Gain Fixed Pattern Noise Correction via Optical Flow

SukHwan Lim and Abbas El Gamal

Information Systems Laboratory
Department of Electrical Engineering, Stanford University, CA 94305, USA

ABSTRACT

Fixed pattern noise (FPN) or nonuniformity caused by device and interconnect parameter variations across an image sensor is a major source of image quality degradation especially in CMOS image sensors. In a CMOS image sensor, pixels are read out through different chains of amplifiers each with different gain and offset. Whereas offset variations can be significantly reduced using correlated double sampling (CDS), no widely used method exists for reducing gain FPN. In this paper, we propose to use a video sequence and its optical flow to estimate gain FPN for each pixel. This scheme can be used in a digital video or still camera by taking any video sequence with motion prior to capture and using it to estimate gain FPN. Our method assumes that brightness along the motion trajectory is constant over time. The pixels are grouped in blocks and each block's pixel gains are estimated by iteratively minimizing the sum of the squared brightness variations along the motion trajectories. We tested this method on synthetically generated sequences with gain FPN and obtained results that demonstrate significant reduction in gain FPN with modest computations.

Keywords: Digital camera, CMOS image sensor, Fixed pattern noise (FPN), Optical flow estimation, gain FPN, CDS

1. 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] In a CCD sensor, since all pixels share the same output amplifier, 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.

Assuming linear sensor response, the pixel intensity value i as a function of its input signal $^*s$ can be expressed as

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

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 results from pixel to pixel variations in i_{os} and can be significantly reduced by correlated-double sampling (CDS).[2] Gain FPN is caused by variations in the gain factor h. Unlike offset FPN, gain FPN is difficult to correct and no widely used method exists for reducing it. One method is to characterize each sensor's pixel gains after manufacture and use a lookup table to correct gain FPN.[3] A problem with this method is that gain FPN changes with temperature and aging, making such "static" lookup table method 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.

Recently, there have been research efforts aimed at using the high speed capabilities of CMOS image sensors to improve image quality or enhance the performance of various imaging applications. Liu *et. al* utilized multiple image captures during a normal exposure time to reduce read noise and hence extend dynamic range and prevent motion blur.[4,5] Lim *et. al* utilized a high speed video sequence to enhance the accuracy of optical flow estimation.[6,7] The basic idea behind these research efforts is to operate the

Correspondence: Email: sukhwan@isl.stanford.edu, abbas@isl.stanford.edu; Telephone: 650-725-9696

*Input signal here can be taken as photocurrent density.[2]

Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III, Morley M. Blouke, John Canosa, Nitin Sampat, Editors, Proceedings of SPIE Vol. 4669 (2002) · 0277-786X/02/$15.00

sensor at a higher frame rate than the standard frame rate, process the video sequence and only output the enhanced images with any application specific data at the standard frame rate.[6,8] In this paper, we apply this idea to gain FPN correction. We describe a method to estimate and correct gain FPN using a high speed video sequence and its optical flow. The method 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. Such motion can either result from moving objects or caused by panning during capture.

The rest of the paper is organized as follows. In the following section, we describe the image and FPN model used throughout the paper. In Section 3, we describe our algorithm for estimating and correcting gain FPN and illustrate its operation via simple 1D examples. In Section 4, we show simulation results using synthetically generated sequence and its optical flow. We show that gain FPN is significantly reduced using our method.

2. IMAGE AND FIXED PATTERN NOISE MODEL

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

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

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 σ_H/h_0. Although gain FPN can slowly vary with temperature and aging, we assume here that $a(x,y)$ is constant while capturing several frames with high speed imager. Note that $a(x,y)=1$ for all (x,y) in an ideal sensor having no gain FPN.

We assume brightness constancy of the scene.[9,10] This implies that brightness is constant along the motion trajectory. Thus, if we capture $F+1$ frames 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 ideal pixel intensity at $t=0$, $j(x,y) = i_0(x,y,0)$ as

$$i_0(x + d_x(x,y,t), y + d_y(x,y,t), t) = j(x,y),$$

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 $d_x(x,y,0) = d_y(x,y,0) = 0$. We will assume that an accurate optical flow estimate is available, *e.g.*, using the method we described in Lim *et al.*[7]

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

$$i(x+d_x, y+d_y, t) = a(x+d_x, y+d_y) j(x,y) + N(x+d_x, y+d_y, t),$$

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 the gain FPN component $a(x,y)$ is constant over time t.

3. GAIN FPN CORRECTION ALGORITHM

Our objective 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

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

where $k(x,y)$ is the coefficient function we need to estimate. Because of the brightness constancy assumption, $j(x,y)$ is constant over time, and hence $\hat{j}(x,y,t)$ should not depend on time. Thus, we solve for $k(x,y)$ that minimizes the mean square error (MSE) between $\hat{j}(x,y,0)$ and $\hat{j}(x,y,t)$.

To reduce the computational complexity of estimating $k(x,y)$ for the entire image, we divide the image into non-overlapping blocks and independently estimate $k(x,y)$ for each block. To estimate $k(x,y)$ for pixels (x,y) in block B, we minimize the MSE function

$$E_B = \sum_{t=1}^{F} \sum_{(x,y)\in B} (\hat{j}(x,y,0) - \hat{j}(x,y,t))^2 \quad (2)$$

$$= \sum_{t=1}^{F} \sum_{(x,y)\in B} (k(x,y)i(x,y,0) - k(x+d_x, y+d_y)i(x+d_x, y+d_y, t))^2. \quad (3)$$

We first solve this problem for the simple case when the displacements are integer valued. 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 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 $T(t)$. When brightness at the i^{th} pixel location in frame 0 moves to the j^{th} pixel location in frame t, ith row of $T(t)$ is formed by assigning 1 to the jth element and 0s to the rest of the elements in that row. 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 Equation 3 in a matrix form as

$$E_B = \sum_{t=1}^{F} \| \begin{bmatrix} I(0) & 0_{n_B \times (n_R - n_B)} \end{bmatrix} \mathbf{k} - I(t)T(t)\mathbf{k} \|^2,$$

where

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

To obtain an unbiased estimate of $\hat{j}(x,y,t)$, we require that the average value of the elements of $\mathbf{k}$ equals 1. In matrix form, this constraint can be represented as $\mathbf{1}^T\mathbf{k} = n_R$, where $\mathbf{1} = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T$. Thus, we wish to minimize

$$E_B = \sum_{t=1}^{F} \| (\begin{bmatrix} I(0) & 0_{n_B \times (n_R - n_B)} \end{bmatrix} - I(t)T(t))\mathbf{k} \|^2, \quad (4)$$

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. The vector $\mathbf{k}^*$ that minimizes E_B can be solved using standard methods such as steepest descent or conjugate gradient.[11]

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

$$\hat{j}(x,y,0) = k(x,y)i(x,y,0) \quad (5)$$

$$= a(x,y)k(x,y)i_0(x,y,0) + k(x,y)N(x,y,0) \quad (6)$$

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 non-integers (which we will discuss later in this section). For our method to be successful, $a(x,y)k(x,y)$ should be very close to 1 for all (x,y). Note also that our method does not result in higher temporal noise since the average value of the $k(x,y)$s is 1.

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

$$I(0) = \begin{bmatrix} 131.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 153 & 0 & 0 & 0 & 0 \\ 0 & 0 & 135 & 0 & 0 & 0 \\ 0 & 0 & 0 & 97 & 0 & 0 \end{bmatrix}, \quad I(1) = \begin{bmatrix} 140.76 & 0 & 0 & 0 \\ 0 & 162 & 0 & 0 \\ 0 & 0 & 121.25 & 0 \\ 0 & 0 & 0 & 95 \end{bmatrix},$$

$$I(2) = \begin{bmatrix} 149.04 & 0 & 0 & 0 \\ 0 & 145.5 & 0 & 0 \\ 0 & 0 & 118.75 & 0 \\ 0 & 0 & 0 & 103 \end{bmatrix}, \quad T(1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad T(2) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

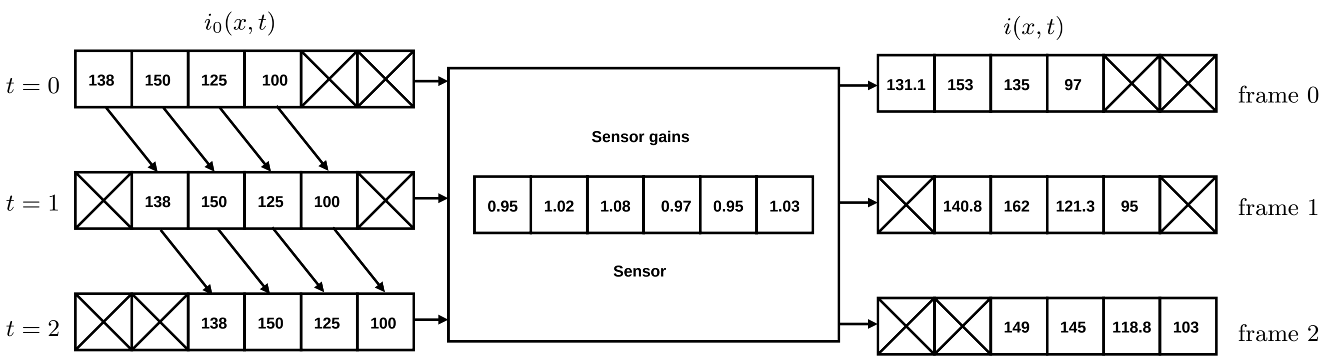


Figure 1. 1-D case simple example when the displacements are integers.

Solving Equation 4, we obtain

$$\mathbf{k}^* = \begin{bmatrix} 1.0503 & 0.9782 & 0.9239 & 1.0286 & 1.0503 & 0.9687 \end{bmatrix}^T$$

We can now correct gain FPN using Equation 5 and we obtain $I(0)\mathbf{k}^* = \begin{bmatrix} 137.68 & 149.66 & 124.72 & 99.77 \end{bmatrix}^T$, which is the ideal intensity multiplied by a 0.998 factor.

Our algorithm can be extended to the more realistic non-integer displacement case. We will not include a detailed description here. Instead, we illustrate the extension with another simple 1D example (see Figure 2). In this example, we track brightness at 4 pixel locations and try to correct gain FPN for 5 pixels. Thus, we have $n_B = 4$, $F = 2$ and $n_R = 5$. In this example, 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 assign $T(1)_{11} = T(1)_{12} = 0.5$. Other rows of $T(1)$ can be assigned similarly. Also, since $i(x + d_x, y + d_y, 1)$ is not defined for non-integer 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, we performed bilinear interpolation to obtain the elements of $I(1)$. We assume pixel gains of $1.01, 0.94, 1.02, 0.97$, and 1.06 and ignore temporal noise. Thus,

$$I(0) = \begin{bmatrix} 139.38 & 0 & 0 & 0 & 0 \\ 0 & 141 & 0 & 0 & 0 \\ 0 & 0 & 127.5 & 0 & 0 \\ 0 & 0 & 0 & 97 & 0 \end{bmatrix}, \quad I(1) = \begin{bmatrix} 134.1 & 0 & 0 & 0 \\ 0 & 137.8 & 0 & 0 \\ 0 & 0 & 124.7 & 0 \\ 0 & 0 & 0 & 100.9 \end{bmatrix},$$

$$I(2) = \begin{bmatrix} 129.72 & 0 & 0 & 0 \\ 0 & 153 & 0 & 0 \\ 0 & 0 & 121.25 & 0 \\ 0 & 0 & 0 & 106 \end{bmatrix}, \quad T(1) = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{bmatrix}, \quad T(2) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

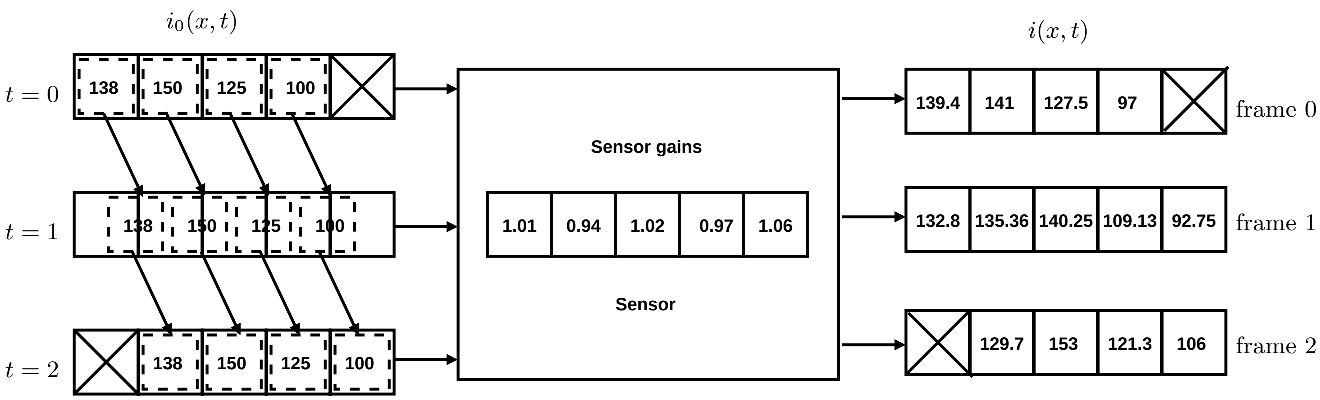


Figure 2. 1-D case simple example when the displacements are non-integers.

Solving Equation 4, we obtain

$$\mathbf{k}^* = \begin{bmatrix} 0.9737 & 1.0465 & 0.9879 & 1.0389 & 0.9530 \end{bmatrix}^T$$

We can now correct gain FPN using Equation 5 and we obtain $I(0)\mathbf{k}^* = \begin{bmatrix} 135.73 & 147.56 & 125.96 & 100.77 \end{bmatrix}^T$, Note that unlike in the integer displacement example, where gain FPN was canceled, in this example gain FPN cannot be completely canceled due to interpolation errors.

4. SIMULATION RESULTS

To test the effectiveness of 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 easily measured. We generated the sequences using a realistic image sensor model, which included motion blur, read noise, shot noise, and gain FPN. Detailed description of how these sequences are generated is provided in Lim.[7] We measure the performance of our algorithm by computing the mean square error (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.

The original image with no FPN or temporal noise is shown in Figure 3 together with its optical flow. We assume the following image sensor parameters[2]

conversion gain: $32.5\mu V/e^-$

read noise: 50 electrons

voltage swing: $1V$

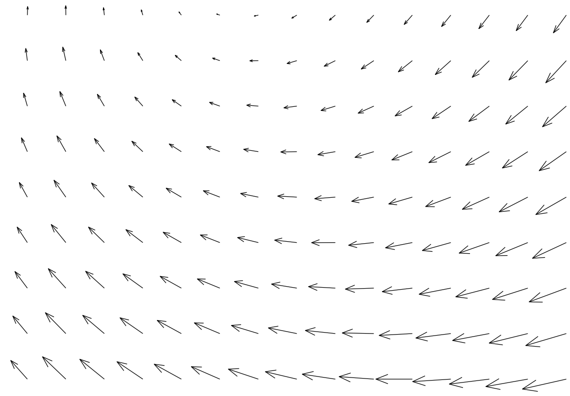
quantization: 8 bits

pixel gain variation: 5%, *i.e.*, $\frac{\sigma_{H_{pix}}}{h_0} = 0.05$

Figure 4 shows one frame of each sequence before and after gain FPN correction. After gain FPN correction with the block size of 5×5, the MSE is reduced from 32.38 to 11.54. Note that temporal noise, interpolation error and motion blur contributed to imperfect gain FPN correction. Although gain FPN is not totally removed, its effect is far less visible as can be seen from the figure. The slight blockiness in the gain FPN corrected image can be reduced by larger block size at the cost of higher computational complexity.

In most CMOS image sensors, FPN results from pixel parameter variations as well as column amplifier variations.[2] To test our gain FPN correction when both pixel and column FPN are present, we assume

Original scene



Optical flow

Figure 3. Original scene and its optical flow.

Before correction (MSE=32.38)

After correction (MSE=11.54)

Figure 4. Images before and after correction with 5% of pixel gain variation.

pixel gain variations of 3%, *i.e.*, $\frac{\sigma_{H_{pix}}}{h_0} = 0.03$, and column gain variations of 4%, *i.e.*, $\frac{\sigma_{H_{col}}}{h_0} = 0.04$. All other sensors parameters are kept the same as in the previous example. Figure 5 shows one frame of each sequence before and after gain FPN correction for block size of 5×5. After gain FPN correction, the MSE is reduced from 31.17 to 13.49.

Before correction ($MSE = 31.17$)

After correction ($MSE = 13.49$)

Figure 5. Images before and after correction with 3% of pixel and 4% of column gain variation.

5. DISCUSSION

The paper described a method for gain FPN correction using a video sequence and its estimated optical flow. 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 pixels are grouped in blocks and each block's pixel gains are corrected by iteratively minimizing the sum of the squared brightness variations along the motion trajectories.

We paid special attention to minimizing the computational requirements of the algorithm. To achieve the best results we can consider the whole image as a single block and calculate the $\mathbf{k}^*$ vector. However, this would require huge amounts of computations and would not be feasible to implement in a digital camera. To lower the computational complexity we opted to break the image into small blocks as is commonly done in video coding standards. We found that block size of 5×5 provides a reasonable tradeoff between complexity and performance. Using iterative 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 synthetically generated sequences and demonstrated significant gain FPN reduction even in the presence of motion blur and temporal noise.

ACKNOWLEDGMENTS

The work in the paper is supported under Programmable Digital Camera Program by Agilent, Canon, HP, Interval Research and Kodak. The authors wish to thank Khaled Salama, Ting Chen, Ali Ozer Ercan, Helmy Eltoukhy and Xinqiao Liu for helpful discussions.

REFERENCES

1. A. Blanksby and M. J. Loinaz, "Performance analysis of a color CMOS photogate image sensor," in *IEEE Transactions on Electron Devices*, vol. 47, pp. 55–64, January 2000.
2. A. El Gamal, *EE392B Classnotes: Introduction to Image Sensors and Digital Cameras*, http://www.stanford.edu/class/ee392b, Stanford University, 2001.
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 *Proceedings of SPIE*, vol. 3968, pp. 31–38, February 2000.
4. X. Liu and A. El Gamal, "Photocurrent estimation from multiple nondestructive samples in CMOS image sensors," *Proceedings of the SPIE Electronic Imaging Conference* **4306**, January 2001.
5. X. Liu and A. El Gamal, "Simultaneous Image Formation and Motion Blur Restoration via Multiple Capture," *Proceedings of the 2001 International Conference on Acoustics Speech and Signal Processing* **3**, pp. 1841–1844, May 2001.
6. S. H. Lim and A. El Gamal, "Integration of Image Capture and Processing – Beyond Single Chip Digital Camera," *Proceedings of the SPIE Electronic Imaging Conference* **4306**, pp. 219–226, January 2001.
7. S. H. Lim and A. El Gamal, "Optical flow estimation using high speed sequence," *Proceedings of the 2001 International Conference on Image Processing* **2**, pp. 925–928, October 2001.
8. D. Handoko, S. Kawahito, Y. Takokoro, M. Kumahara, and A. Matsuzawa, "A CMOS Image Sensor for Focal-plane Low-power Motion Vector Estimation," *Symposium of VLSI Circuits* , pp. 28–29, June 2000.
9. J. Barron, D. Fleet, and S. Beauchemin, "Performance of Optical Flow Techniques," in *International Journal of Computer Vision*, vol. 12, pp. 43–77, February 1994.
10. A. M. Tekalp, *Digital Video Processing*, Prentice-Hall, New Jersey, USA, 1995.
11. G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, Maryland, USA, 1996.