

# ADMIRE: a locally adaptive single-image, non-uniformity correction and denoising algorithm: application to uncooled IR camera

Y. Tendo<sup>a</sup>, J. Gilles<sup>b</sup>,

<sup>a</sup> Centre de Mathématiques et de Leurs Applications (CMLA)

École Normale Supérieure de Cachan - 61 av du Pdt Wilson 94235 Cachan Cedex France.

<sup>b</sup> Department of Mathematics - University of California Los Angeles  
520 Portola Plaza, Los Angeles, CA 90095.

## ABSTRACT

We propose a new way to correct for the non-uniformity (NU) and the noise in uncooled infrared-type images. This method works on static images, needs no registration, no camera motion and no model for the non uniformity. The proposed method uses an hybrid scheme including an automatic locally-adaptive contrast adjustment and a state-of-the-art image denoising method. It permits to correct for a fully non-linear NU and the noise efficiently using only one image. We compared it with total variation on real raw and simulated NU infrared images. The strength of this approach lies in its simplicity, low computational cost. It needs no test-pattern or calibration and produces no “ghost-artefact”.

**Keywords:** Non uniformity correction, Infrared, Fixed Pattern Noise, Focal Plane Array, NUC, denoising.

## 1. INTRODUCTION

Infrared (IR) imaging has proved to be a very efficient tool in a wide range of industry, medical, and military applications. Infrared cameras are used to measure temperatures, signatures, to perform detection, etc. However, the performance of the imaging system is strongly affected by the random spatial response of each pixel sensor. Under the same illumination the readout of each sensor is different. It leads to a structured noise resulting in a row or line pattern in the images (depending on the readout system). This “noise” is called fixed pattern noise and produces “non uniformity” in the observed images. These differences between sensor readout are due to mismatches in the fabrication process, among other issues<sup>1</sup> and are stronger at longer wavelength such as in infrared imaging.<sup>2</sup> The readout of a pixel sensor is a non linear<sup>1</sup> function of the incoming luminance. The non uniformity is a serious practical limitation to both civilian and military applications - as it severely

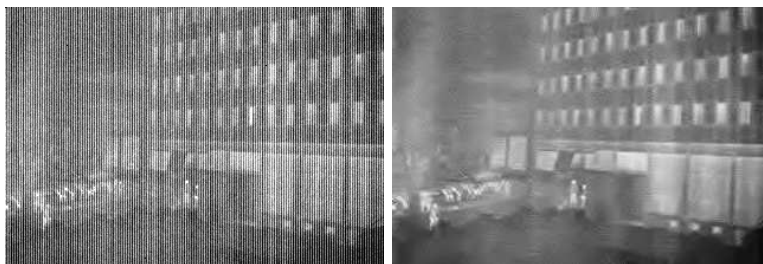


Figure 1. On the left : an image (RAW) taken with an infrared camera. The non uniformity is so strong that it is hard to distinguish between the noise and the underlying landscape. On such an image performing an identification, matching pattern, etc. is almost hopeless. On the right the same image corrected with the proposed method.

---

Further author information: (Send correspondence to Y. Tendo)

E-mail: tendero@cmla.ens-cachan.fr, Phone: +33 1 47 40 59 48

degrades image quality<sup>3</sup> (see Fig. 1). For uncooled infrared cameras the problem is even worse because the detector response evolves quickly with time. Therefore the correction cannot be done once and for all by the manufacturer. It also means that we need to estimate, for each pixel, a function with little or no model at all and using little or one image to aim a good correction. Indeed the use of numerous images to achieve the correction leads to artefacts –those are called “ghosts artefacts” and are challenging to remove<sup>4–9</sup>– because of the sensor drift.<sup>2,10,11</sup> In other words, for each pixel, the correction at times  $t_1$  and  $t_2 \neq t_1$  are different. A correction is so much needed, that in many uncooled infrared cameras a flap closes every 30 seconds to perform a partial calibration.<sup>12,13</sup> This interrupts the image flow, which is calamitous for many applications. Thus, for uncooled infrared cameras a periodic update of the non uniformity correction is required. A good non uniformity algorithmic correction is a key factor in ensuring the best image quality and the robustness of the downstream applications such as pattern recognition, image registration, etc.

In this paper we introduce locally adaptive version of our precedent work.<sup>14</sup> A review of existing techniques is proposed in section 3. A single image, fully automatic, non uniformity correction algorithm is detailed in section 4 and generalized in section 5. It shows that motion compensation or accumulation algorithms are not necessary to achieve a good image quality. The proposed method can compensate for a fully non linear non uniformity, without any parametric model on the non uniformity side. It does not require motion, or motion compensation, does not need a test pattern or calibration and does not produce any “ghost artifact”. A state of the art denoising algorithm is modified to suit our context. The proposed method is illustrated in section 6 on simulated non linear non uniformity in section Figs. 6-7, compared with a total variation based method (this method is described in section 6.1) in section 6.2 and evaluated on real raw images from thermal infrared cameras in section 6.3.

## 2. IMAGE ACQUISITION MODEL

An imaging sensor is a device that collects photons and converts them into charges. The majority of imaging sensors are Charge-Coupled Devices (CCD). The standard readout technique of CCDs works for each row (or line) independently and consists of transporting charges from the pixels to a counter (which produces the numerical value to be read). Each pixel has its own (and unknown) transfer function response. Furthermore, for each column the counter transfer function is different. The function resulting of the whole chain sensor-counter is not linear.<sup>1</sup> In the sequel we will assume without loss of generality (w.l.o.g.) that the non uniformity comes only from the sensor part. It is the difference between (transfer) functions that produces the non uniformity and leads to a structured noise resulting in a row or line pattern in the images. The perturbation model is

$$o(i, j, t) = \phi_{(i,j,t)}(u(i, j, t) + \eta(i, j, t)), \quad \forall (i, j, t) \in \{1, \dots, N\} \times \{1, \dots, M\} \times \mathbb{R}_+, \quad (1)$$

where

- $(i, j, t) \in \{1, \dots, N\} \times \{1, \dots, M\} \times \mathbb{R}_+$  is the pixel at position  $(i, j)$  and time  $t \geq 0$ ;
- $u(i, j, t)$  is the ideal (noiseless) landscape value;
- $\eta(i, j, t)$  is a random photon noise;
- $\phi_{(i,j,t)} : \{0, \dots, 255\} \mapsto \{0, \dots, 255\}$  is the contrast change (transfer function) of the pixel sensor at position  $(i, j)$  and time  $t \geq 0$ ;
- $o(i, j, t)$  is the observed value at position  $(i, j)$  and time  $t \geq 0$ .

Omitting the noise,  $\phi_{(i,j,t)}(x)$  represents the readout of the pixel  $(i, j)$  at time  $t$  for some incident radiance  $x$ . At time  $t$  the transfer function of the pixel  $(i, j)$  and the transfer function of the counter are contained in  $\phi_{(i,j,t)}$  (w.l.o.g). At each pixel sensor  $(i, j)$  and time  $t$  the photon (Poisson) noise  $\eta(i, j, t)$  is sensed (and added to the ideal landscape pixel value  $u(i, j, t)$ ), thus it also undergoes the contrast change function  $\phi_{(i,j,t)}$ . Consequently, (1) models thoroughly the whole acquisition process including the noise that is also modified by the non-uniformity

of the sensor array. It permits to deal with a realistic *non linear* sensor response<sup>1</sup> contrarily to the classic linear (gain/offset) approximation used in the literature. The goal of a non-uniformity correction algorithm is to compensate for the *local* contrast changes induced by the  $\phi_{(i,j,t)}$  which means to apply some  $\tilde{\phi}_{(i,j,t)}$  such that  $\tilde{\phi}_{(i,j,t)}(\phi_{(i,j,t)}) = g(x)$  for  $(i, j, t) \in \{1, \dots, N\} \times \{1, \dots, M\} \times \mathbb{R}_+$ . Notice that, in general, it is useless to ask for  $g = Id$  since users, screens (gamma correction), etc. usually tune the contrast of the images at ease.

### 3. RELATED WORK

To get the rid of the non-uniformity many techniques have been developed over the years. It is possible to classify them into two main kinds:

- Calibration based techniques consist in an equalization of the response to an uniform black body source of radiations. They are not convenient for real time applications, since they force to interrupt the image flow. (This calibration is usually automatic, a shutter closing in front of the lens periodically). They usually assume that  $\phi_{(i,j,t)}(x) = x + b_{(i,j)} \forall t \in [t_1, t_2[$  (so called one point NUC, one black body) or that  $\phi_{(i,j)}(x) = a_{(i,j)}x + b_{(i,j)} \forall t \in [t_1, t_2[$  is linear (two points NUC using two black bodies). Thus it assumes piecewise constants  $\phi_{(i,j,t)}$  on the time interval  $[t_1, t_2[$ . A new correction is performed every  $t_2 - t_1$  (because of the sensor drift). Of course the two points NUC<sup>15</sup> performs better.
- Scene based techniques, involving motion compensation or temporal accumulation. Such methods are complex and require certain observation conditions (motion). They usually assume linear  $\phi_{(i,j)}$ .

In the sequel, we will focus on scene based techniques as calibration based techniques require to interrupt the camera which is calamitous in practice. Numerous algorithms have been reported in the literature to remove the fixed pattern noise caused by the lack of a cross-pixels sensor equalization. Some algorithms estimate the sensor parameters while, equivalently, others attempt at recovering the “true” landscape value  $u(i, j, t)$ . These algorithms process a sequence of images  $(o(i, j, t))_{t \in \{1, \dots, L\}}$ , not a single frame. Thus they are subject to the creation of “ghost artefacts”, the reason is discussed bellow. Most of them use a simplified (linear) model for the transfer function of the pixel sensor:

$$o(i, j, t) = a_{(i,j,t)}(u(i, j, t) + \eta(i, j, t)) + b_{(i,j,t)}, \quad \forall (i, j, t) \in \{1, \dots, N\} \times \{1, \dots, M\} \times \mathbb{R}_+ \quad (2)$$

There are methods like<sup>16</sup> suggesting to equalize the mean and standard deviation (*stddev*) *through time* of each pixel sensor by a linear transform. Such algorithms rely on the data diversity found in most of the video sequences with some degree of motion. The key idea is

[ $\mathcal{H}$ :] If all pixel sensors have seen the same landscape, they should have (at least) the same mean and same standard deviation, namely

$$\underset{t \in \{1, \dots, L\}}{\text{mean}} (o(i, j, t)) = C_m \quad \forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, M\} \quad (3)$$

$$\underset{t \in \{1, \dots, L\}}{\text{stddev}} (o(i, j, t)) = C_{std} \quad \forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, M\}. \quad (4)$$

To summarize the authors suggest to adjust the sensor readout using a linear transform to enforce the equalities (3-4) above. But this is only possible if there is a long camera sequence with enough motion where each sensor sweeps many different parts of the scene. Indeed a small window leads to little or no correction at all since it weakens [ $\mathcal{H}$ ]. A subtle consequence of using long sequences is that the sensor (the contrast changes  $\phi_{(i,j,t)}$ ) is assumed to be constant over the whole sequence. If the user cannot move sufficiently the camera, the convergence will be slow. This means that it is probable that between the beginning and the end of the sequence the  $\phi_{(i,j,t)}$  will have changed because of the sensor drift. It leads to an inaccurate estimation of  $\phi_{(i,j,t)}$ : two pixels whom have not seen the same radiance should not satisfy (3) or (4). Moreover since the sensor is not linear, there will be some residuals. These residuals may be negligible except when the scene change suddenly: the approximation

is wrong, because it is based on past observations. It will take time to update the estimation, depending on  $L$ . The residues of the correction as well as the previous landscape will remain superimposed in the subsequent frames. Those are the “ghost artefacts”. The usual circumvent to those “ghost artefacts” is to restart the learning process (forget some past data) if a new scene appears. Nevertheless, the detection of scene changes may be treacherous particularly if it occurs in a small portion of the image (a new vehicle, etc.) since it may be masked by the non uniformity (see Fig. 1).

A variant like, for instance,<sup>17</sup> adjusts the minimum and the maximum of the readout values, assuming the time histograms observed in each sensor are equals over a long enough time sequence:

$$\text{mean}_{t \in \{1, \dots, L\}}(o(i, j, t)) = C_1 \quad \forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, M\} \quad (5)$$

$$\text{stddev}_{t \in \{1, \dots, L\}}(o(i, j, t)) = C_2 \quad \forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, M\}. \quad (6)$$

This last method is called Constant Range.<sup>18</sup> As pointed out by several authors<sup>5</sup> the length  $L$  of the sequence is a crucial factor of success here. There is not way to tune  $L$  *a priori* and two problems may arise:

- If  $L$  is too small and the estimation is wrong because all sensors have not seen the same landscape ( $[\mathcal{H}]$  is wrong);
- If  $L$  is too large and because of the approximation bias and time drift of the sensor behavior, the previous images may appear as superimposed in the last ones. We retrieve the previously cited “ghost artefact” effect.

There is a way to avoid the “ghost artefacts”,<sup>5</sup> which consists in a reset of the estimation when the scene changes too much. For example, in<sup>5</sup> the authors use a simple threshold to perform scene change detection (but the level of this threshold is not easy to tune in general). In<sup>19</sup> the author state that “slow global motion and strong edges in the scene are the main causes” of the non uniformity. Indeed non consistent with  $[\mathcal{H}:]$  motions and/or bad length  $L$  of the sequence used to enforce (3-4 or 5-6) lead to “ghost artefacts” because of the sensor drift. But, it is only more visible near edges. Indeed the higher dynamic near edges weakens the linear approximation of the correction. The idea of treating edges separately also appears, for example, in.<sup>20</sup> To sum up, all these algorithms require a long exposition time with a varying scene and a serious (and sometimes involuntary) camera motion.

There are numerous implementations and studies<sup>8, 18, 21-27</sup> for these two major algorithms. A recursive filter<sup>16</sup> estimates the parameters of the linear function which approximates the  $\phi_{(i,j,t)}$ , or a Kalman filter is preferred.<sup>28</sup> Other authors<sup>29, 30</sup> propose a neural network based algorithm. Several other variants can be found in.<sup>19</sup> The registration based algorithms<sup>31</sup> consider often only translations<sup>32</sup> (but homographies should be used instead, at least on a static scene). Creating a panorama has been proposed<sup>33</sup> to obtain a ground truth, and to use it as a calibration pattern. However, as pointed out,<sup>34</sup> in presence of the structured fixed pattern noise occurring in most infrared cameras, panoramas won’t lead to a good result. Indeed a mean act as a low pass filter thus low frequencies of the non uniformity will remain in the produced images.

Recently in<sup>6</sup> the authors minimize the total variation of the produced images. They also assume a linear model for the non uniformity. It generalizes<sup>35</sup> but works on image sequences. Thus, it is also subject to “ghost artefacts”. In the following, the presented algorithm works on a single image. Furthermore, in the sequel we will omit the time dependence  $t$  of the non uniformity ( $\phi_{(i,j,t)} := \phi_{(i,j)}$ ).

#### 4. PREVIOUS WORK: THE MIDWAY INFRARED CORRECTION

The goal of this section is to give the background (section 4.1) and details (see sections 4.2,4.3) of our previous work<sup>14</sup> (see<sup>36</sup> for an on line use and implementation in  $C++$ ) as it is the starting point of the improvement presented here in section 5.

## 4.1 The midway histogram equalization method

The midway algorithm was designed initially to correct for gain differences between cameras.<sup>37</sup> It permits to compare two images taken with different cameras more easily after their histograms have been equalized. This algorithm was later extended to flicker correction.<sup>38</sup> The midway equalization achieves much better and smoother results than giving flat (uniform) histograms to images. The idea of the midway equalization is to replace the (arbitrarily chosen) uniform histogram of classic uniform equalization by an histogram depending on the input images. It is optimal in the sense of the Wasserstein (transport) distance; the midway histogram being at equal distance of the histograms of input images. The midway equalization is defined and explained below.

Consider two cumulative histograms  $H_1, H_2$  of two images. The midway cumulative histogram of the corrected image is simply

$$H_{mid}^{-1} := \frac{H_1^{-1} + H_2^{-1}}{2},$$

and this average can be extended to an arbitrary number of images (and to non constant weights). A precise definition of the pseudo inverse is given in section 4.3. Once the midway histogram is computed, a monotone contrast change is applied to images to specify  $H_{mid}$  as their common histograms. Thus, all images get the midway histogram, which is the best compromise between all histograms (see Fig 2).

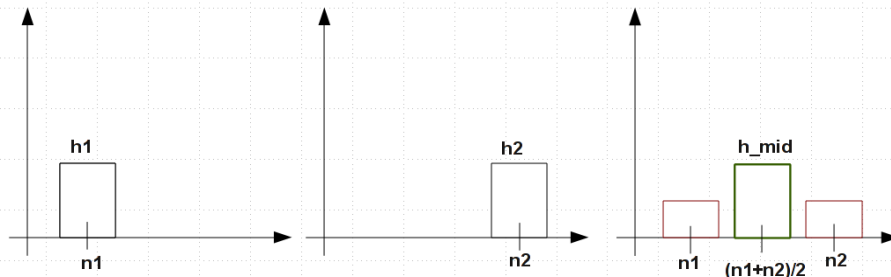


Figure 2. Two histograms  $h_1, h_2$  (left side) and the corresponding midway histogram  $h_{mid}$  (on the right), compared to the direct histogram average, which would create two modes (centered at  $n_1$  and  $n_2$ ) and is therefore wrong. The uniform equalization would destroy the grey level dynamic and create artefacts. It is not a good candidate to get good quality images.

## 4.2 The idea

Since many infrared correction algorithms actually propose to equalize the temporal histograms of each pixel sensor, the midway is quite adapted to get a better result than a simple equalization. Equalization can be based on the fact that single columns (or lines, depending of the readout system) carry enough information by themselves for an equalization. The images being continuous, the difference between two adjacent columns is statistically small, implying that two neighboring histograms should be nearly equal. This hypothesis here is similar to the temporal one [H] but is better suited to the decision to carry the equalization inside the image itself. It does not require any additional *hypothesis* on the non uniformity (linearity, etc.). In other words, the proposition is to transport the histogram of each column (or line) to the midway of histograms of neighboring columns (resp. lines). In presence of strong fixed-pattern-noise (FPN) it will be useful to perform this *sliding midway method* over a little more than two columns, because the FPN is not independent in general.

## 4.3 The midway infrared equalization algorithm (MIRE)

We give here the numerical details to implement the midway infrared equalization algorithm. It is fully automatic, compensate for non linear non uniformity thus, it is well suited to be used in an infrared image denoising chain (as a preprocessing for example). Assume in the sequel that the equalization is performed among the columns of a discrete (8-bits w.l.o.g.) image  $o(i, j) \in \{0, \dots, 255\} \forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, M\}$ . The “midway infrared equalization (MIRE)” algorithm proceeds as follows

For each column  $j \in \{1, \dots, M\}$ ;

1. Compute the cumulative histogram  $H_j$  of each column  $c_j$

$$H_j(l) = \frac{1}{N} \sum_{k=0}^l \sum_{i=1}^N \mathbf{1}_{\{o(i,j)=k\}};$$

2. For each column  $c_j$  compute a local midway histogram  $\tilde{H}_j^{-1} := \sum_{k \in (-n, \dots, n)} g(k) H_{k+j}^{-1}$  using Gaussian weights  $g(k) = g_s(k) = \frac{1}{s\sqrt{2\pi}} e^{-\frac{k^2}{2s^2}}$  with standard deviation  $s$  and  $n = \text{round}(4s)$  where,

$$H_j^{-1}(l) = \min\{z \in \{0, \dots, 255\} / H_j(z) \geq l\};$$

3. Specify the histogram of the column  $c_j$  onto this midway histogram  $\tilde{H}_j$

$$d(i, j) = \tilde{H}_j^{-1}(H_j(o(i, j))) \quad \forall i \in \{1, \dots, N\}.$$

Since we work on images separately the method is not affected by motions or scene changes. This completely avoids “ghost artefacts”<sup>4, 5, 39</sup> and any problem caused by the calibration parameters drifting over time. Notice that contrarily to the classic literature algorithms it does not assume any additional properties (linearity, etc.) on the non uniformity. The histogram specification step is also non linear in general. Thus, we can compensate for fully non linear non uniformity. An algorithm selecting  $s$  is given in the next paragraph.

#### 4.3.1 Automatically fitting the perfect $s$ parameter

The non-uniformity leads to an increased total-variation norm. Hence, following the idea of,<sup>35</sup> the smoothest image is also the one with little or no non-uniformity at all. So the simplest way to find the good ( $s^*$ ) parameter automatically is :

$$s^* = \operatorname{argmin}_s \|I_s\|_{TV\text{-line}}$$

where  $I_s$  is the image processed by the MIRE algorithm described above with the parameter  $s$ . The discrete (line) total variation is defined by  $\|I\|_{TV\text{-line}} = \sum_{i,j} |(\nabla I)_{i,j}|$ ,  $(\nabla I)_{i,j} = (I_{i+1,j} - I_{i,j})$ . The optimization can be done by scanning a broad range of  $s$ . Choose a  $s\_step$  and a  $s\_max$  ( $s\_step = 0.5$  and  $s\_max = 8$  by default in<sup>36</sup> and never lead to unsatisfactory results). Start with  $s = 0$ , repeat : 1) process the image 2) increase  $s$  of  $s\_step$ , then stop when  $s > s\_max$ .

The following ensures a certain safety of the proposed method, a fact that is confirmed by the experience of Fig 3.

**Theorem 1.** If  $H_i$   $i \in \{-n, \dots, n\}$  are  $2n + 1$  cumulative histograms of the same landscape seen by  $2n + 1$  different columns of the sensor, and  $H_{mid}^{-1} = \sum_{j=-n}^n \frac{H_j^{-1}}{2n+1}$  then :

$$\|H_{mid} - H_{true}\|_2 \leq \max_{i \in \{-n, \dots, n\}} (\|H_i - H_{true}\|_2)$$

Moreover if the  $H_i \forall i \in \{-n, \dots, n\}$  from the  $2n + 1$  columns of the sensor are *i.i.d. and centered* on  $H_{true}$  then

$$\|H_{mid} - H_{true}\|_2 \xrightarrow[n \rightarrow \infty]{} 0$$

## 5. THE ADAPTIVE AND DENOISING MIDWAY EQUALIZATION ALGORITHM (ADMIRE)

In this section we describe the novelties proposed to our previous work (see section 4 and<sup>14, 36</sup>). It consists in a modification of the MIRE (see section 4.3) to make it locally adaptive to the image. The need for a locally adaptive scheme is illustrated bellow (see Fig. 5). This modification is detailed in section 5.1. The result of this locally adaptive scheme is nevertheless corrupted by the noise (which may be strong as in Fig. 4). Thus, we shall also embed a denoising scheme in order to increase the signal to noise ratio.



Figure 3. On the left : an uncorrupted test image (boat). On the middle : the result of the MIRE algorithm ( $s^* = 0$ ); the produced image is the same. This experiment was done using<sup>36</sup>. On the right : the result of the locally adaptive variant of MIRE described in section 5.1 ( $s^* = 0$  everywhere in the image). As predicted the algorithms does not make the image worse or create artefacts (safety check). Results on real raw images corrupted with non uniformity are detailed in section 6.

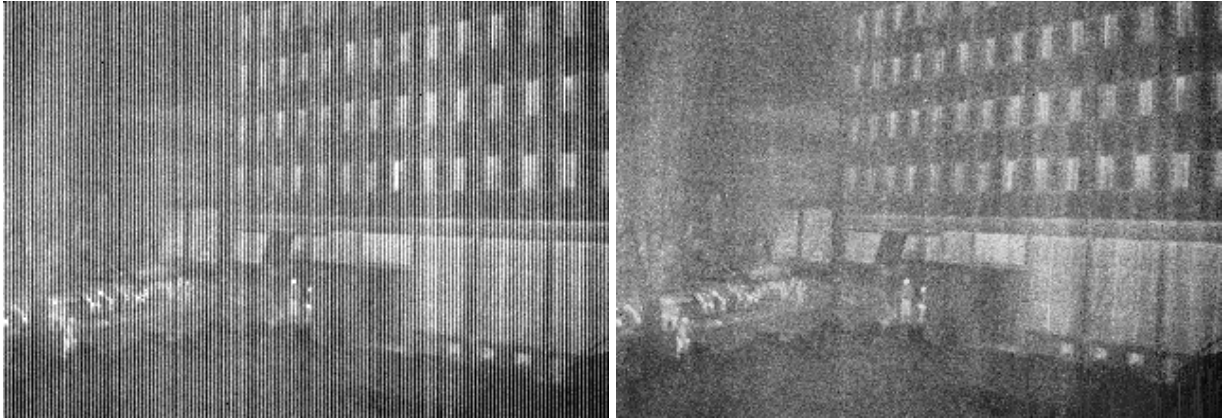


Figure 4. On the left : a real raw image produced by an infrared camera. On the right : the result of the locally adaptive MIRE algorithm (see section 5.1). It is strongly corrupted by noise.

### 5.1 The adaptive midway equalization algorithm

There is no real justification (except simplicity) to keep a constant parameter  $s$  over the whole image. Indeed the image can present different contents and structures in different areas. Thus focusing on different part of the image the best  $s$  parameter may be different. A fact that is illustrated in Fig. 5. Thus, we propose to adapt the  $s$  parameter of the MIRE algorithm locally. The proposed algorithm is :

1. For each  $s$  in  $s_{min} : s_{max}$  (see section 4.3.1) process the image by MIRE;
2. Decompose all images in patches ( $8 \times 8$  patches always used here);
3. For each patch keep the one with smallest  $TV - line$  (the best one as in section 4.3.1);
4. Average and aggregate all patches to get the non uniformity corrected image.

### 5.2 The denoising step : anisotropic $DCT$ threshold on overlapping patches

Since the processed images seem to be corrupted by a strong noise we propose to adapt<sup>40</sup> to perform a good denoising. Thus we propose to modify the  $DCT$  threshold denoising algorithm of<sup>40</sup> which performs a  $DCT$



Figure 5. On the left : a real raw image produced by an infrared camera. Middle : the result of the algorithm with the parameter  $s = 1$ . On the right with  $s = 7.5$ . For example, focusing on a zone in the middle of the image the image with  $s = 7.5$  is nicer but, the zone below poles is bad. On the other hand in the image processed with  $s = 1$  the zone in the middle is still corrupted by the non uniformity. Thus, a fixed  $s$  parameter for the whole image will not lead to the best quality possible everywhere.

threshold on sliding and overlapping patches. It is well suited with the previous step of section 5 which also use patches. Moreover<sup>40</sup> is unaffected\* by residues of non uniformity. We kept  $8 \times 8$  patches as in<sup>40</sup> for the examples given below in section 6. The modification consists of the use of two different thresholds in the direction of the non uniformity and the orthogonal direction. Indeed the coefficients in the direction of the lines should be bigger (for an image corrupted by columns non uniformity noise as in Fig. 1) to denoise more. Indeed this direction is more corrupted by residues of non uniformity. It leads to an anisotropic filtering of the patches. To sum up, we suggest to perform an anisotropic *DCT* threshold on overlapping patches as a final denoising step. Provided two thresholds  $T_i$  and  $T_j$  (denoising strongness) in the  $i$  (lines being indexed by  $i$ ) and  $j$  direction of the image the algorithm is :

1. Decompose the image sliding patches;
2. For each patch :
  - (a) Compute 2D-DCTII transform of the patch;
  - (b) Threshold the DCTII coefficients, with a threshold equal to  $T_j$  in the  $j$  direction and  $T_i$  (lines being indexed by  $i$ ) everywhere else;
  - (c) Calculate inverse 2D-DCT transform of the patch;
  - (d) (normalize by a factor of  $\frac{1}{4patch\_size*patch\_size}$ );
3. Average and aggregate all patches to get the denoised image.

### 5.3 Implementation

The implementation is easy and was done with *C++*. To avoid border effects we used a reflection of the image across borders. A *C++* source code of the MIRE algorithm is available in,<sup>36</sup> it allows on line experiments. The demo performs the MIRE algorithm, permits to see and download the result and shows the  $s^*$  parameter computed by the algorithm. For the denoising step we refer to,<sup>42</sup> where a *C++* source code is available (as well as an on line demo). The overall chain of ADMIRE is :

1. Perform the adaptive midway equalization algorithm described in section 5.1;
2. Perform the anisotropic *DCT* threshold on overlapping patches described in section 5.2.

Of course a temporal extension (to videos) of the proposed method avoiding flicker (and “ghost artefacts”) is possible, using a temporal midway.<sup>38</sup>

---

\*The non local means denoising<sup>41</sup> has issues on some images (depending on the amount of residues of the non uniformity). Indeed residues of non uniformity interfere with patch distances (even after a column by column normalization of patches by their variance.)



## 5.4 Quality analysis; contrast invariant $RMSE$

In many cases reconstruction errors inherent to a method can be quantified using the Root-Mean-Squared-Error  $RMSE(u, u_{est}) := \sqrt{\frac{\int_D |u(x) - u_{est}(x)|^2 dx}{measure(D)}}$ . However, when a small contrast change occurs between the original and the processed image, the  $RMSE$  can become substantial, while the images remain perceptually indistinguishable. For example take an image  $u(m, n)$  defined over a sub-domain  $D \subset \mathbb{Z}^2$ , and another one  $u_{est} = u + 10$ . Then  $RMSE(u, u_{est}) = 10$  is large but does not reflect the quality of the reconstruction  $u_{est}$ . Comparatively, a convolution with, for example, a Gaussian can give a smaller  $RMSE$  while making considerable damage. This bias is avoided by normalizing the images before computing the  $RMSE$ . The principle of the normalization is that two images related to each other by a contrast change are perceptually equivalent. Their distance should reflect this fact and be zero. The midway equalization<sup>37</sup> is best suited for that purpose, because it equalizes the image histogram to a ‘‘midway’’ histogram depending on both images. By the midway operation both images undergo a minimal distortion and adopt exactly the same histogram. Thus we shall define the contrast invariant  $RMSE$  ( $RMSE^{CI}$ ) by

$$RMSE^{CI} = RMSE(u_{est_{mid(u, u_{est})}}, u_{mid(u, u_{est})})$$

where  $mid(u, u_{est})(= mid(u_{est}, u))$  is the midway histogram between  $u$  and  $u_{est}$ .  $u_{mid(u, u_{est})}$  is the image  $u$  specified on the  $mid(u, u_{est})$  histogram (having an histogram equal to  $mid(u, u_{est})$ ) and  $u_{est_{mid(u, u_{est})}}$  is  $u_{est}$  specified on  $mid(u_{est}, u)$ .

## 6. EXPERIMENTS

Simulations of Fig. 6-7 are made using a nonlinear randomly generated model of NU. Results are quantified in term of  $RMSE$  and  $RMSE^{CI}$  (the contrast invariant  $RMSE$  reflects more the intrinsic quality of the image). They confirm the guess of visual improvement in quality. The comparisons with the total variation based method is given in Figs. 8-9. The proposed algorithm outperforms it. The algorithm was run on real raw images, Figs. 10-15, thus for these experiments it is not possible to compute a  $RMSE$  (as the groundtruth is unknown). The proposed algorithm outperforms classic literature methods in real situation on real (non simulated) non uniformity while using only one image.

### 6.1 Total variation based method

Let  $o(i, j) \forall (i, j) \in \{1, \dots, N\} \times \{1, \dots, M\}$  be the observed image. The  $TV - line$  based method<sup>35</sup> looks for a constant  $k(j)$  to add to each column. So

$$\|o(i, j) + k(j)\|_{TV-line}$$

is as small as possible. This boils down to the minimization of  $\sum_i |o(i, j+1) + \delta(j) - o(i, j)|$  for each column  $j$  (notice that this sum involves only the column  $j$  and its neighboring column  $j+1$ ). Then  $k(j+1) = k(j) + \delta(j)$ , where  $k(0) = c$  chosen so that the resulting image has the same mean as the observed image  $o$ . In practice this can be done by :

1. Keep the first column intact ( $j = 0$ );
2. For each  $j \in \{1, \dots, M\}$ ;
  - (a) Minimize  $\sum_{i \in \{1, \dots, N\}} |o(i, j+1) + \delta(j) - o(i, j)|$ , by trying all possible  $\delta(j)$  constants (using the quantification of the image);
3. Add a constant to the whole image so the output has the same mean as  $o$ .



Figure 6. On the left : an image with a strong simulated non linear non uniformity. On the right : the image processed by the proposed algorithm.  $RMSE = 9.6629$ ,  $RMSE^{CI} = 5.7314$ .



Figure 7. On the left : an image with a strong simulated non linear non uniformity. On the right : the image processed by the proposed algorithm.  $RMSE = 8.7100$ ,  $RMSE^{CI} = 7.3411$ .

## 6.2 Comparative experiments

The comparative experiments with the total variation were processed using *Megawave*<sup>†</sup> (resthline module<sup>35</sup>). Experiments on real raw infrared images are shown in Figs. 8-9. The denoising step of ADMIRE has been deactivated for these experiments to allow a fair comparison with the total variation (TV) based method (which does not denoise). ADMIRE always shows a significant improvement on the TV based method and the final

<sup>†</sup>*Megawave* is available at [megawave.cmla.ens-cachan.fr/](http://megawave.cmla.ens-cachan.fr/)

visual quality is very satisfactory.

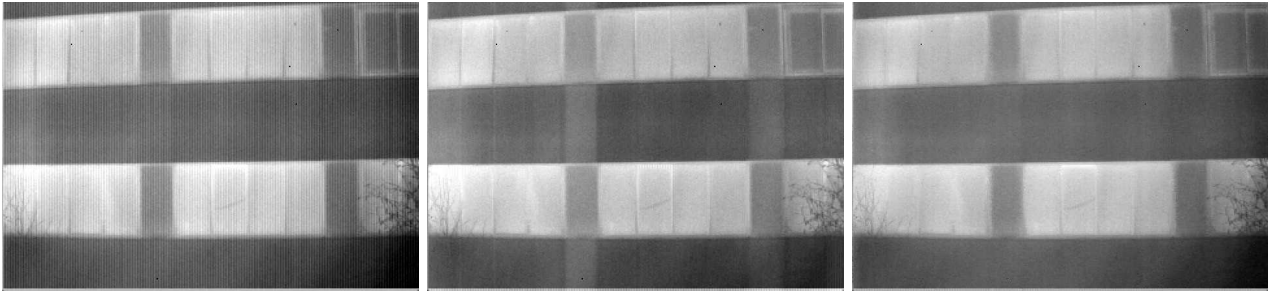


Figure 8. On the left : a real raw image of a building taken by an infrared camera. Middle : the total variation based method. Notice the artefacts created on the concrete stripes. The concrete stripes are at constant temperature thus, it should be at a constant grey level. On the right the proposed method (we deactivate the *DCT* denoising to provide a fair comparison).



Figure 9. On the left : a real raw image of an outdoor scene taken by an infrared camera. Middle : the total variation based method. Notice the artefacts created bellow the poles. On the right the proposed method (we deactivate the *DCT* denoising to provide a fair comparison).

### 6.3 Experiments on real raw images

The subsequent experiments permits to visualize the result of the proposed method on numerous real raw images. We used different types of landscape to visualize the effect of the proposed non uniformity correction on edges, textures, and at different level of time exposure (noise). The conclusion is that the quality is quite satisfactory (see Figs.10-15).

## 7. DISCUSSION AND CONCLUSION

In this paper we start by modeling the image formation chain including the non uniformity and the Poisson (shot) noise terms. From this model we deduce the correct algorithm (chain) to apply to an image in order to perform a good non uniformity correction. We developed an image processing chain to correct for the non uniformity and the noise. A single image locally adaptive non uniformity correction was designed. It can compensate for fully non linear non uniformity, without any parametric model on the non uniformity side. It does not require motion, or motion compensation, does not need a test pattern or calibration and does not produce any “ghost artifact”. A state of the art denoising algorithm was modified according to the model to obtain a non uniformity correction chain. Evaluations using both simulated and real raw images show that the approach performs an efficient non uniformity correction in term of *RMSE*, contrast invariant *RMSE<sup>CI</sup>* and visual image quality. Comparisons were made with a total variation based method. The conclusion is that a single image, ghost-less and non linear non uniformity correction is not only possible but efficient.

The next steps are first to obtain a robust noise estimation following, for example, the “percentile method”



Figure 10. On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

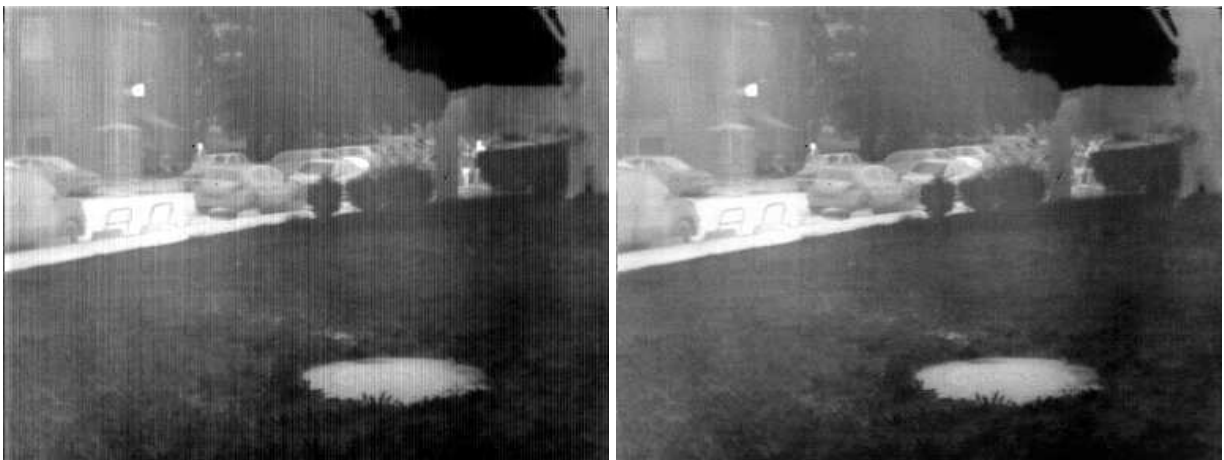


Figure 11. On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.



Figure 12. On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

described in.<sup>43</sup> Notice that to denoise automatically (without have to tune the  $T_i$  and  $T_j$  thresholds) theses images a directional estimation of the noise variances are needed. The second step is to take advantage of movies



Figure 13. On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

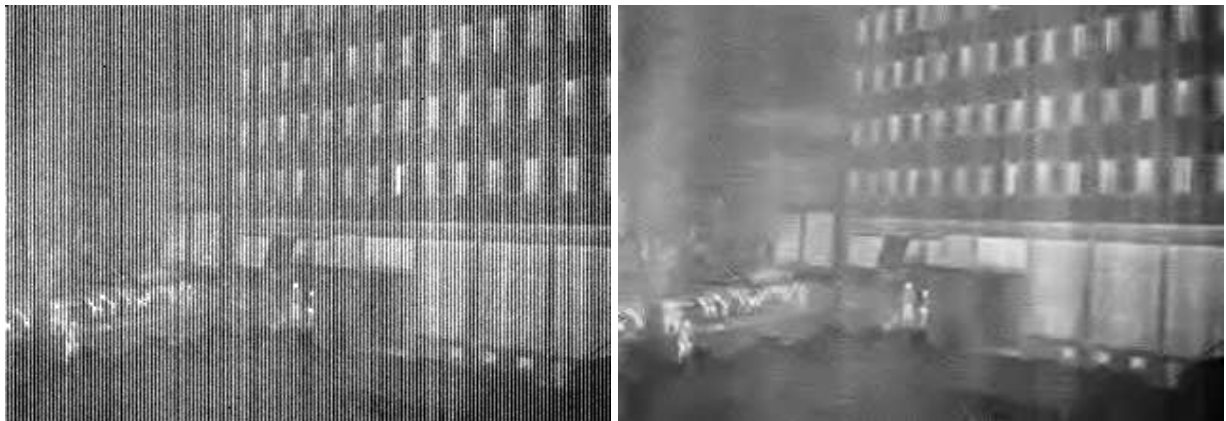


Figure 14. On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

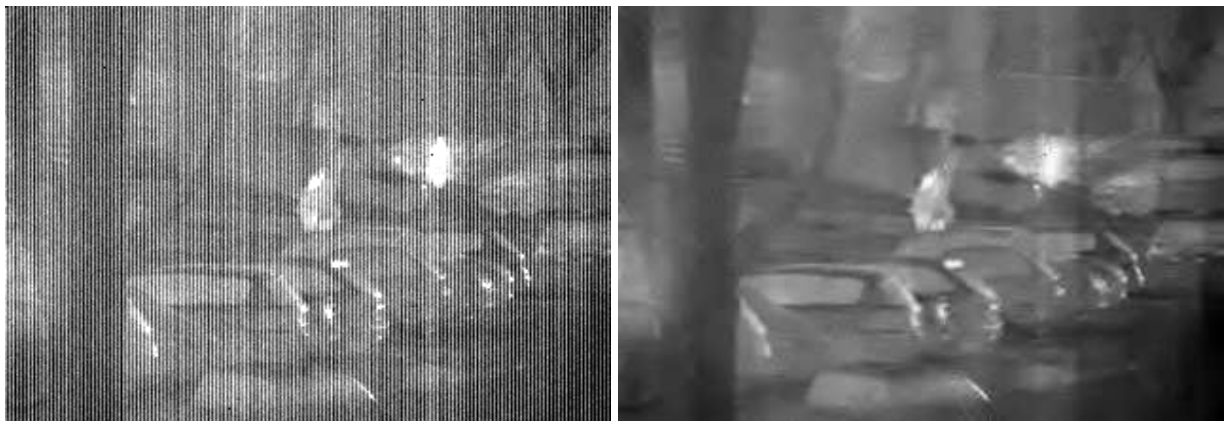


Figure 15. On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

to increase the quality while (still) ensuring the absence of *any* “ghost artefacts”. Indeed, if needed, the scene change detection is way easier to perform just before the *DCT* denoising step.

## ACKNOWLEDGMENTS

We thank the UCLA math department and the Délégation Générale pour l'Armement (DGA) for supporting this work. We are grateful to Stéphane Landeau for providing us the infrared cameras used for the experiments.

## REFERENCES

- [1] Biberman, L. M., ed., [*Electro-Optical Imaging : system performance and modeling*], SPIE Press (october 2000).
- [2] Scribner, D., Kruer, M., and Killiany, J., “Infrared focal plane array technology,” *Proceedings of the IEEE* **79**(1), 66–85 (1991).
- [3] Milton, A., Barone, F., and Kruer, M., “Influence of nonuniformity on infrared focal plane array performance,” *Optical Engineering* **24**(5), 855–862 (1985).
- [4] Rossi, A., Diani, M., and Corsini, G., “Temporal statistics de-ghosting for adaptive non-uniformity correction in infrared focal plane arrays,” *Electronics letters* **46**(5), 348–349 (2010).
- [5] Harris, J. and Chiang, Y., “Minimizing the ghosting artifact in scene-based nonuniformity correction,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **3377**, 106–113 (1998).
- [6] Vera, E., Meza, P., and Torres, S., “Total variation adaptive scene-based nonuniformity correction,” in [*Imaging Systems*], Optical Society of America (2010).
- [7] Zuo, C., Chen, Q., Gu, G., and Sui, X., “Scene-based nonuniformity correction algorithm based on interframe registration,” *Journal of the Optical Society of America. A, Optics, image science, and vision* **28**(6), 1164–1176 (2011).
- [8] Qian, W., Chen, Q., Bai, J., and Gu, G., “Adaptive convergence nonuniformity correction algorithm,” *Applied Optics* **50**(1), 1–10 (2011).
- [9] Zuo, C., Chen, Q., Gu, G., and Qian, W., “New temporal high-pass filter nonuniformity correction based on bilateral filter,” *Optical Review* **18**(2), 197–202 (2011).
- [10] Scribner, D., Sarkady, K., Caulfield, J., Kruer, M., and Katz, G., “Nonuniformity correction for staring ir focal plane arrays using scene-based techniques,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **1308**, 224–233 (1990).
- [11] Riou, O., Berrebi, S., and Bremond, P., “Nonuniformity correction and thermal drift compensation of thermal infrared camera,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **5405**, 294–302 (2004).
- [12] Jin, W., Liu, C., and Xiu, J., “Infrared nonuniformity correction and radiometric calibration technology using U-shaped blackbody,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **8194**, 819405 (2011).
- [13] Houchin, J. and Parulski, K., “Method and apparatus for pixel non-uniformity correction,” (Sept. 10 1991). US Patent 5,047,861.
- [14] Tendero, Y., Gilles, J., Landeau, S., and Morel, J., “Efficient single image non-uniformity correction algorithm,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **7834**, 10 (2010).
- [15] Friedenber, A. and Goldblatt, I., “Nonuniformity two-point linear correction errors in infrared focal plane arrays,” *Optical Engineering* **37**, 1251 (1998).
- [16] Harris, J. and Chiang, Y., “Nonuniformity correction of infrared image sequences using the constant-statistics constraint,” *Image Processing, IEEE Transactions on* **8**(8), 1148–1151 (1999).
- [17] Pezoa, J., Torres, S., Córdova, J., and Reeves, R., “An enhancement to the constant range method for nonuniformity correction of infrared image sequences,” *Progress in Pattern Recognition, Image Analysis and Applications*, 259–279 (2004).
- [18] Torres, S., Vera, E., Reeves, R., and Sobarzo, S., “Scene-based non-uniformity correction method using constant range: Performance and analysis,” in [*Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, IX: 224–229*], (2002).

- [19] Rossi, A., Diani, M., and Corsini, G., “A comparison of deghosting techniques in adaptive nonuniformity correction for ir focal-plane array systems,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], (2010).
- [20] Zhang, T. and Shi, Y., “Edge-directed adaptive nonuniformity correction for staring infrared focal plane arrays,” *Optical Engineering* **45**, 016402 (2006).
- [21] Scribner, D., Sarkady, K., Krueer, M., Caulfield, J., Hunt, J., Colbert, M., and Descour, M., “Adaptive retina-like preprocessing for imaging detector arrays,” in [*Neural Networks, 1993., IEEE International Conference on*], 1955–1960, IEEE (1993).
- [22] Narendra, P., “Reference-free nonuniformity compensation for IR imaging arrays,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **252**, 10–17 (1980).
- [23] Goyal, P., “NUC algorithm by calculating the corresponding statistics of the decomposed signal,” *International Journal on Computer Science and Technology* **1**(2) (2010).
- [24] Hart, R. and O. Thomas, T. O. L., “A study of non-uniformity correction methods for staring array ir detectors,” 1st EMRS DCT Technical Conference, Edinburgh (2004).
- [25] Hayat, M., Torres, S., Cain, S., and Armstrong, E., “Model-based real-time nonuniformity correction in focal plane array detectors,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **3377**, 122–132 (1998).
- [26] Vera, R. and Torres, I., “Ghosting reduction in adaptive nonuniformity correction of infrared focal-plane array image sequences,” in [*Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*], **2**, II–1001, IEEE (2003).
- [27] Vera, E. and Torres, S., “Fast adaptive nonuniformity correction for infrared focal-plane array detectors,” *EURASIP Journal on Applied Signal Processing* **13**, 1994–2004 (2005).
- [28] Torres, S. and Hayat, M., “Kalman filtering for adaptive nonuniformity correction in infrared focal-plane arrays,” *Journal of the Optical Society of America. A, Optics, image science, and vision* **20**(3), 470 (2003).
- [29] Scribner, D., Sarkady, K., Krueer, M., Caulfield, J., Hunt, J., and Herman, C., “Adaptive nonuniformity correction for IR focal-plane arrays using neural networks,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **1541**, 100–109 (1991).
- [30] Torres, S., Vera, E., Reeves, R., and Sobarzo, S., “Adaptive scene-based nonuniformity correction method for infrared-focal plane arrays,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **5076**, 130–139 (2003).
- [31] Tzimopoulou-Fricke, S. and Lettington, A., “Scene-based techniques for nonuniformity correction of infrared focal plane arrays,” in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **3436**, 172–183 (1998).
- [32] Cain, S., Armstrong, E., and Yasuda, B., “Joint estimation of image, shifts, and nonuniformities from IR images,” in [*Infrared Information Symposium (IRIS) on Passive Sensors*], (1997).
- [33] Hardie, R., Hayat, M., Armstrong, E., and Yasuda, B., “Scene-based nonuniformity correction with video sequences and registration,” *Applied Optics* **39**(8), 1241–1250 (2000).
- [34] Zhao, W. and Zhang, C., “Efficient scene-based nonuniformity correction and enhancement,” in [*Image Processing, 2006 IEEE International Conference on*], 2873–2876, IEEE (2006).
- [35] Moisan, L., “Resthline.” Computer program, MegaWave2 Modulus (2007).
- [36] Tendero, Y., Landeau, S., and Gilles, J., “Non-uniformity correction of infrared images by midway equalization,” *Submitted to Image Processing On Line*, 1–23 (2012).
- [37] Delon, J., “Midway image equalization,” *Journal of Mathematical Imaging and Vision* **21**(2), 119–134 (2004).
- [38] Delon, J., “Movie and video scale-time equalization application to flicker reduction,” *Image Processing, IEEE Transactions on* **15**(1), 241–248 (2006).
- [39] Hardie, R., Baxley, F., Brys, B., and Hytla, P., “Scene-based nonuniformity correction with reduced ghosting using a gated lms algorithm,” *Optics express* **17**(17), 14918–14933 (2009).
- [40] Coifman, R. and Donoho, D., “Translation-Invariant De-Noising,” *Wavelets and statistics* **103**, 125 (1995).
- [41] Buades, A., Coll, B., and Morel, J., “A non-local algorithm for image denoising,” in [*Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*], **2**, 60–65, Ieee (2005).

- [42] Guoshen Yu, G. S., “DCT image denoising: a simple and effective image denoising algorithm,” *Image Processing On Line* (2011).
- [43] Lebrun, M., Colom, M., Buades, A., and Morel, J., “Secrets of image denoising cuisine,” *Acta Numerica. To appear.* , 81 (2012).