This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

CFA Video Denoising and Demosaicking Chain via Spatio-Temporal Patch-Based Filtering

Antoni Buades and Joan Duran

***Abstract*—Demosaicking and denoising are key steps in the camera imaging chain for both images and videos. The reconstruction errors during these stages will have undesirable effects on the final result if not handled properly. Demosaicking provokes the spatial and color correlation of noise, which is afterwards enhanced by the processing pipeline. This structured noise generally degrades the image quality and, for dark scenes with low signal to noise ratio, prevents the correct interpretation of the image. When trying to mitigate such structured noise on already processed data, denoising methods attenuate details and texture. We present a video processing chain, consisting of a novel strategy for the removal of noise at the camera sensor and a novel video demosaicking algorithm. In both cases, a spatio-temporal patch-based filter with motion compensation is introduced. The experimental results, including real examples, illustrate the performance of the proposed chain, avoiding the creation of interpolation artefacts and colored spots.**

***Index Terms*—CFA, noise estimation, video denoising, video demosaicking, non-linear 3D filter**

I. Introduction

The output from a digital photo or video camera is the end result of a processing pipeline that transforms the incoming amount of photons emitted by the observed scene into voltage to be read out as pixel values. The first and most crucial steps are usually related to chromatic interpolation (demosaicking) and noise removal (denoising). The reconstruction errors during these early stages of the processing pipeline will have undesirable effects on the final result if not handled properly.

Digital photographs and video frames are usually represented by three color values at each pixel. However, most common cameras integrate a CCD or CMOS sensor device measuring a single color per pixel. Demosaicking is the interpolation process by which the two missing values at each position are estimated. The selected configuration of the sensor usually follows the Bayer color filter array (CFA) [1]: out of a group of four pixels, two are green (in quincunx), one is red and one is blue. Furthermore, the sensor recording is perturbed by noise at the time of their loads and discharges.

Demosaicking is usually performed by combining close values from the same channel or the other two [2], [3]. As a result, the noise, being almost white at the sensor, gets correlated. The rest of the processing chain, which mainly consists in color and gamma corrections and compression, enhances the noise in dark parts of the image leading to contrasted colored spots. The size of these spots depends on the applied demosaicking strategy and their removal after the processing pipeline is a challenging task since image structure and noise are not easily differentiated. Consequently, details and texture might be attenuated during the denoising when having only access to the already processed data.

In dark and indoor scenes, the limitations of imaging pipelines are more noticeable. If the camera is set to a long exposure time, the photograph gets blurred by the camera motion and aperture. If it is taken with short exposure, the captured image is dark and its enhancement reveals the noise. The use of a high ISO value for short exposures makes the image brighter but also makes noise more annoying. Many times, such a dilemma can be solved by taking a burst of images [4], [5], [6], which means a set of digital images taken with the same camera, in the same state and quasi instantaneously. These *bursts* are obtained by video, or by using the burst mode incorporated in recent reflex and compact cameras, or simply by taking several snapshots while holding firmly the camera in a fixed direction.

In this paper, we propose a video processing chain consisting of a novel strategy for the removal of noise at the sensor and a novel video demosaicking algorithm. The denoising method works directly on the CFA data with unknown noise distribution. Instead of dealing with the CFA structure, each frame is spatially subsampled and converted into a complete 4-channel image accounting for the red, the two green and the blue values. We learn a piecewise linear signal-dependent noise model and then apply a simple variance-stabilization function. Based on [7], denoising is performed by grouping similar spatio-temporal patches and thresholding in an adapted Principal-Component-Analysis (PCA) basis. The authors of [7] learn an adapted representation of color patches, which is computationally expensive for nowadays resolutions. For this reason, we introduce a decorrelation transform and denoise each channel of the decorrelated sequence. In addition, the method in [7] is limited to the denoising of full color images assuming a uniform white noise model.

For demosaicking, we introduce a non-linear filter that takes into account inter-frame motion and exploits spatio-temporal redundancy. The selection of candidate patches depends on a motion-compensated 3D distance, which makes it robust to noise and aliasing. To the best of our knowledge, such a 3D patch selection procedure using optical flow for registration has only been proposed in [7] for denoising and in [8] for super-resolution. While in [7] all selected patches are used in order to learn an adapted model, we compute a

A. Buades and J. Duran are with Institute of Applied Computing and Community Code (IAC3) and with the Dept. of Mathematics and Computer Science, Universitat de les Illes Balears, Cra. de Valldemossa km. 7.5, E-07122 Palma, Spain (email: {toni.buades, joan.duran}@uib.es). The authors were supported by the Ministerio de Economía y Competitividad of the Spanish Government under grant TIN2017-85572-P (MINECO/AEI/FEDER, UE).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

weighted average depending on the Euclidean distance to the reference patch. This makes our approach intrinsically robust to flow inaccuracies and occlusions, while the algorithm in [7] needs to explicitly detect occluded patches and discard them. Furthermore, we interpolate missing values but also modify the CFA ones, which permits the removal of residual noise and avoids the creation of *zipper*, an on-off pattern created by the juxtaposition of contiguous original and interpolated pixels. Finally, our non-linear filter does not work on the full color frames as in [8]. We first apply it to the green sequence and then to the differences green-red and green-blue in order to exploit the inter-channel correlation in natural images.

This work extends our previous conference paper [9], in which we introduced the demosaicking method but no treatment of the noise at the sensor was conducted.

The rest of the paper is organized as follows. In Section II, we review the state of the art in video demosaicking and denoising. We present in Section III the proposed algorithm for denoising a video sequence at the CFA sensor and, in Section IV, the spatio-temporal filtering approach for video demosaicking. The performance of the full chain is evaluated in Section V. Finally, conclusions are drawn in Section VI.

II. State of the Art

In this section, we outline the state of the art in video demosaicking and denoising.

A. Video demosaicking

Despite the extensive literature in single color image demosaicking ([10], [11], [12], [3]), there exist few works on its extension to video. While the demosaicking of a single image might give reasonable results, the consecutive play of several frames might introduce artefacts. Wu *et al.* [13] proposed to match the CFA green sample blocks in adjacent frames via motion analysis and fuse them with intra-frame estimates of the missing green samples. The interpolation of red and blue channels use the updated green and combine intra-frame and inter-frame information. Lukac *et al.* [14] used a set of stencils comprising three consecutive frames. Gevrekci *et al.* [15] extended the projection onto convex sets algorithm [16] to image sequences by adding a new constraint set based on the spatio-intensity neighborhood. Vandewalle *et al.* [17] proposed to align the set of images looking for a rotation plus translation in the Fourier domain. After registration, a demosaicked image is reconstructed using the full set of images.

B. White noise removal in videos

Local average methods as the bilateral filter [18], or patch-based filtering approaches as nonlocal means (NL-means) [19], block-matching 3D (BM3D) [20] and nonlocal Bayes (NL-Bayes) [21] can be easily adapted to video just by extending the neighboring area to the adjacent frames. In this regard, Boulanger *et al.* [22] extended NL-means to image sequences by growing adaptively the spatio-temporal neighborhood, while Dabov *et al.* [23] adapted BM3D to video (VBM3D) by grouping patches of consecutive frames and performing transform thresholding. Arias *et al.* [24] introduced a patch-based empirical Bayesian video denoising algorithm based on NL-Bayes. Methods using sparse decompositions have also been extended to video [25] as well as approaches based on low rank approximation [26]. Other methods combine a single image estimate with a purely temporal one. Dai *et al.* [27] applied a temporal Linear Minimum Mean Square Error (LMMSE) estimate on motion trajectories. Yue *et al.* [28] used a BM3D estimate of each frame and BM3D applied to similar patches of neighboring frames. In [29], the authors added intercolor prediction to previous approaches.

Previous approaches exploiting spatio-temporal redundancy are only valid for relatively small displacement in the full video sequence. The performance of many denoising methods can be improved by introducing motion compensation. These compensated filters estimate explicitly the motion of the sequence and compensate the neighborhoods yielding stationary data [30]. Maggioni *et al.* [31] introduced the VBM4D algorithm that extends BM3D to video by exploiting the mutual similarity between 3D spatio-temporal volumes constructed by tracking blocks along trajectories defined by the motion vectors. In [7], the authors proposed to denoise each frame by warping the neighboring ones via an optical flow method, defining a 3D volume with almost identical frames. The problems due to occlusions and dense temporal sampling are handled by performing spatio-temporal patch comparison and denoising in an adapted PCA basis. Several other approaches use a domain transform to obtain decorrelated representations of the signal. In this setting, Yu *et al.* [32] presented a motion-compensated 3D wavelet transform with integrated recursive temporal filtering. Varghese and Wang [33] proposed a spatio-temporal Gaussian scale mixture model in the wavelet domain that simultaneously captures local correlations across both space and time. Such correlations are strengthened with a motion compensation process.

C. Color and spatially correlated noise removal in videos

The previous techniques apply only to additive uniform white noise but not to real photography and video data. The literature on color and spatially correlated noise removal from video is scarce. Bennet and McMillan [34] introduced a spatio-temporal bilateral filtering combined with the enhancement technique proposed in [35]. Filtering parameters at each pixel are fixed depending on the amount of enhancement to be applied. Liu and Freeman [36] integrated motion compensation into the NL-means framework while adaptively estimating the noise of the sequence. Xu *et al.* [37] separated the temporal from the spatial filtering using NL-means and combine them using a motion indicator. Gao *et al.* [38] transformed the video sequence into the YCbCr space and applied a bilateral filter to both luminance and chrominance. A multi-scale wavelet transform permits to deal with non-white noise. Jovanov *et al.* [39] simultaneously performed multiview image sequence denoising, color correction and the improvement of sharpness in slightly defocused regions for sequences of images coming from different cameras and, thus, with different degrees of blur and noise. In [40], the authors proposed a multi-scale

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

algorithm, estimating and removing the noise at each scale. It uses a variance stabilization transform and the white noise removal algorithm previously introduced in [7].

A particular type of algorithms are those dealing with a burst of images. In such cases, a parametric transform, commonly an homography, is able to register the images within the sequence. A robust combination of these images after registration permits removing correlated noise [41]. Liu *et al.* [4] presented a fast denoising method dealing with a burst of noisy images. The homography selection is accelerated by a multi-scale procedure, which includes a local refining of the homography and of the image fusion.

D. Sensor noise removal and joint denoising-demosaicking

If denoising is applied to the sensor data, the algorithm must adapt to the CFA structure. Zhang *et al.* [42] proposed a PCA based spatially-adaptive denoising algorithm that works directly on the CFA image using a supporting window, which contains color components from different channels. Patil *et al.* [43] denoised the CFA data by dictionary learning combined with a variance stabilization transform. While the two previous approaches were designed for single image denoising, Kim *et al.* [44] introduced an algorithm for noise reduction and enhancement of extremely low-light videos.

Some authors handled denoising and demosaicking in a sequential manner. Chatterjee *et al.* [45] adapted NL-means to single mosaicked images by averaging only patches having the same CFA pattern. Any variance stabilization transform is used. For the demosaicking stage, the authors proposed a variational method for which the CFA is taken as the low-resolution counterpart in a super-resolution framework. Zhang *et al.* [46] pioneered a denoising-demosaicking strategy for CFA video data. A spatio-temporal extension of [47] is first applied for denoising by combining only patches having the same CFA pattern. Noise is reduced by thresholding in an adaptive PCA basis. A single-frame demosaicking algorithm is then applied to the denoised CFA data and subsequently the demosaicked frames are post-processed by exploiting the spatio-temporal redundancy to reduce color artefacts.

Recent approaches try to jointly solve denoising and demosaicking, but mainly work with single images. Paliy *et al.* [48] performed interpolation and denoising using inter-color filters selected by polynomial approximation. Tan *et al.* [49] proposed an unified energy functional combining several priors and solved the resulting minimization problem by alternating direction methods of multipliers (ADMM) [50]. Heide *et al.* [5] presented a single optimization technique to deal with all stages of the image processing pipeline. Denoising, demosaicking and deconvolution are written as an energy minimization problem. Hasinoff *et al.* [6] proposed a chain for a burst of CFA images. The method aligns and merges all images using a tiled translation in a Gaussian pyramid process.

With the increasing prominence of convolutional neural networks (CNNs), deep architectures have been built for joint denoising and demosaicking. However, the literature is still scarce and, to the best of our knowledge, all proposed methods work either on a single image or on a burst of images. Regarding single-image joint denoising and demosaicking, Gharbi *et al.* [51] introduced a data-driven filtering approach and trained their model on a large corpus of images instead of using hand-tuned filters to optimally leverage regularities found in natural images. Kokkinos and Lefkimmiatis [52] proposed a CNN arquitecture inspired by classical image regularization methods and large-scale convex optimization techniques. Regarding burst of images, Mildenhall *et al.* [53] used a CNN for predicting spatially varying kernels that can both denoise and align a burst of images taken from a handled camera, but no demosaicking is performed. Very recently, Kokkinos and Lefkimmiatis [54] presented an iterative residual network for denoising and demosaicking a burst of images. In the two latter cases, an affine transformation is assumed between consecutive images, so their applicability to video sequences with general scene motion is very limited.

III. CFA Video Denoising

Let $\{f_n\}_{n=1}^N$, $f_n = \left(f_n^R, f_n^G, f_n^B\right)$, be the mosaicked and noisy video sequence acquired at the sensor. According to the CFA, f_n^R and f_n^B are only known at one fourth of the pixels and f_n^G at one half. Fixed a particular frame f_k of the sequence, let u_k be its underlying denoised and demosaicked counterpart. One commonly assumes that each raw video frame f_n is connected to u_k through the following model:

$$f_n = D_n W_n u_k + \varepsilon_n, \tag{1}$$

where D_n is the degradation matrix that models the spatial response of the physical device and in particular the CFA pattern, W_n is a backward warping operator, and ε_n is the realization of the noise at the sensor. Note that D_n determines the position of the pixels to be interpolated from the known values and will be referred as *CFA mask*. In order to compute u_k from (1), one needs to establish a correspondence between the reference image and each of the frames in addition to denoising and demosaicking.

Since chromatic interpolation correlates the noise and introduces aliasing artefacts like zipper effects, we denoise the CFA video sequence before demosaicking. Instead of dealing directly with the CFA structure, we spatially subsample each frame and convert it into a 4-channel image accounting for the red, the two green and the blue values. Accordingly, we get a sequence of complete 4-channel images, denoted by $\{I_n\}_{n=1}^N$ with $I_n = \left(I_n^R, I_n^{G_1}, I_n^{G_2}, I_n^B\right)$, where each new frame has half the width and the height of the original sensor data.

Most state-of-the-art denoising algorithms deal only with white and uniform noise. However, in real scenes, the level of noise depends on the level of the signal. We use a piecewise linear signal-dependent model for estimating the noise of the raw video sequence. Based on the obtained noise curves, the sensor data is transformed by applying a variance stabilization. Denoising is finally performed on the stabilized sequence.

A. Noise estimation

In order to estimate the noise, we assume that all images of the sequence have been acquired under the same conditions, i.e., the same ISO gain factor and exposure. In such a case, all

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

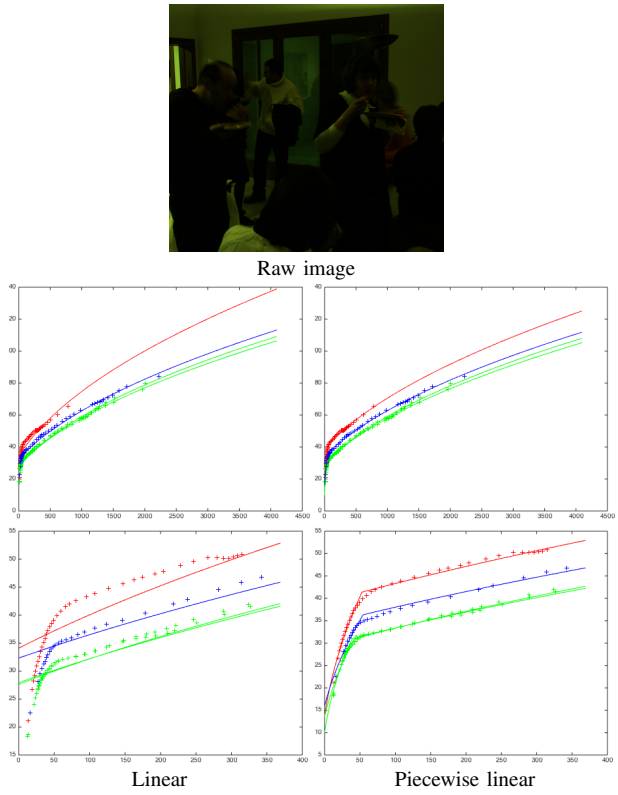


Fig. 1. Noise curves obtained for a 12-bit raw image. The crosses denote the estimated noise pairs while the continuous lines represent their interpolation to the full color range. The graphics at the bottom display zoom in on the low values of the intensity range. A single linear function is not able to correctly approximate the whole range since it does not fit correctly the darker regions of the image, which is critical for the denoising stage. The proposed piecewise linear approach (Algorithm 1) provides a much more accurate noise curve.

video frames share the same noise model. We adapt the single-image noise estimation algorithm in [55], which divides each image in patches and applies the Discrete Cosine Transform (DCT) as proposed by Ponomarenko *et al.* [56] for uniform noise estimation. The low frequencies of the DCT allows selecting the less oscillating patches and the high frequencies of these selected patches yield a standard deviation estimate.

The noise is estimated channel by channel. For each band $C \in \{R, G_1, G_2, B\}$, we classify 2D patches from the sequence $\{I_n^C\}_{n=1}^N$ depending on its mean. The full range is partitioned into non-overlapping bins and each patch is arranged into the corresponding one, so that each interval contains a large enough number of elements. That is, the intensity level range is not divided into uniform length intervals, but these intervals are adapted to the image itself.

The algorithm yields a set of noise observations $\{(\lambda_i, \sigma_i)\}$, where σ_i is the noise standard deviation associated to the intensity value λ_i, which have to be interpolated to the whole range in order to have a complete noise model. Noise at the sensor is often assumed to follow a Poisson distribution with linear variance. However, this simplification is not valid in dark regions where other sources of noise are dominant. Figure 1 illustrates how the obtained set $\{(\lambda_i, \sigma_i)\}$ is not well fitted by a single linear curve. Instead, we propose to look for a piecewise linear noise model with a linear fitting curve for the darker values of the color range and another one for the lighter values. We ask the two linear models to be jointly continuous, and the point in the range for which the approximation changes is also considered as a part of the optimization problem, which is solved by LMMSE. In Figure 1, it is observed how the proposed piecewise linear approach correctly fits all the initial noise pair estimations. The full noise estimation method is summarized in Algorithm 1.

Algorithm 1 Signal-dependent noise estimation of raw videos

Input: noisy 4-band video $\{I_n\}_{n=1}^N$, $I_n = \left(I_n^R, I_n^{G_1}, I_n^{G_2}, I_n^B\right)$.
Output: a noise curve per channel.

1: Extract all 8×8 (overlapping) 2D blocks from all the frames in $\{I_n\}_{n=1}^N$.
2: **for** each color channel $C \in \{R, G_1, G_2, B\}$ **do**
3: Compute average value of each block in $\{I_n^C\}_{n=1}^N$.
4: Classify blocks in bins according to average value. Adapt number of bins such that every bin contains at least 42000 blocks.
5: **for** each bin i **do** ▷ Ponomarenko et al.'s [56]
6: W = set of 8×8 image blocks (1 band) in bin i.
7: $D = DCT(W)$, 2D orthonormal DCT-II.
8: Compute V^L = low-frequency variances of D.
9: Compute V^H = high-frequency variances of high-frequency blocks in D with small value in V^L.
10: λ_i = average intensity value of bin i.
11: σ_i^2 = median of variances in V^H.
12: **end for**
13: Interpolate $\{(\lambda_i, \sigma_i)\}$ based on a piecewise linear model. The point where the linearity changes is considered as part of the optimization problem, solved by LMMSE.
14: Filter noise curve.
15: **end for**

B. Variance stabilization

We usually refer to the Anscombe transform as the transformation $\Psi(u) = 2\sqrt{u + \frac{3}{8}}$, which is known to stabilize the variance of a Poisson noise model. However, any signal-dependent additive noise can be stabilized by a simple transform. Let $v = u + \alpha(u)\varepsilon$ be a noisy signal with α denoting the learnt noise curve, we search for a function Ψ such that $\Psi(v)$ has uniform standard deviation. When the noise is small compared to the signal, we can use the Taylor's decomposition $\Psi(v) = \Psi(u) + \Psi'(u)\alpha(u)\varepsilon$. Forcing the noise term to be constant, $\Psi'(u)\alpha(u) = c$, and integrating we obtain

$$\Psi(u) = \int_0^u \frac{c\, dt}{\alpha(t)}. \tag{2}$$

Note that the parameter c represents the amplitude of the noise after stabilization. When a linear variance noise model is taken, (2) gives back the Anscombe transform. Since noise amplitude is different in each channel, we apply a different stabilization to each of them. This is not the case of the classical Anscombe which uses the same transformation for all channels. The inverse transform shall be applied back after denoising to get the original color range.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Classical Anscombe Proposed stabilization Noise curves after stabilization

Fig. 2. Variance stabilization analysis on a Nikon CFA image, which has been first spatially sampled according to the CFA to obtain a full 4-channel image. We apply the classical Anscombe and the proposed variance stabilization (2). The graphic displays the noise curves estimated on each transformed image using the method in [55]. The dashed lines (proposed variance stabilization) are clearly more constant than the continuous ones (classical Anscombe).

Figure 2 compares the application of the classical Anscombe transform with the proposed variance stabilization on a single Nikon CFA image. In order to apply the proposed approach, we compute a noise curve for each channel through the signal-dependent noise estimation method proposed in [55]. The parameter c in (2) is selected in such a way that the color range of the transformed image equals the range of the stabilized image with Anscombe. The noise in the two stabilized images is finally estimated using [55], with 5 bins, to check its uniformity. We observe that the noise amplitude is constant for the full range after the use of the proposed variance stabilization, which is not the case with Anscombe.

C. Noise removal

For the sake of simplicity, we keep the notation $\{I_n\}_{n=1}^N$, $I_n = \left(I_n^R, I_n^{G_1}, I_n^{G_2}, I_n^B\right)$, for the sequence of 4-channel images after applying the proposed variance stabilization. The goal is to obtain a denoised CFA video. We focus on the description of the method for a particular frame of the sequence which is used as reference and denoted by I_k. The same procedure is applied sequentially to all the other frames.

We will adapt the SPTWO video denoising algorithm proposed in [7] to the CFA sensor data. This algorithm can provide optimal results since at this stage the input video contains uniform noise.

1) SPTWO video denoising algorithm: First, the optical flow between I_k and adjacent frames in a temporal neighborhood is computed and used for warping them onto I_k. Occlusions are detected depending on the divergence of the estimated flow: negative divergence values indicate occlusions. Additionally, the color differences are checked after motion compensation: a large difference indicates occlusion, or at least failure of the brightness constancy assumption [57].

Once the neighboring frames have been warped, the algorithm uses a 3D volumetric approach to search for similar patches, while still 2D image patches are averaged for denoising. For each patch P of the reference frame I, the 3D patch $\mathcal{P}$ referring to its extension to the temporal dimension is considered, having M times more pixels than the original one (assuming M patches in the temporal neighborhood). Since the images have been resampled according to the estimated flow, the data is supposed to be static. The algorithm looks for the K extended patches $\mathcal{Q}$ closest to $\mathcal{P}$ minimizing the Euclidean distance of the intensity values. If a pixel was labelled as occluded in a frame, then this frame is not included neither in $\mathcal{P}$ nor in $\mathcal{Q}$. The selected 3D volumes are then sliced per frame, providing a collection of 2D patches in the spatio-temporal domain. As each extended patch contains M 2D patches, the group contains $K \cdot M$ selected patches. The PCA of these patches is computed and their denoised counterparts are obtained by thresholding of the coefficients. As proposed in [47], the decision of canceling a coefficient is not taken depending on its magnitude, but the magnitude of the associated principal value. The whole patch is restored in order to obtain the final estimate by aggregation. A second iteration of the algorithm is performed using the oracle strategy. Finally, color videos are denoised directly without the use of any color decorrelating transform, thus each color patch has three times more components than in the grayscale case.

2) Adapted SPTWO for CFA video denoising: The application of the color video denoising approach [7] to the 4-channel sequences is computationally expensive for videos of nowadays camera resolutions. Indeed, in order to denoise a set of patches of $m \times m$ pixels, it involves the computation of the PCA for vectors of size $4m^2$. In addition, we need the set of patches to be larger than the dimension of the vectors in order to correctly compute the adapted representation model. In view of these conditions, we apply the SPTWO algorithm to each channel of the sequence after decorrelation.

We introduce a decorrelation transform that maps each 4-channel frame of the sequence $\{I_n\}_{n=1}^N$ into a channel-decorrelated space, which will be called YUVW by analogy with the classical YUV space. In order to estimate such a transform, we took several CFA images, converted them to 4-channel data and considered for each pixel the four dimensional vector containing the red, the two green and the blue values. The PCA analysis of this set yields the transform

$$A = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ -0.5 & 0.5 & 0.5 & -0.5 \\ 0.65 & 0.2784 & -0.2784 & -0.65 \\ -0.2784 & 0.65 & -0.65 & 0.2784 \end{pmatrix}, \quad (3)$$

where each row represents a principal vector. Similar analysis to standard RGB images leads to an orthonormal version of the YUV space. Since A is an orthonormal matrix, its application after the variance stabilization step makes the new components have uniform noise with the same standard deviation. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Algorithm 2 CFA video denoising with uniform noise

Input: noisy 4-channel stabilized video $\{I_n\}_{n=1}^N$, with $I_n = (I_n^R, I_n^{G_1}, I_n^{G_2}, I_n^B)$, and noise standard deviation σ.
Output: denoised 4-channel video sequence.

$AI_n = (I_n^Y, I_n^U, I_n^V, I_n^W)$, using (3).
for each frame $I_k^Y \in \{I_n^Y\}_{n=1}^N$ **do**
  $N_k =$ temporal neighborhood (M adjacent frames)
  Compute optical flow between I_k^Y to all $I_j^Y \in N_k$ using TV-L^1 algorithm [58].
end for
for each channel $C \in \{Y, U, V, W\}$ **do**
  for each frame $I_k^C \in \{I_n^C\}_{n=1}^N$ **do**
    Warp all $I_j^C \in N_k$ using the estimated flow.
    $S_k =$ static sequence around I_k^C.
    for each pixel x **do**
      $\mathcal{P}_x =$ patches centered at x for all frames in S_k.
      Remove from $\mathcal{P}_x$ patches with occluded pixels.
      for each pixel y **do**
        $\mathcal{P}_y =$ patches centered at y for frames in S_k.
        Remove from $\mathcal{P}_y$ patches with occlusions.
      end for
      $\mathcal{S}_K = K$ closest sets $\mathcal{P}_y$ to $\mathcal{P}_x$.
      Get denoised patch $\hat{P}_x$ centered at x:
        PCA analysis of patches in $\mathcal{S}_K$.
        $\hat{P}_x =$ reconstruction from thresholded coefficients in PCA basis (thresholds depend upon σ).
    end for
    Get frame by aggregation of denoised patches.
  end for
end for
Get denoised RG_1G_2B video by applying inverse decorrelation transform $A^\top$.

transposed matrix generates the inverse transform that converts an image from YUVW to the RG_1G_2B space.

Let $\{AI_n\}_{n=1}^N$, $AI_n = (I_n^Y, I_n^U, I_n^V, I_n^W)$, be the sequence after the decorrelation transform in (3) has been applied to each frame I_n of the RG_1G_2B sequence. The optical flow between AI_k and the other frames is computed on the I_n^Y components, which contain the main geometry of the scene. For this purpose, we use the well-known TV-L^1 optical flow variational model [58]. Then, for each $C \in \{Y, U, V, W\}$, we consider the single-channel video sequence $\{I_n^C\}_{n=1}^N$ and denoise it using the SPTWO algorithm. The selection of candidate 3D patches is performed on the I_n^Y for all components. We drop the second oracle step proposed in [7] to reduce the computational cost.

After all components in the YUVW space have been denoised, the inverse decorrelation transform is used to get a RG_1G_2B sequence back. See Algorithm 2 for a general overview of the proposed denoising strategy. The inverse variance stabilization is applied to each channel and the CFA structure is finally recovered. In the end, we are left with a video of denoised CFA frames, which will be denoted by $\{\hat{f}_n\}_{n=1}^N$, $\hat{f}_n = (\hat{f}_n^R, \hat{f}_n^G, \hat{f}_n^B)$, where $\hat{f}_n^R$ and $\hat{f}_n^B$ are only known at one fourth of the pixels and $\hat{f}_n^G$ at one half of them.

IV. Video Demosaicking

We now introduce a motion-compensated non-linear filtering approach to both interpolate the missing color values in the CFA video denoised sequence $\{\hat{f}_n\}_{n=1}^N$ and remove residual noise from all pixels. The CFA masks D_n introduced in (1) keep the trace of values acquired at the sensor and permits to distinguish them from the initially interpolated ones.

First, we describe the proposed strategy for interpolating single-channel sequences and then adapt it to CFA video data.

A. Spatio-temporal single-channel video interpolation

In this subsection, we assume that the frames of the sequence are grayscale images. The incomplete single-channel video data $\{\hat{f}_n\}_{n=1}^N$ is first interpolated using bicubic interpolation, any anisotropic or demoisaicking technique. Therefore, we get a sequence of spatially filled frames, denoted by $\{\tilde{f}_n\}_{n=1}^N$. The goal is to remove the aliasing and increase the quality of the initially interpolated images by weighted averaging. In order to remove aliasing, we intend to average only pixels belonging to the original observations as labelled by the interpolation masks D_n. We focus on increasing the quality of a reference frame $\tilde{f}_k$ from the sequence. The extension to the full video is straightforward by applying the same procedure to each frame.

In order to process the reference frame $\tilde{f}_k$, we need to establish an inter-frame motion correspondence. We compute the optical flow $v_{k,n}$ between $\tilde{f}_k$ and each of the other frames $\tilde{f}_n$, $n \neq k$. Due to occlusions in the sequence, such a strategy is more accurate than computing the motion between consecutive frames and concatenate the resulting flows. We again use the well-known TV-L^1 variational model [58].

The algorithm proceeds patch per patch of the reference frame $\tilde{f}_k(P)$. The selection of candidate patches to be averaged, $\mathcal{N}_P$, actually depends on a 3D distance taking into account inter-frame motion. This makes the selection procedure more robust to noise and aliasing artefacts. For each reference patch $\tilde{f}_k(P)$, we denote as $\mathcal{P}$ its motion-compensated extension to the frame dimension, having N times more pixels than the original one. We can write this as

$$\mathcal{P} = \bigcup_{n=1}^{N} \left(P + v_{k,n}(P)\right), \tag{4}$$

where $v_{k,n}(P)$ denotes the motion shift for patch P and nth frame. In practice, we take this shift to be the estimated flow between $\tilde{f}_k$ and $\tilde{f}_n$ at the pixel in the centre of the patch P.

Instead of selecting similar patches $\tilde{f}_k(Q)$ in the reference frame, we consider for each of these patches its extension $\mathcal{Q}$ to the temporal dimension. The algorithm looks for the L extended patches $\mathcal{Q}$ closest to $\mathcal{P}$ minimizing the distance

$$d(\mathcal{P}, \mathcal{Q}) = \sum_{n=1}^{N} \|\tilde{f}_n(P + v_{k,n}(P)) - \tilde{f}_n(Q + v_{k,n}(Q))\|^2, \tag{5}$$

with $\|\cdot\|$ denoting the Euclidean norm. As each extended patch $\mathcal{Q}$ contains N 2D patches, the selected group $\mathcal{N}_P$ contains the following $L \cdot N$ patches:

$$\mathcal{N}_P = \left\{P_n^l \,|\, P_n^l = P^l + v_{k,n}(P^l), 1 \leq n \leq N, 1 \leq l \leq L\right\}. \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Algorithm 3 Video demosaicking

Input: denoised CFA video $\{\hat{f}_n\}_{n=1}^N$, $\hat{f}_n = (\hat{f}_n^R, \hat{f}_n^G, \hat{f}_n^B)$.
Output: demosaicked video $\{u_n\}_{n=1}^N$, $u_n = (u_n^R, u_n^G, u_n^B)$.

for each frame $\hat{f}_k \in \{\hat{f}_n\}_{n=1}^N$ **do**
  $\tilde{f}_k$ = demosaicking of $\hat{f}_k$ using [12, Section II].
end for
for each frame $\tilde{f}_k^G \in \{\tilde{f}_n^G\}_{n=1}^N$ **do**
  Compute optical flow between $\tilde{f}_k^G$ to all $\tilde{f}_n^G$ using [58].
end for
Compute $\{u_n^G\}_{n=1}^N$:
  Apply spatio-temporal single-channel video interpolation method from Subsection IV-A to $\{\tilde{f}_n^G\}_{n=1}^N$, with D_n = CFA of the green, and patch-based distances (5) and (9) computed on $\{\tilde{f}_n^G\}_{n=1}^N$.
  Get updated frames by patch aggregation.
for each channel $C \in \{R, B\}$ **do**
  Compute $\{\tilde{f}_n^C - u_n^G\}_{n=1}^N$:
    Apply spatio-temporal single-channel video interpolation method from Subsection IV-A to $\{\tilde{f}_n^C - u_n^G\}_{n=1}^N$, with D_n = CFA of C channel, and patch-based distances (5) and (9) computed on $\{u_n^G\}_{n=1}^N$.
    Get updated frames by patch aggregation.
  Get $\{u_n^C\}_{n=1}^N$ by adding back the green values.
end for

The corrected reference patch is then written as

$$u(P) = \frac{1}{C_P} \cdot \sum_{P_n^l \in \mathcal{N}_P} \omega\big(\tilde{f}_k(P), \tilde{f}_n(P_n^l)\big)\, D_n(P_n^l) \cdot \tilde{f}_n(P_n^l), \quad (7)$$

where the operator $\cdot$ denotes the product element by element of each patch, $\omega\big(\tilde{f}_k(P), \tilde{f}_n(P_n^l)\big) > 0$ measures the similarity between the 2D patches P in $\tilde{f}_k$ and P_n^l in $\tilde{f}_n$, which are respectively denoted by $\tilde{f}_k(P)$ and $\tilde{f}_n(P_n^l)$ to make clear the dependence on the frame. The index $n \in \{1, \dots, N\}$ means that P_n^l belongs to the nth frame of the sequence, while $l \in \{1, \dots, L\}$ means that P_n^l belongs to one of the selected 3D patches. In this setting, C_P is the normalization factor

$$C_P = \sum_{P_n^l \in \mathcal{N}_P} \omega\big(\tilde{f}_k(P), \tilde{f}_n(P_n^l)\big)\, D_n(P_n^l) \quad (8)$$

and the division by C_P in (7) is performed element by element. The use of the interpolation masks D_n in (7) and (8) makes the algorithm average only original pixel values.

The similarity of each of these patches $\tilde{f}_n(P_n^l)$, $P_n^l \in \mathcal{N}_P$, with respect to $\tilde{f}_k(P)$ is finally given by

$$\omega\big(\tilde{f}_k(P), \tilde{f}_n(P_n^l)\big) = \exp\left(-\frac{\|\tilde{f}_k(P) - \tilde{f}_n(P_n^l)\|^2}{h^2}\right). \quad (9)$$

The value of h depends on the degree of aliasing and the noise statistics. The preselection procedure using motion compensation makes this value less critical than in other patch-based regularization techniques [59].

Importantly, no occlusion detection is performed on the estimated motion. In addition, as occluded regions might be different from one frame to another, it makes no sense to use the distance $d(\mathcal{P}, \mathcal{Q})$ (5) for computing the final weights. The comparison between patches $\tilde{f}_k(P)$ and $\tilde{f}_n(P_n^l)$ acts as a validation step and avoids averaging very different patches.

Finally, each pixel of the filtered frame is estimated by aggregating the values of all patches containing it.

B. Spatio-temporal video demosaicking

We introduce now a motion-compensated spatio-temporal demosaicking algorithm to interpolate the denoised CFA video sequence $\{\hat{f}_n\}_{n=1}^N$ by adapting the non-linear filter proposed in the previous subsection.

An initial demosaicked video sequence denoted by $\{\tilde{f}_n\}_{n=1}^N$, $\tilde{f}_n = (\tilde{f}_n^R, \tilde{f}_n^G, \tilde{f}_n^B)$, is needed to compute the inter-frame motion and perform the weighted averaging. We use the local directional single-image demosaicking method proposed in [12, Section II], which is applied to each frame of the sequence independently.

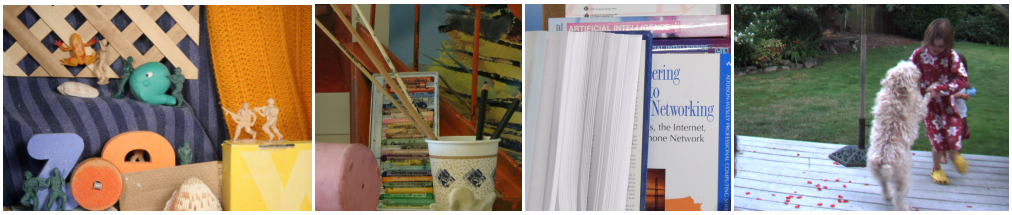
Once the initial demosaicked video has been generated, we proceed to compute the optical flow between each pair of images in the sequence. Due to the higher sampling rate of the green component, which permits an easier reconstruction of the main geometry and texture than the red and blue channels, the flow is computed on the sequence of interpolated green images $\{\tilde{f}_n^G\}_{n=1}^N$ using the TV-L^1 variational model [58].

The initially demosaicked frames $\{\tilde{f}_n\}_{n=1}^N$ may contain zipper effects and residual noise. The proposed filter corrects erroneous structures due to interpolation issues and remove the remaining noise. The sequence of green channels $\{\tilde{f}_n^G\}_{n=1}^N$ is first updated following the described spatio-temporal interpolation method. The mask D_n involved in (7) and (8) is the CFA mask corresponding to the green channel, i.e., a quincunx of factor 2 for each line and column. Thus, only original green values are used in the weighted averaging. Furthermore, the patch-based Euclidean distances (5) and (9) are computed on the initially interpolated green sequence.

Once the green channels of all frames have been updated, which will be denoted by $\{u_n^G\}_{n=1}^N$, we apply the non-linear filter to the differences red-green and blue-green instead of the red and blue themselves. For the sake of simplicity, we describe only the process for the red channel. We consider the sequence of red channels of the initially interpolated frames $\{\tilde{f}_n^R\}_{n=1}^N$ and compute at each pixel the difference with the updated green, i.e., $\tilde{f}_n^R - u_n^G$. We apply now the spatio-temporal interpolation method by considering $\{\tilde{f}_n^R - u_n^G\}_{n=1}^N$ as the input grayscale sequence. The CFA mask of the red channel is used as D_n. Furthermore, the patch-based Euclidean distances (5) and (9) are computed on $\{u_n^G\}_{n=1}^N$ instead of the channel differences. Once the method has been applied and the channel differences have been updated at each pixel by patch aggregation, the green value is added back to get the final red component. This process is performed on each frame, so we get the final red channels $\{u_n^R\}_{n=1}^N$.

The proposed video demosaicking procedure, which is outlined in Algorithm 3, interpolates missing values but also modifies the ones acquired at the sensor. This avoids the creation of zipper effects and permits the removal of residual noise.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology



Army Art Books Dog

Fig. 3. Central frame of the video sequences used in Subsection V-A for the experiments on simulated data.

Method	Video sequence				Avg.
	army	*art*	*books*	*dog*	
$\sigma = 5$					
Single-image demosaicking LDD [12, Section II]	5.39	4.98	5.15	9.31	6.21
Single-image joint denoising-demosaicking ADMM [49]	4.26	3.47	4.48	8.35	5.14
Single-image joint denoising-demosaicking ResNet [52]	4.79	4.02	4.21	9.07	5.52
Video demosaicking [13]	4.30	4.40	4.48	8.59	5.44
Single-image CFA denoising [42] + Video demosaicking [13]	3.84	3.74	3.59	7.98	4.79
Video demosaicking [13] + VBM3D [23]	3.18	3.53	3.46	**7.94**	4.53
Our video demosaicking	3.07	3.23	3.48	8.19	4.49
Our video demosaicking + VBM3D [23]	3.18	3.07	3.19	8.07	4.38
Our CFA video denoising + Single-image demosaicking LDD [12, Section II]	3.80	3.20	3.20	8.40	4.65
Our CFA video denoising + Video demosaicking [13]	3.31	3.36	3.26	7.97	4.48
Our full chain	**3.06**	**2.89**	**2.90**	8.10	**4.24**
$\sigma = 10$					
Single-image demosaicking LDD [12, Section II]	9.50	9.31	9.55	12.22	10.15
Single-image joint denoising-demosaicking ADMM [49]	5.80	5.20	5.70	8.74	6.36
Single-image joint denoising-demosaicking ResNet [52]	8.36	7.91	7.61	11.46	8.84
Video demosaicking [13]	8.10	8.09	8.24	11.20	8.91
Single-image CFA denoising [42] + Video demosaicking [13]	5.38	5.14	5.03	8.63	6.05
Video demosaicking [13] + VBM3D [23]	4.62	5.08	5.10	8.54	5.84
Our video demosaicking	4.42	4.85	5.18	8.71	5.79
Our video demosaicking + VBM3D [23]	4.42	4.60	4.86	8.55	5.61
Our CFA video denoising + Single-image demosaicking LDD [12, Section II]	4.64	4.32	4.21	8.75	5.48
Our CFA video denoising + Video demosaicking [13]	4.27	4.30	4.24	**8.46**	5.32
Our full chain	**4.08**	**3.89**	**3.88**	8.47	**5.08**

TABLE I

RMSE COMPARISON ON THE VIDEOS OF FIGURE 3 FOR WHITE GAUSSIAN NOISE OF S.D. $\sigma \in \{5, 10\}$. THE REPORTED METRICS REFER TO THE FOURTH FRAME OF EACH SEQUENCE. BEST RESULTS ARE IN BOLD AND THE SECOND BEST ONES ARE UNDERLINED.

V. DISCUSSION AND EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our method and compare with several state-of-the-art techniques on both simulated data and real raw videos acquired with a Nikon D80 camera. We also include experiments on bursts of raw images and compare with algorithms adapted to this type of data.

A. Experiments on simulated data

We evaluate the performance of the proposed chain on several sequences from the Middlebury datasets[1] [60], [61]. We use *army, art, books* and *dog*, all of them consisting of $N = 8$ frames. Since the ground truths are available, we evaluate numerically the results in terms of the Root Mean Squared Error (RMSE). The numerical results and figures displayed in this subsection refer to the fourth frame, the associated ground truths of which are displayed in Figure 3.

In order to simulate the noisy CFA sequences from the full color ones, we subsample the reference frames according to the GRBG Bayer pattern and add then white Gaussian noise of standard deviations $\sigma \in \{5, 10\}$. We do not add signal dependent noise since most methods we compare with are designed to cope only with uniform noise. Anyway, we do not modify our algorithm and it still estimates a signal dependent noise model. We use the same CFA mask for all frames. In this scenario, as the raw data is simulated by subsampling color photographies, we do not apply any particular image processing chain, thus colored spots are not enhanced by the white balance or gamma correction stages.

We run our algorithm on 3.5GHz Intel Xeon E5 processor on a MacOs system. The application of the whole chain takes in average 50 seconds per frame for the sequences used in this subsection, being the size of the images 512×512. The inter-frame motion estimation, being an iterative algorithm, is the most computationally expensive part.

The literature on CFA noise removal and demosaicking of videos is scarce. For this reason, we compare with the following combinations of denoising and demosaicking strategies:

- The local directional single-image demosaicking (LDD) method [12, Section II] without any denoising step, which is used as initialization in our demosaicking algorithm.

[1]Middlebury datasets: http://vision.middlebury.edu/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 4. Visual comparison on *army* sequence with noise s.d. $\sigma = 10$. *Den* and *dem* respectively means *denoising* and *demosaicking*. CFA den + VDM avoids color spots but introduces zipper effects, while the result by VDM + VBM3D is less affected by interpolation issues but color low frequency noise is noticeable. Our video demosaicking without denoising is competitive although some residual noise remains. Our video demosaicking + VBM3D oversmoothes the result while our CFA video denoising + LDD introduces zipper effects. The proposed full chain provides the best compromise between noise removal and avoidance of aliasing and interpolation artefacts. (Zoom in for better visual inspection).

- The single-image joint denoising and demosaicking method proposed in [49], which is a variational model that introduces multiple and hidden priors and minimized by means of ADMM. We optimize the trade-off energy parameters for each noise level in terms of the RMSE.
- The single-image joint denoising and demosaicking deep learning based method [52], which uses a cascade of convolutional residual denoising networks (ResNet).
- The video demosaicking method (VDM) introduced in [13] without any denoising step.
- Denoising first and demosaicking later scheme. We first apply the single-image CFA denoising algorithm proposed in [42], which is adapted to the noise level, to each frame and then we use VDM [13].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 5. Visual comparison on *dog* sequence with noise s.d. $\sigma = 10$. *Den* and *dem* respectively means *denoising* and *demosaicking*. Only the different variants of the proposed scheme are able to recover the plant leaves and stems because of being robust to flow inaccuracies. (Zoom in for better visual inspection).

- Demosaicking first and denoising later scheme. We combine VDM [13] with subsequent VBM3D [23]. Since the noise standard deviation is modified after demosaicking, we tested several parameters for VBM3D and kept the one with smallest RMSE. Unlike ours, VBM3D performs a second oracle iteration using a first estimate to drive patch selection and thresholding.

In order to analyze the robustness of each module of the proposed chain, we also include the results obtained by applying the following schemes:

- The proposed video demosaicking algorithm (Section IV) without any denoising step. Since all pixels are modified, it actually removes noise. The filtering parameter h in (9) is fixed depending on the noise standard deviation.
- The proposed video demosaicking algorithm (Section IV) followed by VBM3D [23].
- The proposed CFA video denoising method (Section III) followed by LDD [12, Section II] to each frame.
- The proposed CFA video denoising method (Section III) followed by the video demosaicking method VDM [13].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 6. First frames of the sequences acquired in raw format using a reflex Nikon D80 camera with the same ISO gain factor, aperture and exposure time. The *hall*, *desk* and *poster* sequences consist of 5, 8 and 6 frames, respectively. We display the images after the demosaicking and the corresponding processing pipeline (white balance and gamma correction) are applied, but only the original raw data is used for testing.

- The full proposed chain. In this case, the demosaicking algorithm only filters residual noise which has not been completely removed in the denoising stage.

Table I displays the obtained RMSE values. The error of LDD is significantly larger than the others since it does not involve any denoising stage. For moderate noise, our video demosaicking approach without denoising improves the state of the art but also the proposed CFA video denoising scheme with LDD. This reveals the robustness to noise of the motion-compensated non-linear filtering introduced in Section IV. However, as the noise increases, the CFA video denoising module described in Section III becomes necessary before demosaicking in order to avoid the correlation of the noise during interpolation processes. It is also noticeable that a denoising first and demosaicking later scheme is more necessary when the signal to noise ratio decreases. For both noise standard deviations, the proposed full chain performs the best.

Figures 4 and 5 compare the visual quality of all methods on *army* and *dog* sequences, respectively. LDD and VDM do not remove any noise, while the single-image joint denoising and demosaicking proposed in [49] provides results with distortions at edges and fails in recovering texture and fine structures. The images processed by the state-of-the-art video approaches are slightly blurry and many details have been removed. Unlike ours, these methods are also affected by incorrect flow estimations as it can be observed on the leaves and stems of the plants in Figure 5. Since noise is modified by demosaicking schemes, its removal is quite challenging. For this reason, color correlated noise is more noticeable in the results by VDM + VBM3D, our video demosaicking without denoising and our video demosaicking + VBM3D for which demosaicking is performed before denoising, rather than in those for which the denoising stage applies first. However, the latter introduces zipper effects. The proposed full chain gives the best compromise between noise removal and avoidance of aliasing and demosaicking artefacts.

B. Experiments on real raw data

We test now the performance of the proposed full CFA video denoising and demosaicking chain on real raw data. For this purpose, we use several image sequences acquired in raw format using a reflex Nikon D80 camera with the same ISO gain factor, aperture and exposure time. Since we need to have access to the raw data, we could not get videos by ourselves but only images acquired consecutively. Some of these sequences are actually a burst of images, since they were acquired quasi instantaneously while holding firmly the camera in a fixed direction. Since the camera is hand held, the view point and orientation of the camera slightly changes. For these examples, a standard imaging pipeline comprised of white balance by gray world [62] and gamma correction of factor $\gamma = 0.5$ are used. We do not apply any color correction transform in order to convert the RGB values of the sensor to those of the display. Figure 6 displays the first frames of the three sequences used in the experiments.

We compare with LDD [12, Section II], the single-image joint denoising and demosaicking ADMM approach proposed in [49], and the combination of the single-image CFA denoising algorithm from [42] with the temporal video demosaicking (VDM) [13]. In all these cases, the standard image processing pipeline is applied afterwards. We also compare with the method proposed in [40], which removes noise on already processed data. This approach performs in a multi-scale framework, where the noise is estimated at each scale, and the SPTWO algorithm [7] is used after variance stabilization.

Figure 7 compares the application of these methods to the *hall* sequence. Despite being an indoor sequence, the scene is quite illuminated and the noise is moderate. The colored noise spots of the LDD result are noticeable. The denoised sequence by [40], having only access to the data after the imaging pipeline is applied, is not able to completely remove noise and isolated color spots remain as well as some artefacts. The single-image joint denoising and demosaicking method

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

Fig. 7. Visual comparison on the *hall* sequence acquired in raw format with a Nikon D80 camera. *Den* and *dem* respectively means *denoising* and *demosaicking*. Only the proposed method provides a pleasant visual result since the other techniques are either affected by the noise or introduce interpolation artefacts and blur. (Zoom in for better visual inspection).

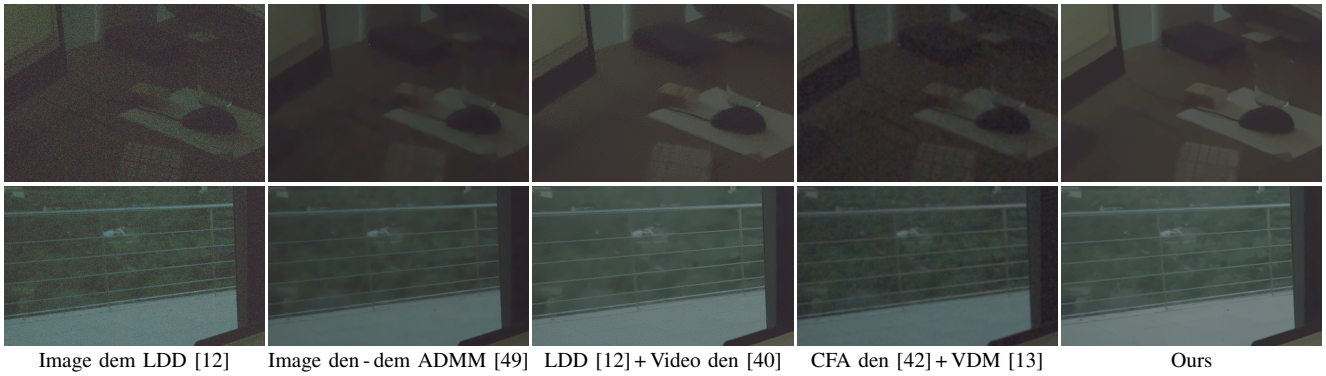


Fig. 8. Visual comparison on the *desk* sequence acquired in raw format with a Nikon D80 camera under low-light conditions. *Den* and *dem* respectively means *denoising* and *demosaicking*. The proposed method gives the best result in terms of noise removal, avoidance of interpolation artefacts, and recovery of the geometry and texture. (Zoom in for better visual inspection).

[49] provides an over-smoothed image with distorted edges, while the combination [42] + [13] gives a blurred result. The proposed chain is able to completely remove the noise and avoid interpolation artefacts while keeping the main details of the underlying scene. Figure 8 displays the results on the *desk* sequence, acquired under low-light conditions. The result by LDD illustrates the poor signal to noise ratio in these conditions, for which image features are hardly visible. The proposed algorithm is able to denoise the data and makes many details appear despite they were initially hidden.

C. Experiments on a burst of raw images

We also test the performance of the method with a burst sequence and compare it with techniques more adapted to this type of data. We took a burst of a poster pasted on a planar surface. It is well known that, in such a case, a parametric transformation is able to correctly register any pair of images. This parametric transformation should include radial distortion parameters if the two view points are quite different. Most methods estimate a global homography, a tiled translation as in [6], or even simpler transformations as a global affinity.

We compare with LDD, the burst algorithm introduced in [41], which is composed by a global registration with an homography using SIFT [63] characteristic points and a weighted combination of the registered images. This method applies to already processed data since one needs to compute the characteristic points. We also test our method replacing the optical flow in the denoising and demosaicking stages by a global homography registration using SIFT points.

Figure 9 displays the results on the *poster* burst sequence. The algorithm in [41] is not able to completely remove the noise since its variance is reduced only as $1/N$, being N the number of frames. However, the denoised result is visually pleasant and has no artefacts. The proposed chain is able to recover all image details and completely removes noise. This result is not improved by using a global registration, being the two solutions nearly identical. This shows the robustness of the proposed scheme to possible flow inaccuracies.

VI. Conclusions

We have proposed a video processing chain, consisting of a novel strategy for the removal of noise at the sensor and a novel video demosaicking algorithm.

The denoising method works directly on the CFA data and estimates a piecewise linear signal-dependent noise model. Denoising is performed by grouping similar spatio-temporal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2956691, IEEE Transactions on Circuits and Systems for Video Technology

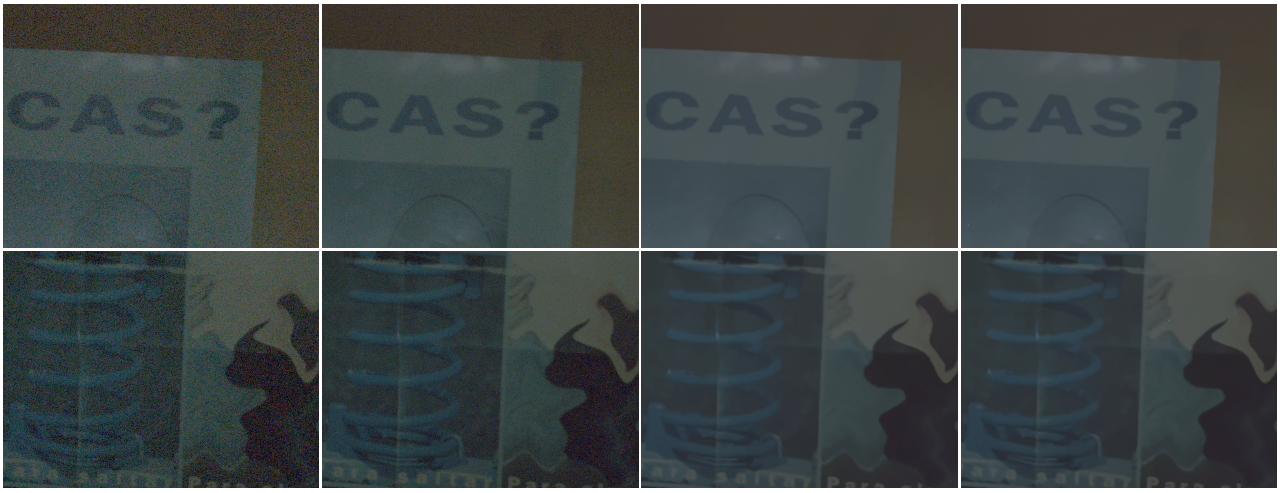


Image demosaicking LDD [12] Burst algorithm [41] Ours Ours with homography

Fig. 9. Visual comparison on the *poster* burst sequence acquired with a Nikon D80 camera under low-light conditions. The burst method in [41] provides a visually pleasant result although not being able to completely remove noise. Our algorithm, which performs almost equally if optical flow or global homography is used, suppresses noise and keeps image features (Zoom in for better visual inspection).

patches and removing noise by thresholding in an adapted PCA basis. A color decorrelation transform is used in order to speed up the process.

For the demosaicking stage, we have presented a non-linear filtering approach making use of the inter-frame motion and spatio-temporal patch similarity. This filter combines patches from several frames not necessarily belonging to the same pixel trajectory. The selection of candidate patches depends on a motion-compensated 3D distance, which makes it robust to noise and aliasing. A weighted average of selected patches depending on 2D comparisons makes the method robust to flow inaccuracies and occlusions. The same strategy could be adapted for other filtering and interpolation tasks such as video deinterlacing or the increase of temporal resolution.

The experiments have illustrated the importance of each module of our video denoising and demosaicking chain in order to achieve state-of-the-art results. The proposed scheme has shown to be the most robust to noise, flow inaccuracies and interpolation artifacts on both simulated and real raw videos.

REFERENCES

[1] B. Bayer, "Color imaging array," 1976, US Patent 3 971 065.

[2] D. Menon and G. Calvagno, "Color image demosaicking: An overview," Signal Process. Image Com., vol. 26, no. 8-9, pp. 518–533, 2011.

[3] J. Li, C. Bai, Z. Lin, and J. Yu, "Optimized color filter arrays for sparse representation-based demosaicking," IEEE Trans. Image Process., vol. 26, no. 5, pp. 2381–2393, 2017.

[4] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun, "Fast burst images denoising," ACM Trans. Graph., vol. 33, no. 6, p. 232, 2014.

[5] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy et al., "Flexisp: A flexible camera image processing framework," ACM Trans. Graph., vol. 33, no. 6, p. 231, 2014.

[6] S. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. Barron, F. Kainz et al., "Burst photography for high dynamic range and low-light imaging on mobile cameras," ACM Trans. Graph., vol. 35, no. 6, p. 192, 2016.

[7] A. Buades, J. Lisani, and M. Miladinović, "Patch based video denoising with optical flow estimation," IEEE Trans. Image Process., vol. 25, no. 6, pp. 2573–2586, 2016.

[8] A. Buades and J. Duran, "Flow-based video super-resolution with spatio-temporal patch similarity," in Proc. British Machine Vision Conf. (BMVC), London, UK, 2017, pp. 656.1–656.12.

[9] ——, "Joint denoising and demosaicking of raw video sequences," in Proc. IEEE Int. Conf. on Image Processing (ICIP), Athens, Greece, 2018, pp. 2172–2176.

[10] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," IEEE Trans. Image Process., vol. 16, no. 1, pp. 132–141, 2007.

[11] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicking," IEEE Trans. Image Process., vol. 18, no. 6, pp. 1192–1202, 2009.

[12] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicking," IEEE Trans. Image Process., vol. 23, no. 9, pp. 4031–4040, 2014.

[13] X. Wu and L. Zhang, "Temporal color video demosaicking via motion estimation and data fusion," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 2, pp. 231–240, 2006.

[14] R. Lukac and K. N. Plataniotis, "Adaptive spatiotemporal video demosaicking using bidirectional multistage spectral filters," IEEE Trans. Consum. Electron., vol. 52, no. 2, pp. 651–654, 2006.

[15] M. Gevrekci, B. Gunturk, and Y. Altunbasak, "POCS-based restoration of Bayer-sampled image sequences," in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, Honolulu, HI, USA, 2007, pp. I–753.

[16] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," IEEE Trans. Image Process., vol. 11, no. 9, pp. 997–1013, 2002.

[17] P. Vandewalle, K. Krichane, D. Alleysson, and S. Süsstrunk, "Joint demosaicing and super-resolution imaging from a set of unregistered aliased images," in Proc. SPIE Electronic Imaging, vol. 6502, San Jose, CA, US, 2007, p. 65020A.

[18] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in Proc. IEEE Int. Conf. on Computer Vision (ICCV), Bombay, India, 1998, pp. 839–846.

[19] A. Buades, B. Coll, and J. Morel, "A non local algorithm for image denoising," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, San Diego, CA, USA, 2005, pp. 60–65.

[20] K. Dabov, A. Foi, V. Katkovnic, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, 2007.

[21] M. Lebrun, A. Buades, and J.-M. Morel, "A nonlocal Bayesian image denoising algorithm," SIAM J. Imaging Sci., vol. 6, no. 3, pp. 1665–1688, 2013.

[22] J. Boulanger, C. Kervrann, and P. Bouthemy, "Space-time adaptation for patch-based image sequence restoration," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 1096–1102, 2007.

[23] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in Proc. IEEE European Signal Processing Conf., Poznan, Poland, 2007, pp. 145–149.

[24] P. Arias and J.-M. Morel, "Video denoising via empirical Bayesian

estimation of space-time patches," J. Math. Imaging Vis., vol. 60, no. 1, pp. 70–93, 2018.
[25] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," SIAM J. Multiscale Model. Simul., vol. 7, no. 1, pp. 214–241, 2008.
[26] H. Ji, S. Huang, Z. Shen, and Y. Xu, "Robust video restoration by joint sparse and low rank matrix approximation," SIAM J. Imaging Sci., vol. 4, no. 4, pp. 1122–1142, 2011.
[27] J. Dai, O. Au, F. Zou, and C. Pang, "Generalized multihypothesis motion compensated filter for grayscale and color video denoising," Signal Process., vol. 93, no. 1, pp. 70–85, 2013.
[28] H. Yue, X. Sun, J. Yang, and F. Wu, "Image denoising by exploring external and internal correlations," IEEE Trans. Image Process., vol. 24, no. 6, pp. 1967–1982, 2015.
[29] J. Dai, O. Au, C. Pang, and F. Zou, "Color video denoising based on combined interframe and intercolor prediction," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 1, pp. 128–141, 2013.
[30] M. Ozkan, M. Sezan, and A. Tekalp, "Adaptive motion-compensated filtering of noisy image sequences," IEEE Trans. Circuits Syst. Video Technol., vol. 3, no. 4, pp. 277–290, 1993.
[31] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4D nonlocal spatiotemporal transforms," in Proc. SPIE Electronic Imaging, vol. 7870, San Francisco, CA, US, 2011, p. 787003.
[32] S. Yu and M. Ahmad, O. Swamy, "Video denoising using motion compensated 3-D wavelet transform with integrated recursive temporal filtering," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 6, pp. 780–791, 2010.
[33] G. Varghese and Z. Wang, "Video denoising based on a spatiotemporal Gaussian scale mixture model," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 7, pp. 1032–1040, 2010.
[34] E. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," ACM Trans. Graph., vol. 24, no. 3, pp. 845–852, 2005.
[35] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," ACM Trans. Graph., vol. 21, no. 3, pp. 257–266, 2002.
[36] C. Liu and W. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in Proc. European Conf. Comput. Vis. (ECCV), ser. LNCS, vol. 6313, Crete, Greece, 2010, pp. 706–719.
[37] Q. Xu, H. Jiang, R. Scopigno, and M. Sbert, "A new approach for very dark video denoising and enhancement," in Proc. IEEE Int. Conf. on Image Processing (ICIP), Hong Kong, 2010, pp. 1185–1188.
[38] Y. Gao, H.-M. Hu, and J. Wu, "Video denoising algorithm via multi-scale joint luma-chroma bilateral filter," in Proc. IEEE Conf. on Visual Communications and Image Processing (VCIP), Singapore, 2015, pp. 1–4.
[39] L. Jovanov, H. Luong, T. Ružic, and W. Philips, "Multiview image sequence enhancement," in Proc. SPIE Electronic Imaging, vol. 9399, San Francisco, CA, US, 2015, p. 93990K.
[40] A. Buades and J. Lisani, "Enhancement of noisy and compressed videos by optical flow and non-local denoising," IEEE Trans. Circuits Syst. Video Technol., 2019.
[41] G. Haro, A. Buades, and J.-M. Morel, "Photographing paintings by image fusion," SIAM J. Imaging Sci., vol. 5, no. 3, pp. 1055–1087, 2012.
[42] L. Zhang, R. Lukac, X. Wu, and D. Zhang, "PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras," IEEE Trans. Image Process., vol. 18, no. 4, pp. 797–812, 2009.
[43] S. Patil and A. Rajwade, "Poisson noise removal for image demosaicing," in Proc. British Machine Vision Conf. (BMVC), York, UK, 2016, pp. 33.1–33.10.
[44] M. Kim, D. Park, D. Han, and H. Ko, "A novel approach for denoising and enhancement of extremely low-light video," IEEE Trans. Consum. Electron., vol. 61, no. 1, pp. 72–80, 2015.
[45] P. Chatterjee, N. Joshi, S. Kang, and Y. Matsushita, "Noise suppression in low-light images through joint denoising and demosaicing," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 2011, pp. 321–328.
[46] L. Zhang, W. Dong, X. Wu, and G. Shi, "Spatial-temporal color video reconstruction from noisy CFA sequence," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 6, pp. 838–847, 2010.
[47] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," Pattern Recogn., vol. 43, no. 4, pp. 1531–1549, 2010.
[48] D. Paliy, A. Foi, R. Bilcu, and V. Katkovnik, "Denoising and interpolation of noisy bayer data with adaptive cross-color filters," in Proc. SPIE Electronic Imaging, vol. 6822, San Jose, CA, US, 2008, p. 68221K.
[49] H. Tan, X. Zeng, S. Lai, and M. Zhang, "Joint demosaicing and denoising of noisy Bayer images with ADMM," in Proc. IEEE Int. Conf. on Image Processing (ICIP), Beijing, China, 2017, pp. 2951–2955.
[50] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning with the alternating direction method of multipliers," Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2011.
[51] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," ACM Trans. Graph., vol. 35, no. 6, p. 191, 2016.
[52] F. Kokkinos and S. Lefkimmiatis, "Deep image demosaicking using a cascade of convolutional residual denoising networks," in Proc. European Conf. on Computer Vision (ECCV), ser. LNCS, vol. 11218, Munich, Germany, 2018, pp. 303–319.
[53] B. Mildenhall, J. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, US, 2018, pp. 2502–2510.
[54] F. Kokkinos and S. Lefkimmiatis, "Iterative residual CNNs for burst photography applications," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 5929–5938.
[55] M. Colom, A. Buades, and J.-M. Morel, "Nonparametric noise estimation method for raw images," J. Opt. Soc. Amer. A, vol. 31, no. 4, pp. 863–871, 2014.
[56] N. Ponomarenko, V. Lukin, M. Zriakhov, A. Kaarna, and J. Astola, "An automatic approach to lossy compression of AVIRIS images," in Proc. IEEE Int. Geoscience and Remote Sensing Symp., Barcelona, Spain, 2007, pp. 472–475.
[57] B. K. P. Horn and B. Schunck, "Determining optical flow," Artificial Intell., vol. 17, no. 1–3, pp. 185–203, 1981.
[58] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in Proc. DAGM Symp., ser. LNCS, vol. 4713, Heidelberg, Germany, 2007, pp. 214–223.
[59] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," IEEE Trans. Image Process., vol. 18, no. 1, pp. 36–51, 2009.
[60] S. Barker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluaton methodology for optical flow," Int. J. Comput. Vis., vol. 92, no. 1, pp. 1–31, 2011.
[61] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, USA, 2007, pp. 1–8.
[62] G. Buchsbaum, "A spatial processor model for object colour perception," J. Franklin Inst., vol. 310, no. 1, pp. 1–26, 1980.
[63] D. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

Antoni Buades received the Ph.D degree in Mathematics from the University of Balearic Islands (UIB), Spain, in 2006. He is currently an associate professor at UIB and member of the Institute of Applied Computing & Community Code (IAC3).

His research is focused on mathematical analysis of digital images and video, particularly, restoration, demosaicking, super-resolution, registration, medical imaging, satellite imaging and deep learning.

Joan Duran received the Ph.D. degree in Mathematics from the University of Balearic Islands (UIB), Spain, in 2016. He is currently an assistant professor at UIB and member of the Institute of Applied Computing & Community Code (IAC3) at the same university.

His research is focused on mathematical analysis of digital images and video, particularly, restoration, registration, super-resolution and satellite imaging.