JOINT BIAS AND GAIN NONUNIFORMITY CORRECTION OF INFRARED VIDEOS USING TENSORIAL-RLS TECHNIQUE

Daniel Pipa, Eduardo A. B. da Silva, Carla Pagliari, and Marcelo M. Perez

CENPES/PETROBRAS, COPPE/UFRJ/PEE, IME, and CTEx

ABSTRACT

Infrared (IR) focal-plane array (FPA) detectors suffer from fixed-pattern noise (FPN), also known as spatial nonuniformity, which degrades image quality. In fact, FPN remains a serious problem despite recent advances in IRFPA technology. This work proposes a scene-based correction algorithm to continuously compensate for bias and gain nonuniformity in focal-plane array sensors. The proposed technique is a recursive algorithm based on recursive least square (RLS) techniques that jointly compensates for both bias and gain for each image pixel. The method converges rapidly and presents robustness to noise. Experiments with synthetic and real IRFPA videos has shown that it is competitive with the state-of-the-art in FPN reduction, presenting recovered images with higher fidelity when compared to them.

Index Terms— nonuniformity correction, fixed-pattern noise, infrared, focal-plane array, recursive-least-squares

1. INTRODUCTION

Nowadays, most infrared imaging sensors use Infrared Focal Plane Arrays (IRFPA). Each IRFPA is formed by an array of infrared detectors aligned at the focal plane of the imaging system. Due to the fabrication process, each detector presents unequal responses under the same infrared (IR) stimulus. This spatially nonuniform response produces corrupted images with a fixed-pattern noise (FPN) that has a slow and random drift requiring constant compensation [1]. Hence, the output signal of IR detectors needs to be corrected to ensure, at least, a workable image for the required application. An accepted approach is to model the pixels' response as linear plus a constant [1, 2]; we thus define for each detector (pixel) an offset, or bias, and a gain. By correcting these offsets and gains one aims to obtain the 'same' response for the entire FPA. Also, since these FPA parameters drift over time, such correction has to be performed periodically or even on a frame-by-frame basis.

As, in most sensors, the bias nonuniformity dominates the gain nonuniformity, many nonuniformity correction methods do not compensate for the latter [1, 3]. However, better results are achieved when both parameters are corrected. This paper proposes a new adaptive scene-based nonuniformity correction (NUC) algorithm that jointly compensates for bias and gain parameters on a frame-by-frame basis.

This paper is organized as follows. Section 2 provides a review of the nonuniformity problem on IRFPA's. Section 3 is devoted to the proposed NUC method. In Section 4, experimental results with real and synthetic infrared videos are presented, together with a comparison to other techniques. Section 5 contains the final remarks and conclusions.

2. IRFPA MODELS, FIXED-PATTERN NOISE AND NONUNIFORMITY CORRECTION METHODS

2.1. FPA Response Model

A commonly used bias-gain linear model for an FPA sensor is given by:

$$y_k(i,j) = a(i,j)x_k(i,j) + b(i,j) \tag{1}$$

where $y_k(i,j)$ is the response (measured signal) of the $(i,j)^{th}$ pixel of the IR camera at frame k, $a(i,j)$ is the gain associated to the $(i,j)^{th}$ pixel, $x_k(i,j)$ is the incident infrared radiation collected by the respective detector at pixel coordinates (i,j) at frame k, and $b(i,j)$ is the bias associated to pixel at coordinates (i,j),where k=1,2,... represents the frame number associated to its time instant.

NUC algorithms target to estimate the actual infrared radiation $x_k(i,j)$ by estimating the gain and offset parameters from the readout values $y_k(i,j)$.

2.2. Nonuniformity Correction Techniques

If we can write equation (1) for every pixel (i,j) and two values of k, we can solve the system of equations and compute $a(i,j)$ and $b(i,j)$. However, this solution requires the knowledge of $x_k(i,j)$. The NUC methods are thus categorized according to the way the values of $x_k(i,j)$ are estimated, namely, calibration-based (or reference-based), and scene-based [4].

Reference-based calibration methods for NUC use uniform infrared sources (blackbody radiators) so that $x_k(i,j)$ is precisely known for all (i,j). The most widespread technique is the Two-Point Calibration method [2], which employs two blackbody radiation sources at different tempera-

tures to calculate both gain and bias parameters. Despite providing radiometrically accurate corrected imagery, such kind of method interrupts the normal operation of the system during the calibration stage, which is inconvenient in many applications.

Scene-based NUC techniques can overcome this drawback by exploiting motion-related features in IR videos in order to estimate $x_k(i,j)$. In general, these techniques are classified as statistical [5] and registration-based. Registration-based techniques are used to track pixel (or pixel-block) motion between frames, and calculate the associated parameters for the detectors related to the estimated displacements [6].

The drift presented by FPNs varies rather slowly. This favours the use of time-invariant parameters modelling, together with the tracking of the slow variation in the parameters.

3. PROPOSED METHOD

The proposed method can be classified as a registration-based algorithm [1]. This technique requires accurate estimation of displacements between frames, which are used to register all image frames within the sequence. For shift estimation the LIPSE algorithm described in [1] was used.

3.1. Problem formulation

Consider the observation model of an infrared image acquisition system given by:

$$\mathbf{y}_k = \mathbf{A}\mathbf{x}_k + \mathbf{b}, \tag{2}$$

where $\mathbf{y}_k$ is a vector representing the observed image at time k, $\mathbf{A} = \mathrm{diag}(a_1, \cdots, a_N)$ is a diagonal matrix whose elements are the gain factors associated to the image pixels, $\mathbf{x}_k$ is a vector representing the real image at time k and $\mathbf{b}$ is a vector representing the bias of the acquired data, with all vectors in a lexicographical order. The gain and the bias (offset) factors are considered time invariant due to their slow drift [4]. If $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are estimated values of the gain and bias, respectively, an estimation of the real image is given by:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{A}}^{-1}\left(\mathbf{y}_k - \hat{\mathbf{b}}\right). \tag{3}$$

As this work proposes the estimation of the bias and gain parameters continuously, we consider the motion between two consecutive frames of an IR image sequence as follows:

$$\mathbf{x}_k = \mathbf{M}_k\mathbf{x}_{k-1} + \boldsymbol{\gamma}_k, \tag{4}$$

where $\mathbf{M}_k$ is the matrix that implements the displacement between consecutive frames $k-1$ and k, and $\boldsymbol{\gamma}_k$ is the vector that models the next frame updates that cannot be obtained by a simple displacement.

We suppose that the motion between two successive frames as being obtained by a motion estimation algorithm as the one in [1], and also that vector γ_k is negligible. By combining equations (3) and (4), it is possible to write the estimation error vector of frame k based on frame $k-1$, the shift matrix $\mathbf{M}$, gain and bias estimates as:

$$\boldsymbol{\epsilon}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k \tag{5}$$

$$\boldsymbol{\epsilon}_k = \mathbf{y}_k - \hat{\mathbf{A}}\mathbf{M}_k\hat{\mathbf{A}}^{-1}\left(\mathbf{y}_{k-1} - \hat{\mathbf{b}}\right) - \hat{\mathbf{b}}, \tag{6}$$

where $\boldsymbol{\epsilon}_k$ is the estimation error vector. The mean square error is given by $\varepsilon_k = \frac{1}{N}\sum_{i=1}^{N}\left[\epsilon_k(i)\right]^2$.

3.2. RLS algorithm

RLS algorithms aim to minimize a weighted sum of square errors [7], that is

$$\xi_k = \sum_{i=0}^{k}\lambda^{k-i}\varepsilon_i, \tag{7}$$

where $0 \ll \lambda \leq 1$ refers to a *forgetting factor*.

After some manipulation, it can shown that the update equation for the RLS algorithm may be written as:

$$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k - \hat{\mathbf{H}}_k^{-1}\nabla_{\mathbf{b}}\varepsilon_k, \tag{8}$$

where $\hat{\mathbf{H}}_k$ is an estimate of the Hessian matrix and $\nabla_{\mathbf{b}}\varepsilon_k$ is the *a priori* error gradient[7]. The following relations hold for the Hessian matrix [8]:

$$\mathbf{H}_k \triangleq \nabla_{\mathbf{b}}^2\xi_k = \frac{\partial^2\xi_k}{\partial\mathbf{b}\partial\mathbf{b}^{\mathrm{T}}} = \lambda\hat{\mathbf{H}}_{k-1} + \frac{\partial^2\varepsilon_k}{\partial\mathbf{b}\partial\mathbf{b}^{\mathrm{T}}}. \tag{9}$$

The above equations show a way to update the Hessian matrix at each step.

3.3. Bias correction by Tensorial-RLS method

The last term of equation (9) can be computed as (k index dropped for simplicity):

$$\frac{\partial^2\varepsilon}{\partial\mathbf{b}\partial\mathbf{b}^{\mathrm{T}}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial^2}{\partial\mathbf{b}\partial\mathbf{b}^{\mathrm{T}}}\left[\epsilon(i)\right]^2 \tag{10}$$

Since the term $\frac{2}{N}$ is constant, we define $\hat{\mathbf{H}} = \frac{2}{N}\hat{\mathbf{H}}'$. It also can be shown that:

$$\nabla_{\mathbf{b}}\varepsilon = \frac{2}{N}\frac{\partial\boldsymbol{\epsilon}}{\partial\mathbf{b}}\boldsymbol{\epsilon}. \tag{11}$$

With these definitions, we have that:

$$\frac{\partial\boldsymbol{\epsilon}}{\partial\mathbf{b}} = \left(\mathbf{A}\mathbf{M}\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}. \tag{12}$$

Then, the Tensorial-RLS algorithm for bias correction is given by:

$$\boldsymbol{\epsilon}_k = \mathbf{y}_k - \mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\left(\mathbf{y}_{k-1} - \hat{\mathbf{b}}_k\right) - \hat{\mathbf{b}}_k \tag{13}$$

$$\hat{\mathbf{H}}'_k = \lambda\hat{\mathbf{H}}'_{k-1} + \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right] \tag{14}$$

$$\cdot\left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right) - \mathbf{I}\right] \tag{15}$$

$$\mathbf{u}_k = \left[\left(\mathbf{A}\mathbf{M}_k\mathbf{A}^{-1}\right)^{\mathrm{T}} - \mathbf{I}\right]\boldsymbol{\epsilon}_k \tag{16}$$

$$\mathbf{v}_k = CG(\hat{\mathbf{H}}'_k, \mathbf{u}_k) \tag{17}$$

$$\hat{\mathbf{b}}_{k+1} = \hat{\mathbf{b}}_k - \mathbf{v}_k, \tag{18}$$

where CG represents the solution of a equation system by the conjugate gradient method in order to avoid the inversion of matrix $\hat{\mathbf{H}}'_k$.

3.4. Gain correction by Tensorial-RLS method

The update equation for the gain estimation by Tensorial-RLS can be written as

$$\hat{a}_{ii,k+1} = \hat{a}_{ii,k} - \hat{\mathbf{\Gamma}}_k^{-1} \nabla_{a_{ii}} \varepsilon_k, \tag{19}$$

where $\hat{\mathbf{\Gamma}}_k$ is a Hessian matrix estimate and $\nabla_{a_{ii}} \varepsilon_k$ is the *a priori* error gradient[7].

When the gradient is applied to the error, one gets the following (k index dropped for simplicity):

$$\nabla_{a_{ii}} \varepsilon = \frac{\partial \boldsymbol{\epsilon}}{\partial a_{ii}} \boldsymbol{\epsilon} = -\mathbf{z}^{\mathrm{T}} \left[\frac{\partial \mathbf{A}}{\partial a_{ii}} \mathbf{M} \mathbf{A}^{-1} + \mathbf{A} \mathbf{M} \frac{\partial \mathbf{A}^{-1}}{\partial a_{ii}} \right]^{\mathrm{T}} \boldsymbol{\epsilon}. \tag{20}$$

The Hessian matrix is given by [8]:

$$\mathbf{\Gamma}_k \triangleq \nabla^2_{a_{ii}} \xi_k = \frac{\partial^2 \xi_k}{\partial a_{ii}^2} = \lambda \hat{\mathbf{\Gamma}}_{k-1} + \frac{\partial^2 \varepsilon_k}{\partial a_{ii}^2}. \tag{21}$$

The second order gradient of the error is obtained as:

$$\frac{\partial^2 \varepsilon}{\partial a_{ii}^2} = \frac{2}{N} \left\{ \left[\frac{\partial^2 \boldsymbol{\epsilon}}{\partial a_{ii}^2} \right]^{\mathrm{T}} \cdot \boldsymbol{\epsilon} + \frac{\partial \boldsymbol{\epsilon}}{\partial a_{ii}} \left[\frac{\partial \boldsymbol{\epsilon}}{\partial a_{ii}} \right]^{\mathrm{T}} \right\}. \tag{22}$$

Applying the second order gradient to the estimation error, one obtains:

$$\frac{\partial^2 \boldsymbol{\epsilon}}{\partial a_{ii}^2} = -\mathbf{z}^{\mathrm{T}} \left[2 \frac{\partial \mathbf{A}}{\partial a_{ii}} \mathbf{M} \frac{\partial \mathbf{A}^{-1}}{\partial a_{ii}} + \mathbf{A} \mathbf{M} \frac{\partial^2 \mathbf{A}^{-1}}{\partial a_{ii}^2} \right]^{\mathrm{T}} \tag{23}$$

where $\mathbf{z} = (\mathbf{y} - \mathbf{b})$.

In order to aid the visualization of the equations, matrix $\bar{\mathbf{A}}_k = \mathbf{A}_k^{-1}$ is defined . Moreover, the higher order derivatives of the gain matrix $\mathbf{A}$ and its inverse are presented as:

$$\dot{\mathbf{A}}_i = \frac{\partial \mathbf{A}}{\partial a_{ii}} = \begin{bmatrix} \mathbf{0} & & \\ & 1_{ii} & \\ & & \mathbf{0} \end{bmatrix} \tag{24}$$

$$\dot{\bar{\mathbf{A}}}_i = \frac{\partial \mathbf{A}^{-1}}{\partial a_{ii}} = \begin{bmatrix} \mathbf{0} & & \\ & -a_{ii}^{-2} & \\ & & \mathbf{0} \end{bmatrix} \tag{25}$$

and

$$\ddot{\bar{\mathbf{A}}}_i = \frac{\partial^2 \mathbf{A}^{-1}}{\partial a_{ii}^2} = \begin{bmatrix} \mathbf{0} & & \\ & 2a_{ii}^{-3} & \\ & & \mathbf{0} \end{bmatrix}, \tag{26}$$

where only the ii^{th} elements differ from zero.

The tensorial-RLS algorithm for gain estimation then becomes:

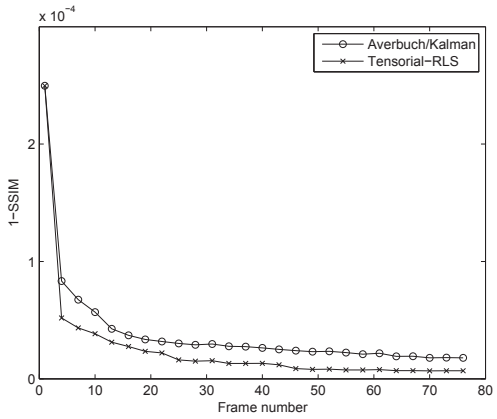

Fig. 1. Averbuch/Kalman vs Proposed Method

$$\mathbf{z}_k = \left(\mathbf{y}_{k-1} - \mathbf{b}\right) \tag{27}$$

$$\boldsymbol{\epsilon}_k = \mathbf{y}_k - \mathbf{A}_k \mathbf{M}_k \bar{\mathbf{A}}_k \mathbf{z}_k - \mathbf{b} \tag{28}$$

$$\forall i \tag{29}$$

$$\mathbf{u}_{i,k} = \left[\dot{\mathbf{A}}_i \mathbf{M}_k \bar{\mathbf{A}}_k + \mathbf{A}_k \mathbf{M}_k \dot{\bar{\mathbf{A}}}_i \right] \mathbf{z}_k \tag{30}$$

$$v_{i,k} = \mathbf{z}_k^{\mathrm{T}} \left[2 \dot{\mathbf{A}}_i \mathbf{M}_k \dot{\bar{\mathbf{A}}}_i + \mathbf{A}_k \mathbf{M}_k \ddot{\bar{\mathbf{A}}}_i \right]^{\mathrm{T}} \boldsymbol{\epsilon}_k + \mathbf{u}_{i,k}^{\mathrm{T}} \mathbf{u}_{i,k} \tag{31}$$

$$\gamma_{i,k} = \lambda \gamma_{i,k-1} + v_{i,k} \tag{32}$$

$$\hat{a}_{ii,k+1} = \hat{a}_{ii,k} - \gamma_{i,k}^{-1} \mathbf{u}_{i,k}^{\mathrm{T}} \boldsymbol{\epsilon}_k. \tag{33}$$

4. EXPERIMENTAL RESULTS

In this section, selected results of the experiments conducted over synthetic and real infrared video are presented. The main conclusions are discussed in Section 5.

In order to compare the proposed method with the state-of-the-art algorithm shown in [6], both methods were applied to 50 synthetic videos, each of dimensions 128×128 and 75 frames. These videos were generated by moving a window over a static image. Displacements were randomly introduced with standard deviation of 1 pixel, as well as bias ($\sigma_b = 0.1$) and gain ($\sigma_a = 0.004$) according to equation (1). In both cases, LIPSE [1] algorithm was used for motion estimation. Zero-mean gaussian noise with $\sigma_n = 0.005$ was also applied to all frames. Figure 1 compares the results produced by the Averbuch/Kalman [6] method with those of the proposed method using the SSIM metric [9]. The synthetic videos were artifficialy corrupted by a computer generated FPN noise. Figure 1 compares the results obtained for the synthetic FPN corrupted videos generated from image LENA.

In addition, tests were performed with real thermal images. We shot video sequences using a FLIR SYSTEMS model ThermaCAM P65 infrared camera, with a *focal plane*

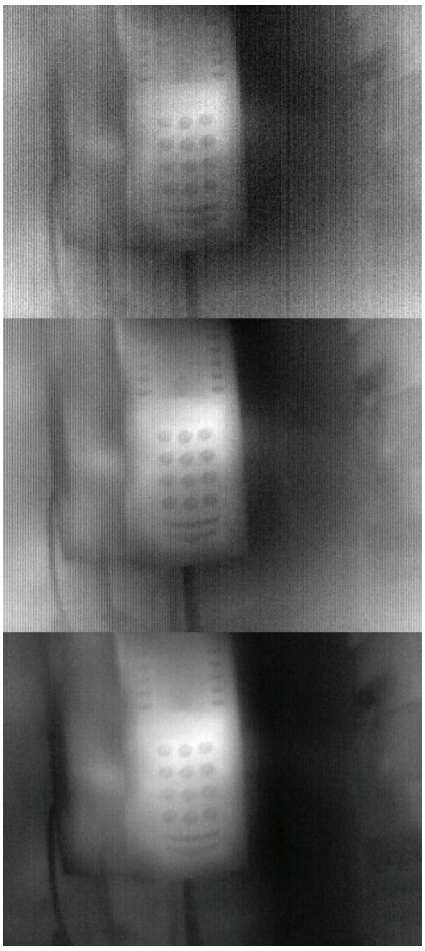

Fig. 2. Original real video (top), Averbuch/Kalman correction (center) and Tensorial-RLS correction (bottom).

array uncooled microbolometer detector. Each infrared sequence consists of 100 frames with picture size of 320×240 pixels at 60Hz. The "*Noise Reduction*" option was switched off, as well as the "*Shutter Period*" option. The latter refers to the FPN correction provided by the camera manufacturer.

The scene pictured in Figure 2 shows one frame of the video of a telephone shot with the above IR camera. On the top is the original, in the middle the Averbuch/Kalman [6] corrected video and at the bottom the video corrected by the proposed method.

5. CONCLUSIONS

The Tensorial-RLS method outperformed the Averbuch/Kalman method [6] according to the SSIM metric, as depicted in Figure 1. Note that the proposed method, unlike [6], corrects both bias and gain; yet, it is computationally faster due to the use of an RLS strategy.

The real videos results displayed in Figure 2 show that the correction achieved by the proposed method is superior to the one produced by the algorithm in [6]. The bottom picture shows that the FPN is completely removed by the proposed method, producing a clear picture of the scene, while the Averbuch/Kalman results in the middle picture still shows the noise distributed all over the image. Simulations have been run for several real videos, achieving similar results.

The proposed method outperfoms the one presented in [6] for real and synthetic videos, displaying noise-free images that allow accurate thermal image analysis for many applications.

6. ACKNOWLEDGMENTS

The authors would like to thank the Research and Development Center of Petrobras-CENPES, the Federal University of Rio de Janeiro-UFRJ and its Graduate School and Research in Engineering-COPPE, the Brazilian Army Technological Center-CTEx,the Military Institute of Engineering-IME, and the Brazilian Innovation Agency-FINEP, Brazil for their financial support.

7. REFERENCES

[1] B. M. Ratliff, *A Generalized Algebraic Scene-based Nonuniformity Correction Algorithm*, Ph.D. thesis, The University of New Mexico, 2004.

[2] D. L. Perry and E. L. Dereniak, "Linear theory of nonuniformity correction in infrared staring sensors," *Optical Engineering*, vol. 32 no. 8, pp. 1853–1859, 1993.

[3] Daniel R. Pipa, Eduardo A. B. da Silva, and Carla Pagliari, "Correção de não uniformidade em vídeo infravermelho por gradiente descendente," in *XXVI Simpósio Brasileiro de Telecomunicações - SBrT'08*, 2008.

[4] J. E. Pezoa S. N. Torres and M. M. Hayat, "Scene-based nonuniformity correction for focal plane arrays using the method of the inverse covariance form," *Applied Optics*, vol. 42 no. 29, pp. 5872–5881, Oct. 2003.

[5] M. M. Hayat et alli, "Statistical algorithm for nonuniformity correction in focal-plane arrays," *Appl. Opt.*, vol. 38 no. 5, pp. 772–780, 1999.

[6] A. Averbuch, G. Liron, and B. Z. Bobrovsky, "Scene based non-uniformity correction in thermal images using kalman filter," *Image Vision Comput.*, vol. 25 no. 6, pp. 833–851, 2007.

[7] P. S. R. Diniz, *Adaptive filtering: Algorithms and Practical Implementations (3rd ed.)*, Springer, Boston, MA, 2008.

[8] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*, Meboo, Palo Alto, 2008.

[9] H. R. Sheikh Z. Wang, A. C. Bovik and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13 no. 4, 2004.