A Noise Reduction Filter for Full-Frame Data Imaging Devices

Angelo Bosco[1], Keith Findlater[2], Sebastiano Battiato[1] and Alfio Castorina[1]

Abstract — *This paper describes a method for video sequences denoising that exploits extra-information provided by the image sensor. Fixed Pattern Noise and Temporal Noise are removed by analyzing a series of lines placed at the top of the imager.*

***Index Terms* — Noise Reduction, Fixed Pattern Noise, Temporal Random Noise.**

I. INTRODUCTION

IN order to improve the quality of images acquired by ccd/cmos digital still cameras [1], the problem of filtering noise must be addressed. Depending on the specific field of application, a spatial or spatio-temporal filter is chosen. Specifically, the noise reduction process can be implemented directly in the Bayer Pattern *CFA* domain (*Colour Filter Array)* [2], or, after color interpolation, in the *RGB* or *YCbCr* color spaces. Fig. 1 illustrates a typical image generation pipeline (*IGP*). By placing noise reduction at the beginning of the pipeline the overall *IGP* efficiency is increased.

We have already proposed methods to filter still pictures [3][4] and video sequences [5][6] both working in the CFA domain. In order to regulate the filter strength adaptively, a noise estimation routine is needed. In the case of the temporal filter [5][6], the identification of flat areas in every frame is necessary in order to determine the statistical properties of the superimposed noise [9]; the estimation of the noise level is calculated over the current frame and it is used to filter the successive frame by means of the Duncan Filter [4][5][6].

In this paper we adopt a different approach that does not use image data for noise estimation. We rely, instead, on supplementary data lines provided by the image sensor.

These extra-lines allow the estimation of two different kinds of noise:

- *Fixed Patten Noise (FPN)*
- *Temporal Random Noise*

Experiments show that skipping image analysis for noise estimation allows a significant improvement in terms of processing time.

(1) A. Bosco, S. Battiato and A. Castorina work for STMicroelectronics in the Imaging Group of the Advanced System Technology Lab in Catania (Italy) (e-mail: angelo.bosco@st.com; sebastiano.battiato@st.com, alfio.castorina@st.com).

(2) K. Findlater works for STMicroelectronics, Imaging Division in Edinburgh (UK) (e-mail: keith.findlater@st.com)

The paper is organized as follows. Next section describes a typical image generation pipeline along with a brief introduction to temporal and fixed pattern noise. Section III gives an overview of the proposed spatio-temporal filter while the following section explains how the extra-sensor data can be effectively used. After illustrating the experimental results, a conclusions section with hints for future work closes the paper.

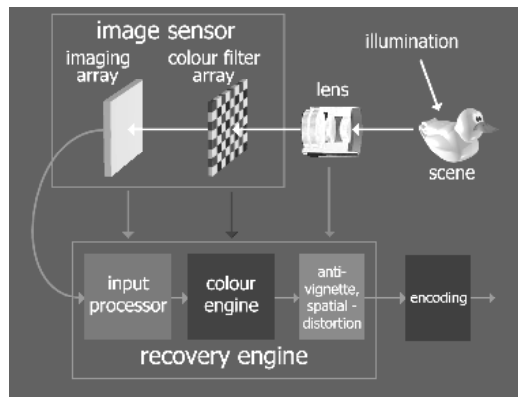

Fig. 1. Typical Image Generation Pipeline.

II. IMAGE GENERATION PIPELINE AND NOISE

A typical image generation pipeline is illustrated in Fig. 1. Noise introduced by the imager reduces the quality of each frame. Removing noise before it reaches the recovery engine, where color interpolation and other relevant image processing algorithms are actually executed [1][10]-[16], is a processing method that we have already exploited [3]-[6]. In a video sequence the frames are usually temporally correlated; a spatio-temporal filter should exploit the frame correlation in order to achieve optimal results. Each frame has to be processed by taking into account temporal information to avoid artifacts generated by the residual noise. Otherwise, the resulting filtered video would be affected by annoying artifacts, such as flickering. Hence, even when no motion occurs, two adjacent frames are never equal on a pixel-by-pixel basis; the differences are introduced by noise, which is spread over each frame. Discriminating between the true image signal and the superimposed noise is a hard task. Adaptive methods are needed in order to perform noise filtering accurately.

A. Temporal Noise

Digital cameras must be able to provide useful pictures both in good and low light conditions. Especially in dim

Manuscript received June 18, 2003

scenes, the camera gain settings are increased in order to boost the signal to acceptable values. This, not only increases the signal level, but also augments noise; we assume this noise to be generally gaussian distributed. Specifically, zero mean additive white gaussian noise (AWGN) of the form (1), is considered [9]:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (1)$$

- 68% of pixels belong to the range $[\mu-\sigma, \mu+\sigma]$
- 95% of pixels belong to the range $[\mu-2\sigma, \mu+2\sigma]$
- 99.7% of pixels belong to the range $[\mu-3\sigma, \mu+3\sigma]$

A certain number of pixels is located in the tails of the Gaussian distribution; hence, they are similar to impulsive noise. The proposed temporal filter is capable to deal also with impulsive noise.

B. Fixed Pattern Noise

In addition to pure temporal noise, one of the most annoying artifacts visible in low light conditions is represented by the *FPN* (*Fixed Pattern Noise).* Sensors built using CMOS technology suffer this noise problem significantly. Especially at high gains, the comparator offsets of the column parallel ADC generate a column-wise error in the image.

(a) (b)

Fig. 2 (a). Visible *FPN* columns in a frame captured in almost complete darkness. (b) After FPN removal.

Hence, *FPN* is particularly visible in low light conditions. Fig. 2 shows a cropped part of an image acquired in almost complete darkness; vertical stripes are visible and must be removed to achieve acceptable quality.

III. SPATIO-TEMPORAL FILTER

The proposed filter uses two working windows; these masks are centered in the current *CFA* noisy frame and in the previous *CFA* filtered frame.

The method is based on Duncan Filtering (*DF*) as shown in [5][6][8].

In order to take advantage of the temporal redundancy, motion between two successive frames has to be considered [10]. This can be done either with a motion estimation algorithm, or by using motion detection. The first method is more reliable but time consuming, as the motion estimation/compensation complexity is higher than motion detection alone.

Motion compensation usually works in a block-wise manner. Images are subdivided in fixed sized blocks (e.g. 16x16 or 8x8); each block in the current frame is coupled with its counterpart in the previous frame (Fig. 3).

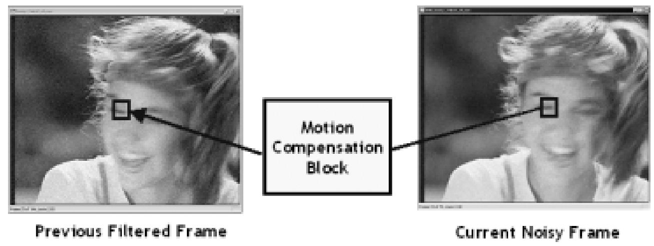

Fig. 3. Motion compensated approach.

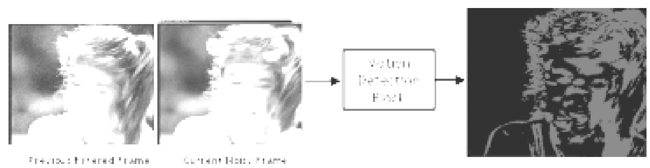

Fig. 4. Motion Detection between two successive frames.

In a real time implementation, usually a simpler motion detection approach is considered. The *SAD* (Sum of the Absolute Differences) between the two working windows is computed; motion is detected if the *SAD* value is higher than a given threshold. In Fig. 4 the results of the motion detection block are shown. Black pixels represent static areas between frames; gray colored regions are relative to pixels where motion between frames has been detected.

The presence of motion hinders the previous frame from being reliable. In case of motion, the data from the previous frame is discarded and the filter support is fully spatial. On the other hand, if no motion is detected, the data from the two frames can be used. In order to regulate the filter strength, noise level estimation has to be performed.

The proposed spatio-temporal filter is based on the knowledge of the noise standard deviation σ (see [5][6][8] for further details). To estimate σ, the homogeneous areas of each frame are used; in these areas, the differences between pixel values are caused mainly by random noise. A texture analyzer, inspecting the local characteristics of a frame, discriminates between flat and textured zones. A threshold T_d is computed, representing the texture degree related to the area where the current pixel is located.

(a) (b)

Fig. 5 (a). A CFA video frame. (b) The CFA frame as seen from the texture analyzer: Dark regions are the homogeneous detected areas; bright areas contain different degrees of texture.

Fig. 6. Some of the flat areas used for the local noise estimation.

If a flat area is detected, its local variance is computed [5][6][7][17]. After scanning the whole frame we end up with a series of local standard deviations; by averaging them, the noise standard deviation of the current frame is available. Actually, the algorithm "sees" an image as different degrees of texture, as illustrated in Fig. 5 and Fig. 6. The estimated noise level is used to regulate the filter strength for the next video frame.

Let σ be the frame variance computed over the previous frame; this value is used to filter the current frame. Denoting with *CurrPix* the noisy pixel under processing, two other values are considered: *CurrPix*+σ and *CurrPix*-σ. Three intervals having wideness W=*f*(σ) are chosen, and centered on *CurrPix, CurrPix*+σ and *CurrPix*-σ (Fig. 7).

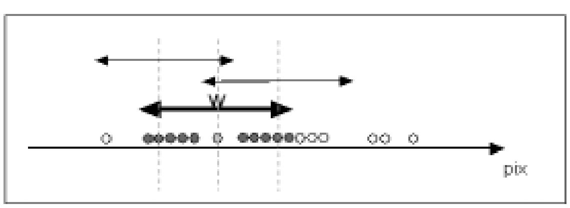

Fig. 7. Duncan Filtering method.

The higher σ, the wider *W*. The interval *W* maximizing the number of pixels is chosen and a weighted average is computed. In the selected interval, pixels far from the central one have lower weights.

This approach, although correct and reliable, is time consuming, as some computations are necessary in order to extrapolate the flat areas and compute the local variances.

IV. FILTERING BY USING SENSOR EXTRA-DATA

A better strategy for noise estimation consists in analyzing the supplementary data provided by the image sensor. As Fig. 8 and Fig. 9 depict, a series of extra-lines is placed at the top of the image sensor. First, there is a series of black lines, followed by a series of dark lines. Black lines have zero integration time; dark lines have the same exposure time as the image lines but they are shielded from the incident light. The extra lines will not be shown in the final color denoised pictures.

These considerations imply that:

- ***black lines*** contain very little noise (specifically, *FPN* noise only);
- ***dark lines*** accumulate almost the same temporal noise as the image, because they have the same integration time of the image lines.

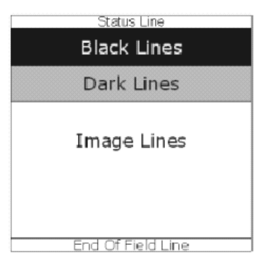

Fig. 8. Black lines are used for *FPN* estimation, Dark lines for random noise estimation.

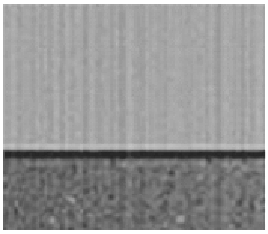

Fig. 9. *FPN* clearly visible in Black lines.

The *FPN* cancellation is achieved by continuously averaging the black sampled data, according to the following equation:

$$FPN\_Est = FPN\_Est - (FPN\_Est/LeakC) + (FPN\_CurSample/LeakC) \quad (2)$$

where:

- *LeakC*: is a constant to weight the previous estimation.
- *FPN_Est*: is the estimation of the *FPN* signature.
- *FPN_CurSample*: is the *FPN* signature, extracted from the current frame.

The current estimation, *FPN_CurSample,* for the *FPN* is obtained by averaging each column *j* of the black lines (3):

$$M_j = \frac{\sum_{i=0,1,\ldots,bk\_rows} pix_{i,j}}{bk\_rows}, \quad j = 0, \ldots, im\_width \quad (3)$$

Hence, denoting with *bk_rows* the number of black rows, a value *Mj* is obtained by averaging each column *j* of the black data.

FPN_Est is initialized to zero and is updated by means of equation (2), each time a new frame arrives.

The first estimation, computed on the first frame, is merely a coarse approximation of the real *FPN* signature. After some iterations the estimation converges towards the correct signature that must be row-wise subtracted from the image data in order to get rid of the *FPN*. The *LeakC* value defines how much weight is attributed to the previous estimations; by changing this value, the speed of convergence can be modulated.

Also, the number of black lines used to "learn" the signature is a key element of the algorithm. If a low number of black lines is used, the estimation would be not reliable, as noise would generate uncertain approximations. On the other hand, using more lines than necessary is a useless waste of resources, both on the sensor and from a computational point of view. Thus, a trade-off between the number of black lines and the leak factor value must be found. Attention must be paid in order to avoid changes in the original illumination, thus *Hmean* is added at each pixel. After convergence, the mean value of the FPN horizontal signature *Hmean* is computed (4):

$$Hmean = \frac{\sum_{i=0,1,\ldots,im\_width} pix_{i,j}}{image\_width} \quad (4)$$

Fig.10 illustrates the simulation results performed to determine the best trade-off between the number of black lines and the value to choose for *LeakC*. On the x-axis, the number of frames is represented; on the *y*-axis, the error from the correct *FPN* signature is shown.

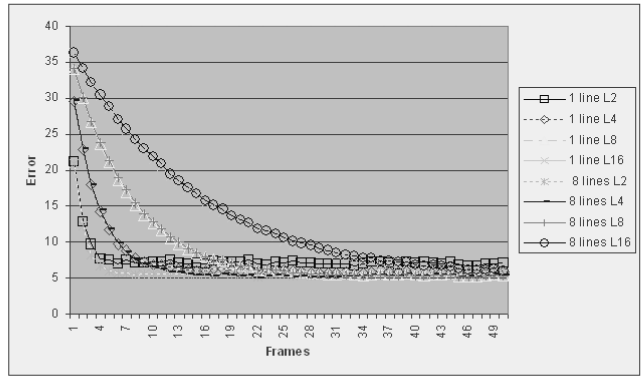

Fig. 10. Simulation results obtained by varying the number of black lines and *LeakC.*

The *FPN* signature is continuously averaged and updated by using the information provided by the black lines. After a certain number of iterations (i.e. after processing the black data provided by a certain number of frames), the estimations change very little from frame to frame (see Fig. 10). At this stage, the signature is considered as "*stabilized"* and it can be subtracted from the image data.

Additionally, before subtraction from image lines, the *FPN* signature must be scaled by the same gain used during capture.

The extra-lines information can be also used to determine the random noise level. An estimation of the temporal noise standard deviation can be obtained by processing the dark data. Pixel fluctuations in dark lines are caused mainly by random noise. Dark lines noise has approximately the same power of image noise, as these lines are held in exposure for the same time. Hence, a straightforward computation of noise standard deviation on dark data is equivalent to a noise level estimation on homogeneous frame areas. The overall filtering process, that removes both *FPN* and random noise, is illustrated in Fig. 11. Two frames are considered: the current ***C**urrent **N**oisy **F**rame* (*CNF*) and the ***P**revious **F**iltered **F**rame* (*PFF*). Black data from *CNF* is used to estimate the *FPN* signature; dark data from *CNF* allows the estimation of temporal noise. The *Update Signature Block* updates the *FPN* estimation by weighting it with the previous one. The *Remove FPN Block*, removes *FPN* from *CNF*. Finally, the temporal filter processes the *PFF* and the *FPN*-free *CNF* to remove the temporal noise, by using the noise level estimation previously computed on the dark data.

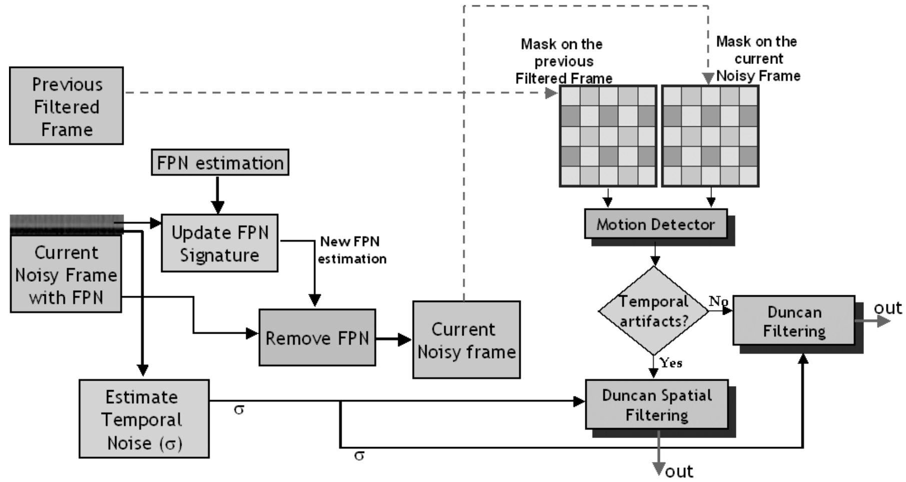

Fig. 11. Overall filter processing pipeline.

V. EXPERIMENTAL RESULTS

The proposed technique estimates noise levels starting from the sensor extra-lines; this method allows to filter image data according to its own noise levels, instead of relying on information computed over the previous frames.

A test sequence composed by 150 frames has been contaminated with gaussian noise having high variance. *FPN* was also simulated. Then, Fixed pattern noise cancellation and temporal noise reduction is performed.

After removing the temporal noise, there is a significant gain in terms of PSNR.

Fig. 12 shows the PSNR computed between the clean sequence and the noisy one, versus the PSNR between the clean and the filtered sequence.

Fig. 13 illustrates the gain in terms of PSNR.

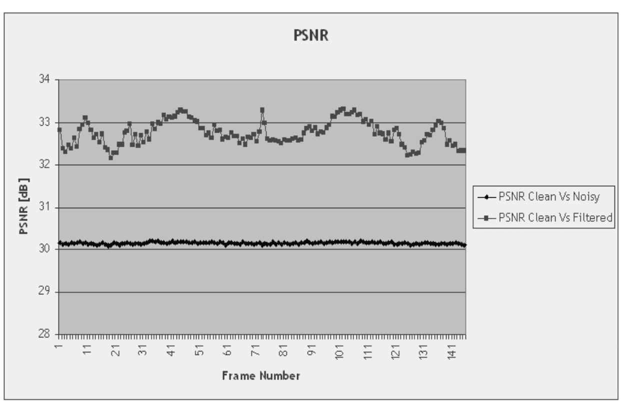

Fig. 12. PSNR (Clean vs. Nosy) and PSNR (Clean Vs. Filtered).

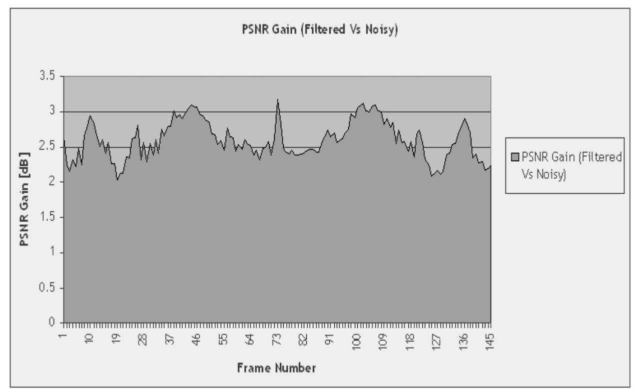

Fig. 13. PSNR Gain relative to the filtered sequence.

The temporal filter can increase the PSNR up to 3-4 dB in our experiments; by removing also *FPN,* a further gain of 1-2 dB can be achieved. The lowest PSNR gains are obtained when a scene change occurs; on the other hand PSNR gains increase with correlation between adjacent frames. Fig. 14 and Fig. 15 show examples of the proposed filtering method.

VI. CONCLUSIONS

A technique to remove noise from video sequences by using sensor extra-lines has been presented. It allows to perform noise levels estimation quickly and to improve significantly the quality of the source noisy video. Two cascaded filters remove *FPN* and temporal noise. Future work will address the problem of implementing a refined spatial filter when temporal information is discarded. A motion compensated approach is also to be investigated.

(a)

(b)

Fig. 14 (a). A video frame, before FPN cancellation; (b) After FPN Cancellation.

(a)

(b)

Fig. 15. (a) A video frame before FPN and temporal noise reduction; (b) After FPN cancellation and temporal noise reduction.

REFERENCES

[1] S. Battiato, M. Mancuso, "*An Introduction to the Digital Still Camera Technology"*, ST Journal of System Research - Special Issue on *Image Processing for Digital Still Camera*, Vol. 2, No.2, December 2001.

[2] B.E. Bayer, "*Color Imaging Array*", U.S. Patent 3,971,065-1976

[3] A. Bosco, M. Mancuso, "*Adaptive Filtering For Image Denoising*", in IEEE Proceedings of ICCE2001, pp.208-209, June 2001.

[4] A. Bosco - *"Adaptive Image Denoising On Bayer Pattern"* - ST Journal of System Research. Vol.2, No.2, Dec.2001.

[5] A. Bosco, S. Battiato, M. Mancuso, G. Spampinato - " *Temporal Noise Reduction of Bayer Matrixed Data* " - In Proceedings of IEEE ICME'02 International Conference on Multimedia and Expo 2002 - Lausanne, Switzerland, August 2002

[6] A.Bosco, S. Battiato, M. Mancuso, G. Spampinato - " *Adaptive Temporal Filtering for CFA Video Sequences* " - In Proceedings of IEEE ACIVS 2002 Advanced Concepts for Intelligent Vision Systems (pp. 19-24) - Ghent University, Belgium, September 2002.

[7] Aishy Amer, Eric Dubois, Amar Mitiche, "*Reliable and Fast Structure-oriented Video Noise Estimation*", IEEE ICIP 2002.

[8] A. M. Borneo, L. Salinari, D. Sirtori, "*An Innovative Adaptive Noise Reduction Filter for Moving Pictures Based on Modified Duncan Range Test*", ST Journal of System Research, Vol.1, No.1, 2001.

[9] G.Cortellazzo, G.A.Mian, R.Parolari, "*Statistical Characteristics of Granular Camera Noise*", IEEE Transactions On Circuits and Systems For Video Technology, Vol.4, No.6, December 1994.

[10] J.Brailean, R.Kleihorst, S.Efstratiadis, A. Katsaggelos, R. Lagendijk, "*Noise Reduction Filters for Dynamic Image Sequence: A Review*", Proceedings of the IEEE, Vol.83, No.9, Sept. 1995.

[11] S. Battiato, A. Castorina, M. Mancuso, "*High Dynamic Range Imaging for Digital Still Camera"*, In Proceedings of SPIE *Electronic Imaging 2002* - Sensors, Cameras, and Applications for Digital Photography IV - San Jose' CA USA, January 2002.

[12] S. Battiato, M. Mancuso, A. Bosco, M. Guarnera - "*Psychovisual and Statistical Optimization of Quantization Tables for DCT Compression Engines*" - In IEEE Proceedings of International Conference on Image Analysis and Processing ICIAP 2001 - Palermo, Italy, pp. 602-606 - September 2001.

[13] R. Kimmel, "*Demosaicing image Reconstruction from color CCD samples", IEEE Transaction on Image Processing,* Vol. 7, No. 3, 1999.

[14] H. J. Trussel, "*Mathematics for Demosaicing*", *IEEE Transanction on Image Processing,* 2001.

[15] R. Ramanath, "*Interpolation methods for the Bayer color Array*", *MS Thesis, North Carolina State University, 2000.*

[16] A. Bruna, M. Mancuso, "*JPEG Compression Factor Control: A New Algorithm"*, in Proceedings of IEEE ICCE2001, pp.206-207, Los Angeles, June 2001.

[17] S.I.Olsem, "*Noise Variance Estimation in Images*" in Proceedings of the 8th Scandinavian Conference on Image Analysis, Troms∅, Norway 1993.

AUTHORS

Angelo Bosco was born in Catania, Italy, in 1972. He received the degree in Computer Science in 1997 at the university of Catania with a thesis in the field of image processing about tracking vehicles in video sequences. He joined STMicroelectronics in June 1999 as a system engineer in the Digital Still Camera and Mobile Multimedia Group. Since then, he has been working on distortion artifacts of cmos imagers and noise reduction, both for still pictures and video. Some of his works have been patented; he is also author and co-author of various papers in the image-processing field.

Keith Findlater was born in Edinburgh, UK, in 1975. He received the M.Eng. and Ph.D. degrees from the University of Edinburgh in 1998 and 2001 respectively, both in Electronics. In 1995 he spent 9 months in the Analog Process Technology Development group of National Semiconductor UK working on TCAD modeling and microelectronic test structures. On completion of his Ph.D., which was industrially sponsored by STMicroelectronics Imaging Division, he joined ST as a Technologist. His current research interests include circuits, architectures, and algorithms for improved CMOS image sensor performance.

Sebastiano Battiato received his Italian degree in Computer Science in 1995 and PhD in Applied Mathematics and Computer Science in 1999 both at Catania University. He attended several international workshops/conferences in the imaging field where some results of his research has been presented and then published in the relative Proceedings. A few papers have been also published in International Journals. Since November 1999 has been working at STMicroelectronics in AST Catania Lab in the Digital Still Camera Group as Algorithm Dev. Team Leader. His current research interests are in the area of Image Processing, Compression (Jpeg, Jpeg2000), Watermarking, and Analysis/Synthesis of Texture.

Alfio Castorina received his Italian degree in Computer Science in 2000 at the University of Catania doing a thesis about watermarking algorithms for digital images. Since September 2000 he has been working in STMicroelectonics in the AST Digital Still Camera Group as System Engineer. His current activities include high dynamic range imaging, post-processing image enhancement techniques and Jpeg .2000.