. 12, pp. 43–77, Feb. 1994.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: John Hopkins Univ. Press, 1996.

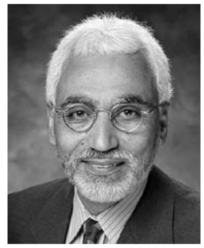
[9] S. H. Lim and A. El Gamal, "Optical flow estimation using high speed sequence," in *Proc. Int. Conf. Image Processing*, vol. 2, Oct. 2001, pp. 925–928.

[10] S. Kleinfelder, S. H. Lim, X. Liu, and A. El Gamal, "A-10000 frame/s 0.18-μm CMOS digital pixel sensor with pixel-level memory," in *Proc. IEEE Int. Solid-State Conf.*, Feb. 2001, pp. 88–89.



SukHwan Lim (S'98–M'03) received the B.S. degree in electrical engineering (with honors) from Seoul National University, Seoul, Korea, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1998 and 2003, respectively.

He is currently a member of technical staff at Hewlett-Packard Laboratories. At Stanford University, he worked on the programmable digital camera project and his research focused on high-speed CMOS image sensor and its video processing applications. He co-designed a 10000-frames/s CMOS image sensor and developed video processing applications that exploit the high-speed imaging capability. His research interests include image sensors, digital cameras and digital signal processing architectures for smart imaging devices. He is also interested in image/video processing algorithms such as optical flow estimation, denoising, deblurring and superresolution algorithms.



Abbas El Gamal (S'71–M'73–SM'83–F'00) received the B.S. degree in electrical engineering from Cairo University, Cairo, Egypt, in 1972, and the M.S. degree in statistics and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1977 and 1978, respectively.

From 1978 to 1980 he was an Assistant Professor of Electrical Engineering at the University of Southern California, Los Angeles. He joined the faculty of Stanford University in 1981, where he is currently a Professor of Electrical Engineering. From 1984 to 1988, while on leave from Stanford, he was Director of LSI Logic Research Lab, Milpitas, CA, and later the Cofounder and Chief Scientist of Actel Corporation, Mountain View, CA. From 1990 to 1995, he was a Cofounder and Chief Technical Officer of Silicon Architects, Sunnyvale, CA, which is currently part of Synopsys. He is currently the Principal Investigator on the Stanford Programmable Digital Camera project. His research interests include CMOS image sensors and digital camera design, image processing, network information theory, and electrically configurable very large-scale integration design and computer-aided design. He has authored or coauthored over 100 papers and 20 patents in these areas.

Prof. El Gamal serves on the board of directors and advisory boards of several integrated circuit and computer-aided design companies. He is a member of the ISSCC Technical Program Committee.