Efficient single image non-uniformity correction algorithm

Y. Tendero [a], J. Gilles[a,c], S. Landeau[c], and J.M. Morel [a]

[a] Centre de Mathématiques et de Leurs Applications (CMLA)
École Normale Supérieure de Cachan - 61 av du Pdt Wilson 94235 Cachan Cedex France.
[c] Délégation Générale pour l'Armement, Centre d'Expertise Parisien - 7 Rue des Mathurins,
92221 Bagneux.

ABSTRACT

This paper introduces a new way to correct the non-uniformity (NU) in uncooled infrared-type images. The main defect of these uncooled images is the lack of a column (resp. line) time-dependent cross-calibration, resulting in a strong column (resp. line) and time dependent noise. This problem can be considered as a 1D flicker of the columns inside each frame. Thus, classic movie deflickering algorithms can be adapted, to equalize the columns (resp. the lines). The proposed method therefore applies to the series formed by the columns of an infrared image a movie deflickering algorithm. The obtained single image method works on static images, and therefore requires no registration, no camera motion compensation, and no closed aperture sensor equalization. Thus, the method has only one camera dependent parameter, and is landscape independent. This simple method will be compared to a state of the art total variation single image correction on raw real and simulated images. The method is real time, requiring only two operations per pixel. It involves no test-pattern calibration and produces no "ghost artifacts".

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

1. INTRODUCTION

Infrared (IR) imaging has proved to be a very efficient tool in a wide range of industry, medical, and military applications. IR cameras are used to measure temperatures, IR signatures, detection, etc. However, the performance of the imaging system is strongly affected by a random spatial response of each pixel sensor. Under the same illumination the readout of each sensor is different. This is due to mismatches in the fabrication process, among other issues.[1] Furthermore for uncooled cameras the problem is even worse because the sensor response non-uniformity is not stationary and slowly drifts in time. For this kind of camera a periodic update of the non-uniformity-correction (NUC) is required.

A good non-uniformity-correction is a key success factor for any post processing such as pattern recognition, image registration, etc. To get the rid of the non-uniformity, two main kinds of methods have been developed:

- Calibration based techniques consist in an equalization of the response to an uniform black body source radiation. 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).
- Scene based techniques, involving motion compensation or temporal accumulation. Such methods are complex and require certain observation conditions.

The perturbation model is

$$z_t(X) = f_{(X,t)}(u0_t(X)) + \eta_t(X)$$

where (X is the position and t is the time for the following) $z_t(X)$ is the observed value, $u0_t(X)$ is the ideal landscape, $f_{(X,t)}$ is the (unknown) transfer function of the sensor, and $\eta_t(X)$ is a random sensor Poisson noise.

Further author information: (Send correspondence to Y. Tendero)
E-mail: tendero@cmla.ens-cachan.fr, Phone: +33 1 47 40 59 48

A non-uniformity correction algorithm aims at discovering $f_{(X,t)}$ or $u0_t(X)$ for each X and t. In this paper we propose a single frame based algorithm and show that motion compensation or accumulation algorithms are not necessary to achieve a good image quality. However, the proposed method can be viewed as a first step fostering the success of more sophisticated motion based correction algorithms. These are slow while the proposed algorithm is real time, and the obtained quality after a single frame correction might be sufficient for many uses.

The paper is organized as follows. Section 2 presents related works. The new algorithm is described in section 3. Experiments on simulated, real, cooled and uncooled images are described in section 4. Section 5 contains a discussion. Possible improvements are envisaged in section 6.

2. ANTERIOR WORK

Numerous algorithms have been reported in the literature to remove the fixed-pattern-noise caused by the lack of a cross-column sensor equalization. Some algorithms estimate the sensor parameter and others attempt at recovering the true landscape. Most of them use a simplified (linear) model for the transfer function of the pixel sensor:

$$z_t(X) = u0_t(X)g_t(X) + b_t(X) + \eta_t(X),$$

where where (X is the position and t is the time for the following) $z_t(X)$ is the observed value, $u0_t(X)$ is the landscape, $g_t(X)$ and $b_t(X)$ are the gains/ offsets (in place of $f_{(X,t)}$) and $\eta_t(X)$ is the random noise. (Nevertheless, the true transfer function is non linear.) These algorithms process a sequence of images $(z_t)_{t\in 1,...,N}$, not a single frame. The proposed algorithm uses no registration, hence we will focus on single frame algorithms. There are methods[2] suggesting to equalize the mean and standard deviation (*stddev*) of each pixel sensor by a linear transform. 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

$$\begin{cases} \underset{t\in(1,...N)}{mean}(z_t(X)) & = C_m \ \forall\ X \\ \underset{t\in(1,...N)}{stddev}(z_t(X)) & = C_{std}\ \forall\ X. \end{cases}$$

So the authors suggest to adjust the sensor readout using a linear transform to obtain the equalities 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.

A variant[3] adjusts the minimum and the maximum of the readout values, assuming the time histograms observed in each sensor to become equal over a long enough time sequence:

$$\begin{cases} \underset{t\in(1,...N)}{min}(z_t(X)) & = C_1\ \forall\ X \\ \underset{t\in(1,...N)}{max}(z_t(X)) & = C_2\ \forall\ X. \end{cases}$$

This last method is called Constant Range.[4] As pointed out by several authors[5] the length N of the sequence is a crucial factor of success here. Two problems may arise:

- If N is too small and the estimation is wrong because all sensors have not seen the same landscape;
- If N is too large and because of the approximation bias and time drift of the sensor behavior, the previous images may appear as ghosts in the last ones. This undesirable effect is known a "ghost artifact".

There is a way to avoid the ghost artifacts,[5] which consists in a reset of the estimation when the scene changes too much. Their[5] paper uses a simple threshold to perform scene change detection. But again, all this requires a long exposition time with a varying scene or a serious camera motion.

There are several implementations for these two major algorithms. A recursive filter[2] estimates the parameters of the linear function which approximates the S-shaped transfer function of the sensor, or a Kalman filter[6] is preferred. Other authors[7,8] propose a neural network based algorithm, which requires a serious computational power and is definitely not real time. The registration based algorithms[9] consider often only translations (but homographies should be used instead, at least on a static scene). Creating a panorama has been proposed[10] to obtain a ground truth, and to use it as a calibration pattern. However, as pointed out,[11] in presence of the structured fixed-pattern-noise occurring in most IR cameras, the panorama won't lead to a good result.

3. MIDWAY INFRARED CORRECTION

3.1 The midway histogram equalization method

The midway algorithm was designed initially to correct for gain differences between cameras.[12] 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.[13]
Consider two cumulative histograms H_1, H_2. The midway cumulative histogram of the corrected image is simply

$$Hmid^{-1} = \frac{H_1^{-1} + H_2^{-1}}{2},$$

and this average can be extended to an arbitrary number of images. Once the midway histogram is computed, a monotone contrast change is applied to image to specify H as its histogram. Thus, all images get the midway histogram, which is the best compromise between all histograms (see Fig 1).

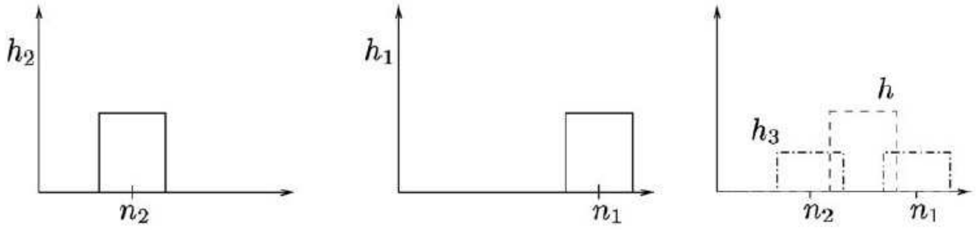

Figure 1. Two histograms h_1, h_2 (left side) and the corresponding midway histogram h (on the right), compared to the direct histogram average, which would create two modes and is therefore wrong.

3.2 The idea

Since many IR 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. Yet, we propose a still much simpler strategy. 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 are nearly equal. This hypothesis here is similar to the temporal one [$\mathcal{H}$] but is better suited to the decision to carry the equalization inside the image itself. So 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.

Assume in the sequel that the equalization is performed with columns. The proposed algorithm proceeds as follows.

Midway Infrared Equalization (MIRE)

- Compute the cumulative histogram H_i of each column c_i;
- For each column c_i compute a local midway histogram $\tilde{H}_{mid}(i)^{-1} := \sum_{j\in(-N,...,N)} \Phi(j) H_{i+j}^{-1}$ using Gaussian weights $\Phi = \Phi_\sigma$ with std-dev σ average;
- Specify the histogram of the column c_i onto this midway histogram $\tilde{H}_{mid}(i)$.

The choice of the standard deviation σ of the Gaussian depends only on the camera, and not on the landscape. Thus, it can be fixed once and for ever for each camera. Since we work on images separately the method is not affected by motions or changes of scene, which completely avoids "ghost artifacts" and any problem caused by the calibration parameters drifting over time. A good σ is simply obtained by

- Trying with a small parameter;
- Increasing it till a good visual image quality is reached.

Yet an automatic method for estimating σ and obtaining a parameterless methods is as follows.

Automatically fitting the perfect parameter
The non-uniformity leads to an increased total-variation norm. Hence the smoothest image is also the one with little or no non-uniformity at all. So the simplest way to find the good parameter automatically is :

$$\sigma^* = argmin_\sigma ||I_\sigma||_{TV}{}^*,$$

where I_σ is the image processed by MIRE with the parameter σ. The optimization could be done by a dichotomy on σ. See Fig 10 for an illustration of this.

Theorem 1. If h_i $i \in 1, ..., N$ are N histograms of the same landscape seen by N different columns of the sensor, and $H_{mid} = \sum_{j=1}^{N} \frac{H_j^{-1}}{N}$ then :

$$||h_{mid} - h_{true}||_2 \leq \max_{i\in(1,...,N)} (||h_i - h_{true}||_2)$$

Moreover if the h_i $\forall i \in (1, ..., N)$ from the N columns of the sensor are *i.i.d. and centered* on h_{true} then

$$||hmid - h_{true}||_2 \underset{N\to\infty}{\to} 0$$

3.3 Implementation

The implementation is easy and was done with *Matlab*. To avoid border effects we used a reflection of the image across borders. The computation times are shown for several image sizes. An on-line demo will be shortly available at www.ipol.im.

Times are shown in seconds on a core duo T7250 running Ubuntu and *Matlab*. We used *Timeit* (written by S. Eddins) to avoid time variation of the multitasking OS.
Of course a temporal extension of the algorithm to avoid temporal flicker is possible, using a temporal midway.[13]

[*] see 4.1 for $||.||_{TV}$ definition.

Image size	512*512	320*220
Seconds	2.8	1.2

Figure 2. Computation time for various sizes and quantifications in seconds (using Matlab). This time could be made to real with any standard processor.

3.4 Quality analysis

Our first criterion is the visual image quality. In the simulated cases the results will be evaluated by the RMSE,

$$RMSE(I,\tilde{I}) = \sqrt{\frac{\sum_{i,j} |I(i,j) - \tilde{I}(i,j)|^2}{M.N}},$$

where I is the groundtruth image, $\tilde{I}$ is the restored one and M, N are the image side lengths.

4. EXPERIMENTS

4.1 Total variation based method

Let $z_t(x,y)$ be the acquired image. The TV based method[14] looks for a constant $k(y)$ to add at each column. So

$$||z_t(x,y) + k(y)||_{TV}$$

is as small as possible, where $||I||_{TV} = \sum_{i,j} |(\nabla I)_{i,j}|$ and $(\nabla I)_{i,j} = \begin{pmatrix} I_{i+1,j} - I_{i,j} \\ I_{i+1,j} - I_{i,j} \end{pmatrix}$. So this amounts to the simple minimization of $\sum_x |z_t(x,y+1) + \delta(y) - z_t(x,y)|$ for each column y. Then $k(y+1) = k(y) + \delta(y)$, where $k(0) = c$ chosen so that the resulting I_t^{TV} and the input images z_t have the same mean.

4.2 Comparative experiments

Simulations (Figs 3-4) are made using a linear randomly generated model of NU. The comparative experiments of MIRE with Total Variation (TV) were processed using a *Megawave* [†] (resthline module[14]). Results are quantified in term of RMSE and confirm the guess of visual improvement in quality.

Real experiments are shown using cooled (Fig 5) and uncooled (Fig 6) cameras. For comparison purpose images are shown with the same variance in every experiments.

MIRE always shows a significant improvement on TV and the final visual quality is overall very satisfactory.

5. DISCUSSION AND CONCLUSION

In this paper a new way to correct for the uncooled IR non-uniformity was proposed. Evaluations using both simulated and real images –from both cooled and uncooled cameras– show that the approach performs an efficient non-uniformity correction (in term of RMSE and visual image quality). Comparison was made with a total variation based method. This simple algorithm is well suited for a parallel implementation, since each column could be processed independently from the others. Furthermore since we process each image of the stream separately ”ghost artifacts” are not present and the velocity of the parameter drift insignificant.

Eventually the output seems to be more corrupted with gaussian temporal noise than with residues of unperfect correction of the non-uniformity. This enables the application of any standard image denoising algorithm, such as NL-Means or the wavelet thresholding, etc. See Fig 8. The only failure case we met, shown in Fig 7, appeared with a small (64*64) simulated textured image. There were not enough bins in the histograms to equalize. The results could still be enhanced by using a registration technique for badly corrupted images. This extension is envisaged in section 6.

[†] *Mégawave* is available at megawave.cmla.ens-cachan.fr/

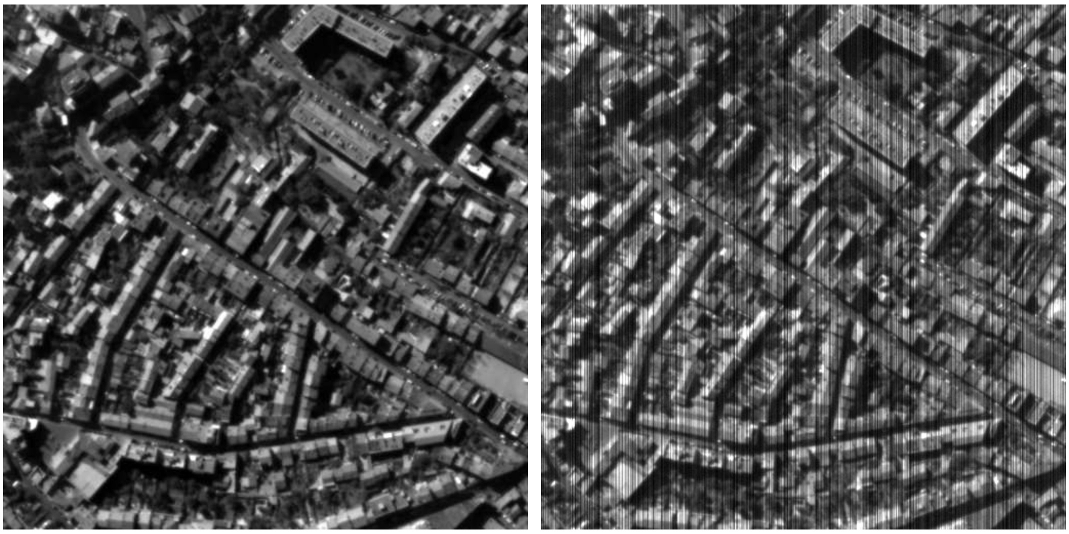

Figure 3. Image 1 : The groundtruth (left) the simulated FPN (right, RMSE=0.1932).

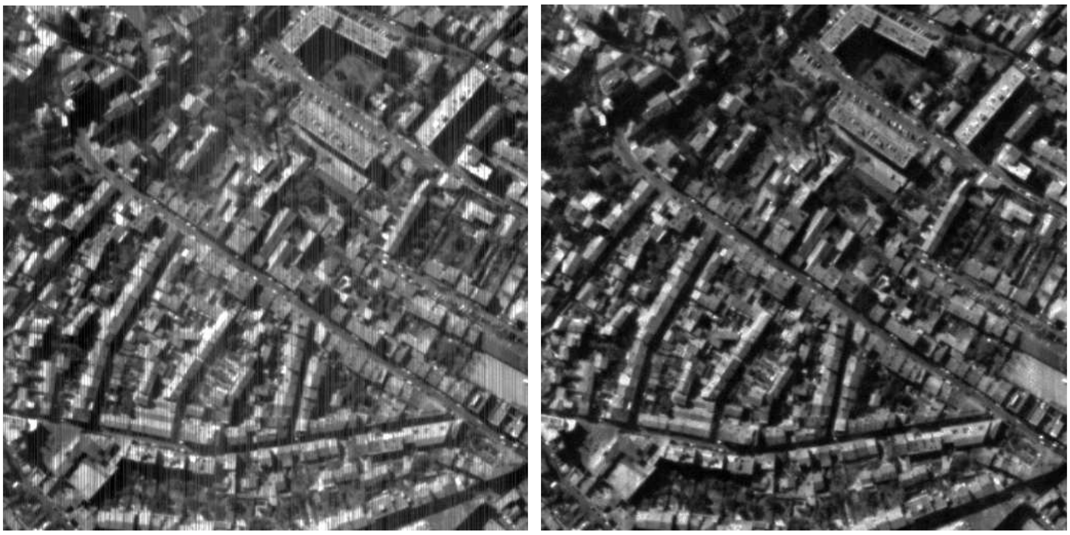

Figure 4. The TV based method (left, RMSE=0.1817), MIRE (right, RMSE=0.1715).

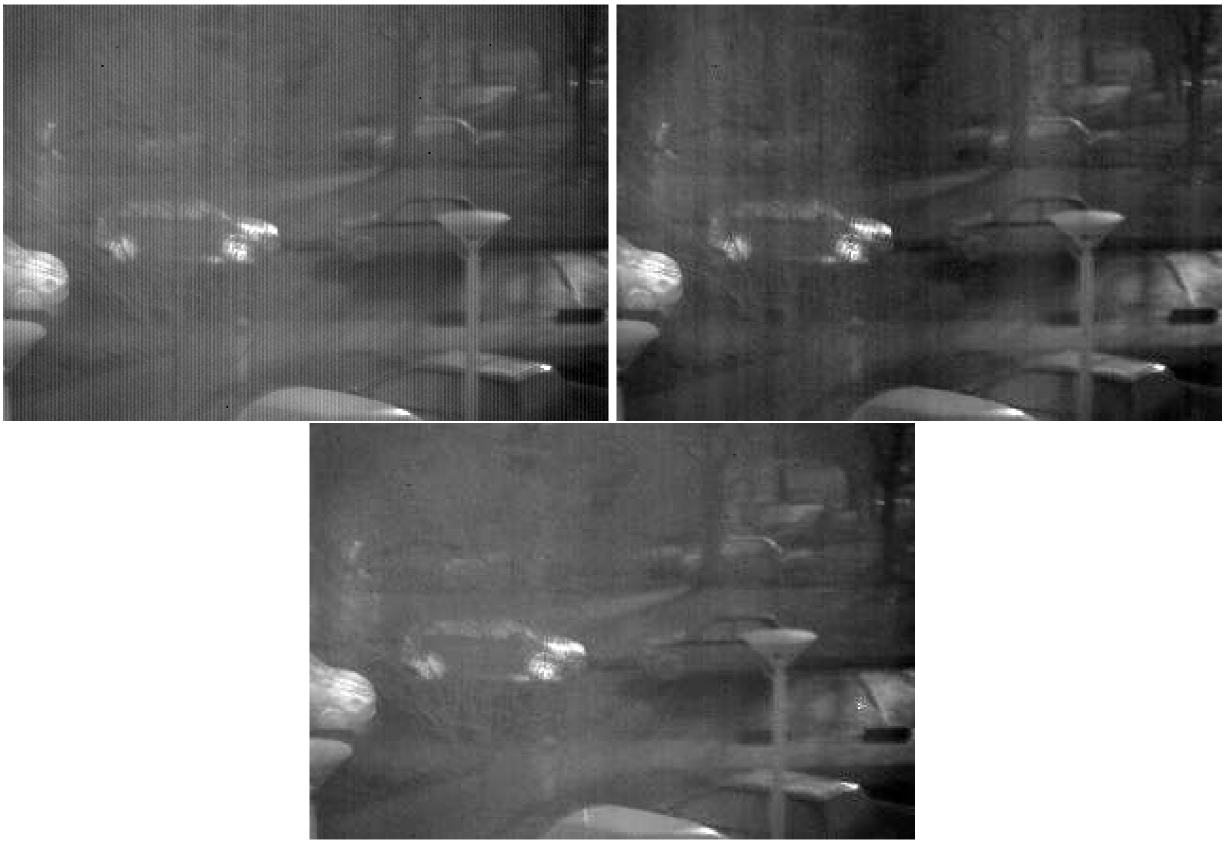

Figure 5. Top left : RAW (cooled camera), top right : TV based method, at the bottom : MIRE.

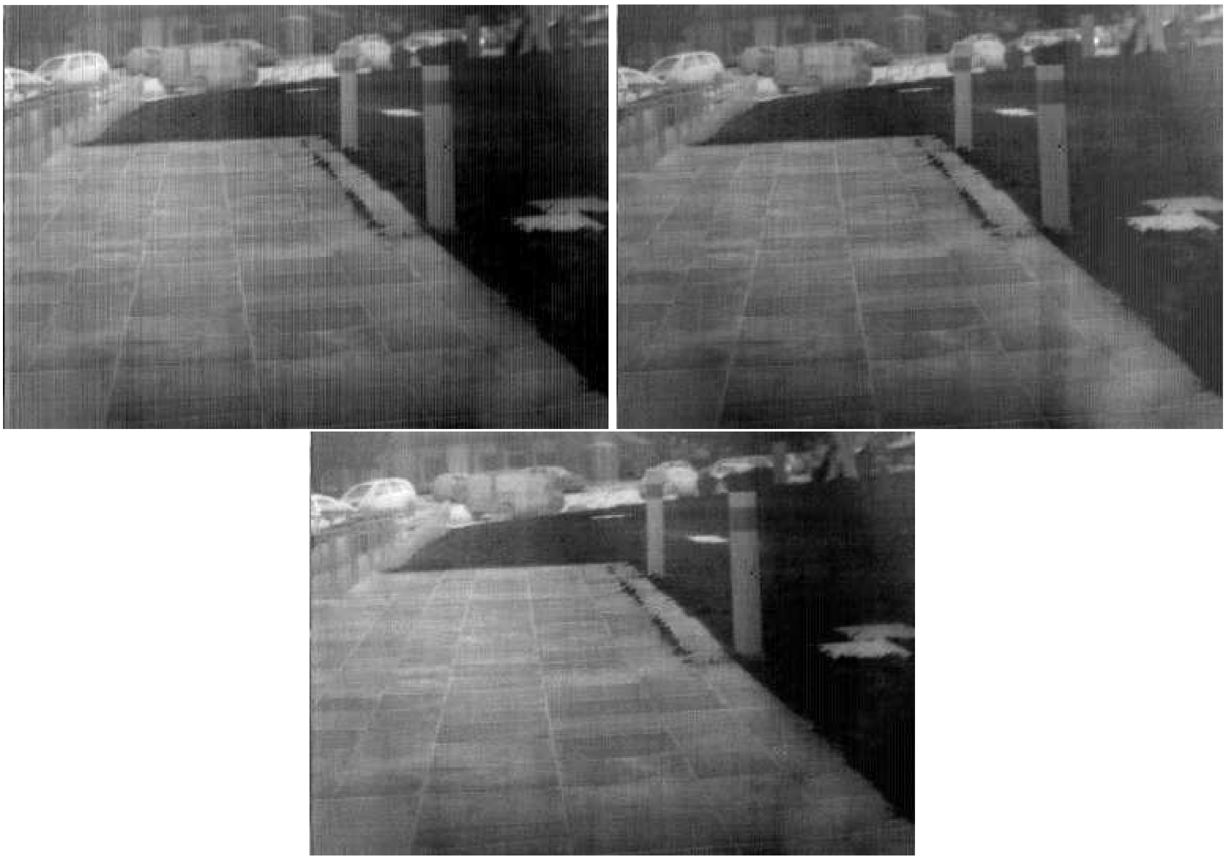

Figure 6. Top left : RAW (cooled camera), top right : TV based method, at the bottom : MIRE.

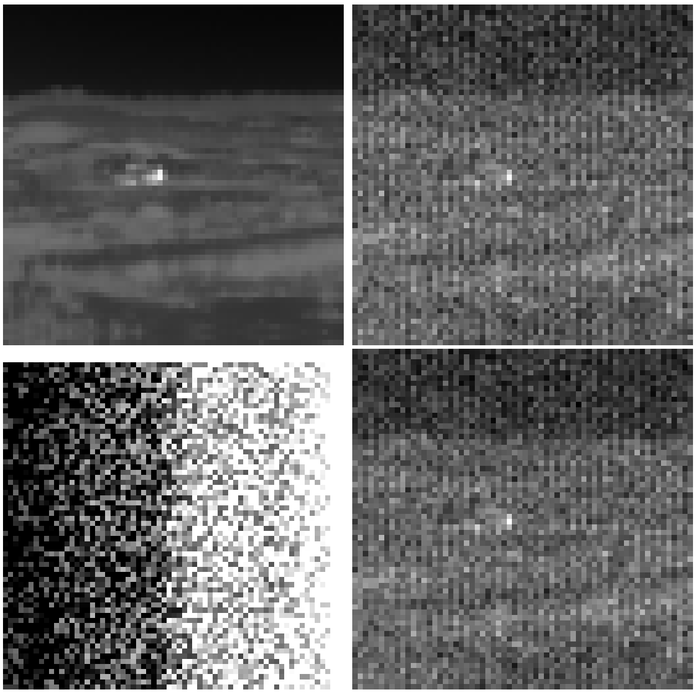

Figure 7. The only case where MIRE failed, on a textural 64x64 image. Top-left the groundtruth, top right the observed (RMSE:0.7033). Bottom left : the TV method (RMSE:1.2681), bottom right : MIRE (RMSE:0.6833).

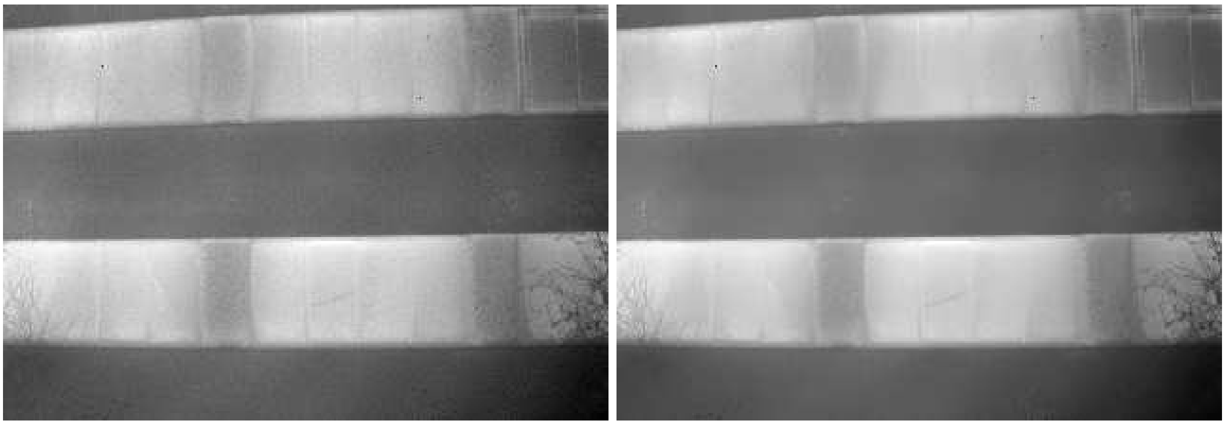

Figure 8. Before and after NL-means denoising, used with default parameters, available at www.ipol.im. Both are X2 zoomed.

6. FUTURE WORK

Here is how MIRE could be combined with motion estimation:

1. First use MIRE
2. Apply a time registration
3. For each line parallel to the motion proceed as follows:
    - Choose a pixel as a reference.
    - Use motion to propagate this information in the direction of the motion.
    - Stop after the number of points is sufficient to the estimation of the non uniformity.

At this step for each pixel sensor we have several points to estimate the transfer function. Hence we could perform any kind of interpolation to estimate a complete transfer function. Then we could compensate for linear as well as non linear uniformity since with N images we will know up to N points of the function. We don't need to perform a panorama to estimate the landscape like in.[10] It is another point of view on the problem, since these authors focus on estimating the landscape (difference of response between the perfect sensor and the real one) while the envisaged method is to obtain an estimation of the uniformity directly (more precisely an estimation of the difference of uniformity between an arbitrary sensor and the others).

If we get an image with strong lining artifact in the motion direction, we have two possibilities : either to use a new motion along another direction or using a single frame algorithm like MIRE.

Fig 9 presents some results on simulated images. We simulated a movie with a (pixelian) translational motion and a NU. The NU remained constant for the whole sequence and *temporal* gaussian noise was added to each image. Then we applied step 3 (assuming the motion is known).

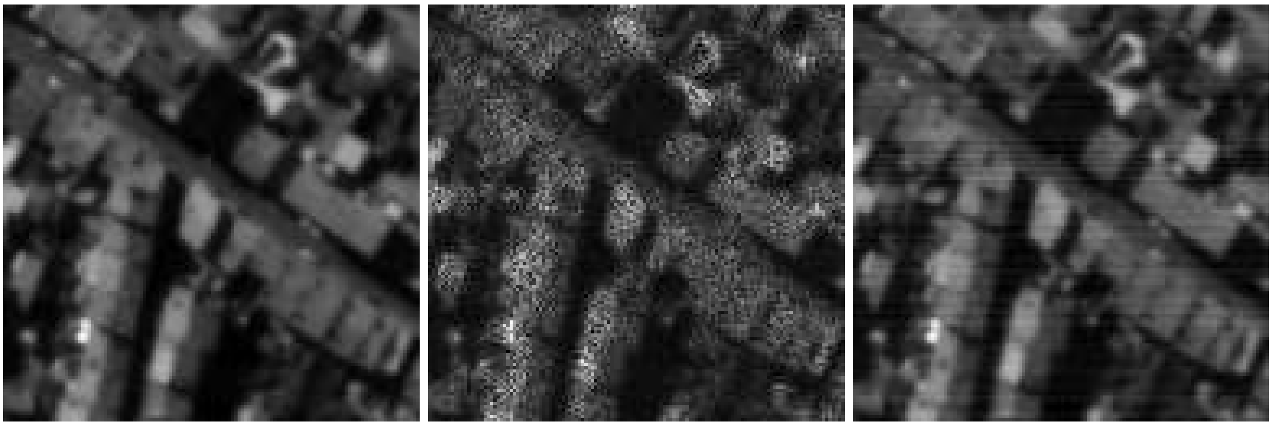

Figure 9. Using the motion, from left to right : the groundtruth, observed (one in the whole sequence), restored.

7. APPENDIX.

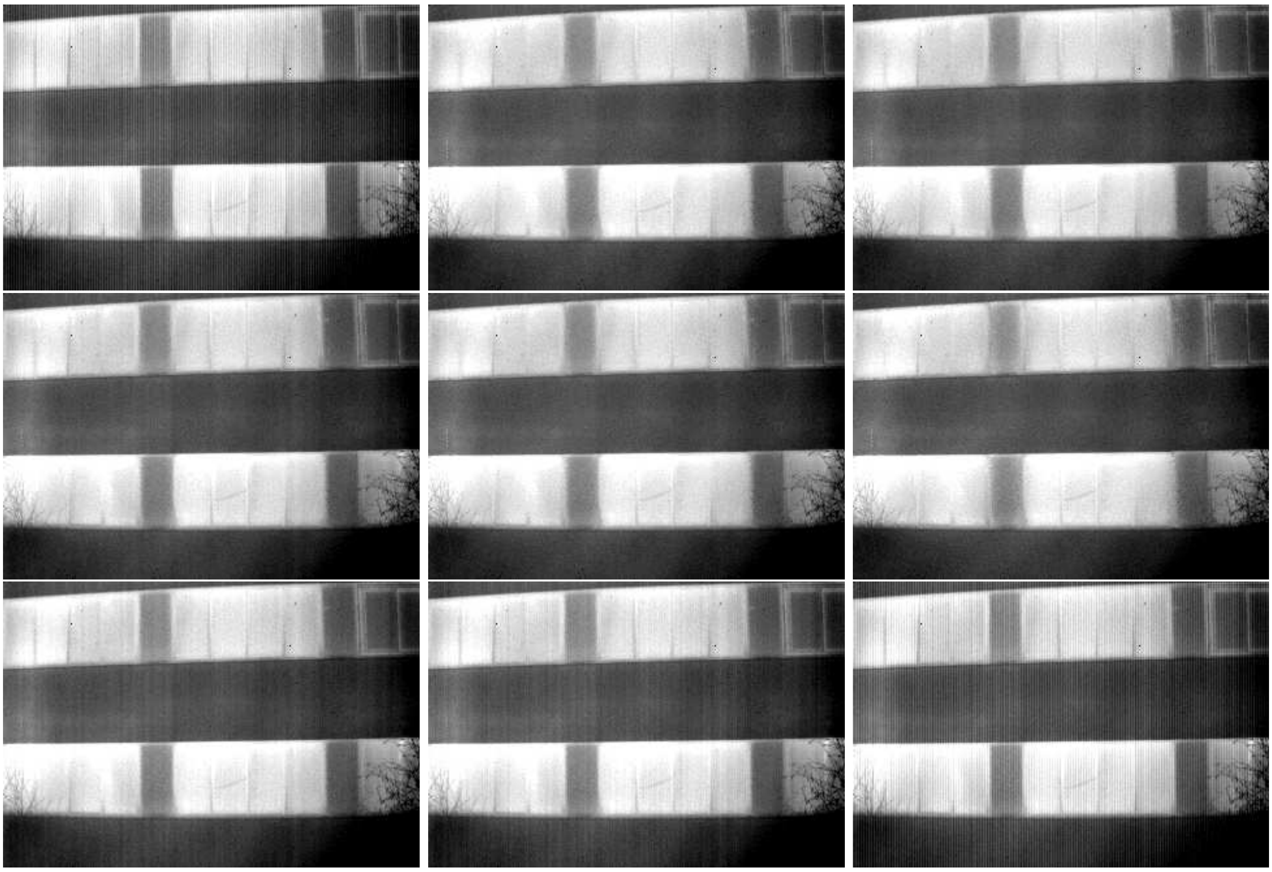

Figure 10. First :RAW (uncooled camera), then from left to right: increasing the σ parameter. TV norms : 8.07, 7.44, 7.38, 7.39, 7.36, 7.59, 7.62, 7.85, 8.07.

ACKNOWLEDGMENTS

We thank the Délégation Générale pour l'Armement (DGA) for supporting this work.

REFERENCES

[1] Biberman, L. M., ed., [*Electro-Optical Imaging : system performance and modeling*], SPIE Press (october 2000).

[2] Harris, J. and Chiang, Y., "Nonuniformity correction of infrared image sequences using the constant-statistics constraint," *IEEE TIP* **8**, 1148–1151 (August 1999).

[3] Pezoa, J. E., Torres, S. N., Córdova, J. P., and Reeves, R. A., "An enhancement to the constant range method for nonuniformity correction of infrared image sequences," *Lecture Notes in Computer Science* **3287**, 525–532, CIARP, Springer (2004).

[4] Torres, S. N., Vera, E. M., Reeves, R. A., and Sobarzo, S. K., "Scene-based non-uniformity correction method using constant range: Performance and analysis," 130–139, Proceedings of the 6th SCI, IX:224?229 (2002).

[5] Harris, J. and Chiang, Y.-M., "Minimizing the "ghosting" artifact in scene-based nonuniformity correction," *Behavioural and Brain Sciences* **16**, 48–9 (1993).

[6] Torres, S. N. and Hayat, M. M., "Kalman filtering for adaptive nonuniformity correction in infrared focal-plane arrays," *J. Opt. Soc. Am. A* **20**(3), 470–480 (2003).

[7] Scribner, A., Sarkady, K. A., Kruer, M. R., Caulfield, J. T., Hunt, J., Colbert, M., and Descour, M., "Adaptive nonuniformity correction for ir focal plane arrays using neural networks," **1541**, 100–109, SPIE (1991).

[8] [*Adaptive Scene-Based Non-Uniformity Correction Method for Infrared-Focal Plane Arrays*], Proceedings of the SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls. Orlando, FL, USA (2003).

[9] Tzimopoulu, S. and Lettington, A. H., "Scene based techniques for nonuniformity correction of infrared focal plane arrays," **3436**, SPIE (1998).

[10] Hardie, R. C., Hayat, M. M., Armstrong, E., and Yasuda, B., "Scene-based nonuniformity correction with video sequences and registration," *Appl. Opt.* **39**(8), 1241–1250 (2000).

[11] Zhao, W. and Zhang, C., "Efficient scene-based nonuniformity correction and enhancement," 2873–2876, ICIP (2006).

[12] Delon, J., "Midway image equalization," *Journal of Mathematical Imaging and Vision* **21**(2), 119–134 (2004).

[13] Delon, J., "Movie and video scale-time equalization application to flicker reduction," *IP* **15**, 241–248 (January 2006).

[14] Moisan, L., "Resthline." MegaWave2 Modulus (2007).