

# Spatial-Temporal Color Video Reconstruction from Noisy CFA Sequence

Lei Zhang<sup>a,1</sup>, *Member, IEEE*, Weisheng Dong<sup>b,a</sup>  
Xiaolin Wu<sup>c</sup>, *Senior Member, IEEE*, Guangming Shi<sup>b</sup>, *Member, IEEE*

<sup>a</sup>Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong

<sup>b</sup>Key Laboratory of Intelligent Perception and Image Understanding (Chinese Ministry of Education),  
School of Electronic Engineering, Xidian University, China

<sup>c</sup>Dept. of Electrical and Computer Engineering, McMaster University, Canada

**Abstract:** Single-sensor digital video cameras use a color filter array (CFA) to capture video and a color demosaicking (CDM) procedure to reproduce the full color sequence. The reproduced video frames suffer from the inevitable sensor noise introduced in the video acquisition process. This paper presents a spatial-temporal denoising and demosaicking scheme that works without explicit motion estimation. We first perform patch based denoising on the mosaic CFA video. For each CFA patch to be denoised, similar patches are selected within a local spatial-temporal neighborhood. The principal component analysis is performed on the selected patches to remove noise. We then apply an initial single-frame CDM to the denoised CFA data, and subsequently post-process the demosaicked frames by exploiting the spatial-temporal redundancy to reduce the color artifacts. The experimental results on simulated and real noisy CFA sequences demonstrate that the proposed spatial-temporal CFA video denoising and demosaicking scheme can significantly reduce the noise-caused color artifacts and effectively preserve the image edge structures.

**Key Words:** Color filter array, video enhancement, denoising, color demosaicking

---

<sup>1</sup> Corresponding author. Email: [cszhang@comp.polyu.edu.hk](mailto:cszhang@comp.polyu.edu.hk). This research is supported by the General Research Fund of Hong Kong RGC (PolyU 5375/09E), the Hong Kong Innovation and Technology Fund (ITS/081/08), and the National Science Foundation Council of China under Grant no. 60634030.

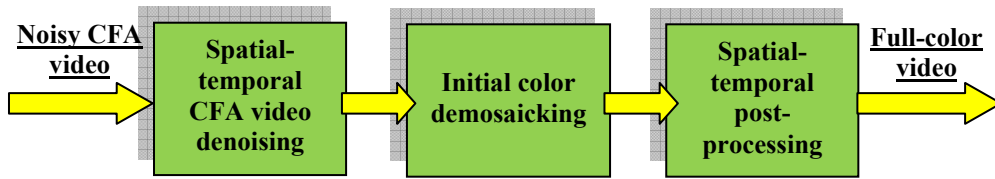
## 1. Introduction

Most of the cost-effective digital video cameras use a single sensor with a color filter array (CFA) [1] to capture visual scenes in color. At each pixel only one of the three primary colors is sampled and the missing color components need to be interpolated. The color interpolation process is usually called color demosaicking (CDM) [2-8]. For video sequences, the temporal correlation in adjacent frames can also be exploited to improve the CDM performance [9-10]. Most of the CDM algorithms assume that the CFA data are noise-free. This assumption, however, is unrealistic and the presence of noise in CFA data will cause serious CDM artifacts that are hard to remove by a subsequent denoising process. Many available denoising algorithms [11-18] are designed for monochromatic/full-color images/videos and they are not directly applicable to the CFA data. Therefore, it is of high demand to develop new algorithms for full-color video reconstruction from noisy CFA sequences.

A commonly used strategy is to demosaick the CFA video first and denoise the demosaicked video later. However, the noise-caused color artifacts generated in the CDM process can be difficult to remove in the following denoising process. Some joint demosaicking and denoising approaches were also proposed. In the joint demosaicking-denoising algorithm [19], both demosaicking and denoising are treated as an estimation problem and solved by the total least square technique. One computationally more efficient joint demosaicking-denoising scheme was developed in [20]. It first performs demosaicking-denoising on the green channel, and then uses the restored green channel to estimate the noise statistics to restore the red and blue channels. Inspired by the directional linear minimum mean square-error estimation based CDM scheme in [4], Paliy *et al.* [21-22] proposed a nonlinear and spatially adaptive filter by using local polynomial approximation for CDM and then adapted this scheme to noisy CFA inputs for joint demosaicking-denoising.

Apart from the above two strategies, another approach to reproducing full-color images from noisy CFA data is to perform denoising before demosaicking. However, due to the mosaic structure of CFA, existing monochromatic image/video denoising methods cannot be applied to the CFA data directly. The CFA image can be divided into several monochromatic sub-images using the approach known from the CFA image compression literature, e.g. [23]. Nonetheless, such a scheme does not exploit the inter-channel correlation which is useful to reduce noise. A wavelet based CFA image

denoising method was proposed in [24]. Recently, Zhang *et al.* [25] proposed a principal component analysis (PCA) based CFA image denoising scheme. A CFA block, which contains color samples from different channels, is employed to analyze the local CFA image structure. By transforming the noisy signal into the PCA domain, the signal energy can be better clustered and the noise can be more effectively removed. The PCA based CFA image denoising algorithm effectively exploits the spatial and spectral correlation in each frame. This scheme was later extended to gray level images [26].



**Fig. 1: The three stages of the proposed video denoising and demosaicking scheme for noisy CFA video.**

This paper presents a spatial-temporal denoising and demosaicking algorithm for noisy CFA videos. The proposed algorithm has three stages, as illustrated in Figure 1. The first stage is to denoise the CFA video. We extend the PCA-based CFA image denoising method in [25] to a spatial-temporal one and apply it to CFA videos. Since the CFA video frames are red, green and blue color interlaced mosaic images, we consider each CFA patch, which contains pixels from all the three channels, as the basic unit in the denoising. By performing patch matching in the current frame and adjacent frames, the similar patches to the given one can be located. By viewing the elements in the CFA patch as variables and viewing the pixels in the matched patches the samples of them, the PCA transformation matrix can be computed and the denoising is then performed in the PCA domain. In the second stage, an initial CDM is applied to the denoised CFA video, resulting in a full-color video sequence. In the third stage, spatial-temporal post-processing is applied to the demosaicked video to reduce CDM artifacts, which are caused by either the limited color sampling frequency in the CFA mosaic pattern or the noise residual in the denoising stage.

The rest of the paper is structured as follows. Section 2 presents the CFA video denoising method, and Section 3 presents the CFA video demosaicking and post-processing steps. Section 4 presents extensive experimental results and Section 5 concludes the paper.

## 2. CFA Video Denoising

### 2.1 Noisy CFA Video Sequence

The CFA video sequence can be represented as a three dimensional (3D) cube. We denote by  $F^c(i, j, k)$  a pixel locating at spatial position  $(i, j)$  and in frame  $k$ . The superscript  $c \in \{r, g, b\}$  represents the color channel of the current pixel and  $c$  depends on the spatial position  $(i, j)$ . We employ the widely used Bayer CFA pattern [1] for discussion in this paper. There is:

$$c = \begin{cases} r & i = 2 \cdot z_i + 1; j = 2 \cdot z_j \\ b & i = 2 \cdot z_i; j = 2 \cdot z_j + 1 \\ g & \text{other positions} \end{cases} \quad (2-1)$$

where  $z_i$  and  $z_j$  are non-negative integers. In the Bayer pattern, the R (red), G (green) and B (blue) samples are interlaced, with the double sampling frequency of G compared to the R and B channels.

Noise will be inevitably introduced in the CFA video sequence, which makes digital video visually unpleasing. It is widely accepted that the corrupted noise in CCD and CMOS sensors is signal-dependent [27-28]. In [19], Hirakawa modeled the raw sensor output as  $y = x + (k_0 + k_1 x)v$ , where  $x$  is the desired noiseless signal,  $v \in \mathcal{N}(0,1)$  is unit Gaussian white noise and  $k_0$  and  $k_1$  are sensor dependent parameters. Although this noise model is quite realistic, the design of denoising algorithms will be very complex and computationally expensive. One simple and widely used noise model is the additive noise model  $y = x + v$ . In [20], Zhang *et al.* proposed a channel-dependent noise model for CFA data, which is adopted in this paper. The noisy video can be represented as

$$F_v^c(i, j, k) = F^c(i, j, k) + v_c(i, j, k), \quad c \in \{r, g, b\} \quad (2-2)$$

where  $v_r$ ,  $v_g$  and  $v_b$  are the noise signals in the red, green and blue channels. We assume that  $v_r$ ,  $v_g$  and  $v_b$  are zero-mean white and mutually uncorrelated in the spatial-temporal domain. The standard deviations of  $v_r$ ,  $v_g$  and  $v_b$  are denoted by  $\sigma_g$ ,  $\sigma_r$  and  $\sigma_b$ , respectively.

## 2.2 The Denoising Strategy

As mentioned in the introduction, if we perform denoising after CDM of the CFA sequence, the noise-caused color artifacts generated in the CDM process can be hard to remove in the denoising process. Recently, a denoising method which works directly on the CFA image was developed in [25]. To fully exploit the spectral correlation in the red, green and blue channels, the CFA block that contains pixels from all the three channels is taken as the basic denoising unit. The elements in the CFA block are modeled as a vector variable, and then the principal component analysis (PCA) is used to remove noise from the block. This method has shown promising denoising performance. In this paper, we adopt this denoising strategy and extend it to CFA videos.

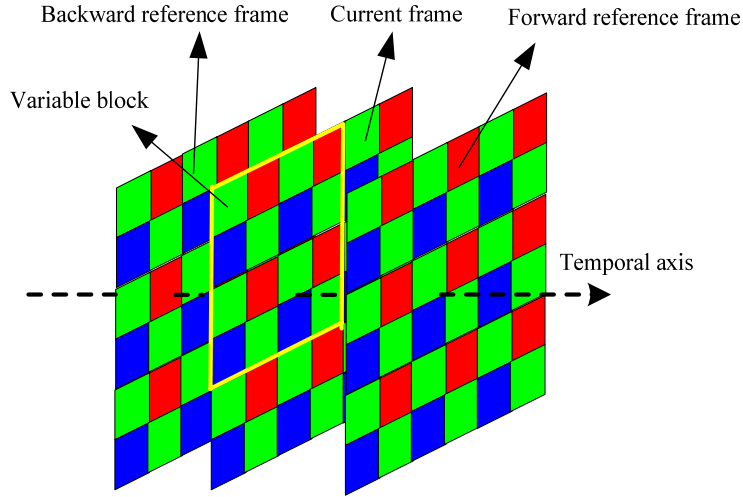


Fig. 2. A CFA block to be denoised in a frame.

Referring to Figure 2, for a current CFA frame to be denoised, we partition it into many blocks, e.g.  $4 \times 4$  or  $6 \times 6$  blocks. Denote by  $\mathbf{W}$  the block and suppose its size is  $w \times w$ . By taking each element in the block as a variable, we stretch the block to a variable vector  $\bar{x} = [x_1^c, x_2^c, \dots, x_m^c]^T$ , where  $m = w^2$ . Since the real data is noise corrupted, with the channel-dependent noise model in (2-2), we have

$$\bar{x}_v = [x_{1,v}^c, x_{2,v}^c, \dots, x_{m,v}^c]^T = \bar{x} + \bar{v} \quad (2-3)$$

where  $x_{i,v}^c = x_i^c + v_i^c$  is the noisy variable and  $\bar{v} = [v_1^c, v_2^c, \dots, v_m^c]^T$  is the noise vector. We denote by  $\mathbf{W}_v$  the noisy block of  $\mathbf{W}$ , i.e.  $\mathbf{W}_v = \mathbf{W} + \mathbf{V}$ , where  $\mathbf{V}$  is the noise block. Now the problem is how to remove the noise  $\bar{v}$  from  $\bar{x}_v$  so we can get a good estimation of  $\bar{x}$ , denoted by  $\hat{\bar{x}}$ .

In order to use PCA for denoising, we need a sample dataset so that the covariance matrix of the vector variable  $\bar{x}$  (or  $\bar{x}_v$ ) can be estimated. Clearly, we can construct the dataset by searching for similar blocks to the given block  $\mathbf{W}_v$  in its spatial-temporal neighborhood. Suppose  $n$  such blocks are found. With the condition that  $n \gg m$ , the PCA based denoising can effectively remove the noise from the noisy CFA data. Next we discuss the sample dataset construction.

### 2.3 Spatial-temporal Sample Dataset Construction

We employ the simple and fast block matching (BM) technique to find similar blocks to  $\mathbf{W}_v$ . Although BM is limited in terms of motion estimation (ME) accuracy, it is efficient enough for the application of denoising, and the PCA based denoising technique is robust to small BM bias (or the ME errors) because it calculates the statistics of the matched samples and a certain amount of BM bias will not affect much the final denoising result. To exploit both the spatial and temporal redundancy in denoising, we search for similar blocks to  $\mathbf{W}_v$  in the current frame and adjacent frames. For a given block  $\mathbf{W}_v$  and a reference block  $\mathbf{W}_{v,(\Delta i, \Delta j)}^{ref}$ , where  $(\Delta i, \Delta j)$  means the spatial displacement between the given block and the reference block, we calculate their  $L_2$  distance as

$$d_{(\Delta i, \Delta j)} = \left\| \mathbf{W}_v - \mathbf{W}_{v,(\Delta i, \Delta j)}^{ref} \right\| = \frac{1}{W^2} \sum_{h=1}^w \sum_{l=1}^w \left( \mathbf{W}_v(h, l) - \mathbf{W}_{v,(\Delta i, \Delta j)}^{ref}(h, l) \right)^2 \quad (2-4)$$

where  $\|\bullet\|$  is the  $L_2$  norm operator. Due to the mosaic CFA pattern (the Bayer pattern is used here), the displacement  $(\Delta i, \Delta j)$  should satisfy the constraint  $(\Delta i, \Delta j) = (2z_i, 2z_j)$ , where  $z_i$  and  $z_j$  are integers. In practice,  $(\Delta i, \Delta j)$  will be in a suitable search range:  $\{-Z_i \leq \Delta i \leq Z_i, -Z_j \leq \Delta j \leq Z_j\}$ , where  $Z_i$  and  $Z_j$  are even integers. To robustly construct the dataset, we select the first  $n$  most similar blocks (i.e. the ones who have the smallest distances  $d_{(\Delta i, \Delta j)}$ ) to the given block  $\mathbf{W}_v$  in consecutive frames and within the search range. In the following PCA based denoising process, these  $n$  sample blocks will be used to calculate the co-variance matrix of the variable vector.

## 2.4 PCA-based Denoising

By using the method in sub-section 2.3, we can find  $n$  blocks, including the given block and  $n-1$  reference blocks similar to it. These  $n$  blocks are viewed as the sample blocks for the vector variable  $\bar{x}_v$  (referring to Eq. (2-3)). In other words, for each element of the vector variable  $\bar{x}_v$ , there are  $n$  samples of it. We can format these data into an  $m \times n$  matrix

$$\mathbf{X}_v = \begin{bmatrix} X_{1,1}^{c,v} & X_{1,2}^{c,v} & \cdots & X_{1,n}^{c,v} \\ X_{2,1}^{c,v} & X_{2,2}^{c,v} & \cdots & X_{2,n}^{c,v} \\ \vdots & \vdots & \vdots & \vdots \\ X_{m,1}^{c,v} & X_{m,2}^{c,v} & \cdots & X_{m,n}^{c,v} \end{bmatrix} \quad (2-5)$$

where each column of the matrix  $\mathbf{X}_v$  is the vector stretched by the sample block. The superscript “ $v$ ” means that the samples are noise corrupted and the superscript “ $c$ ” belongs to  $\{r,g,b\}$ , indicating the spectral channel of the sample. We let the first column be the sample vector to be denoised. The detailed PCA-based denoising procedures can be found in [25]. Denote by  $\hat{\mathbf{X}}$  the denoised dataset of  $\mathbf{X}_v$ , the denoised CFA block can then be extracted from the first column of  $\hat{\mathbf{X}}$ .

## 3. Spatial-Temporal Color Demosaicking

In Section 2, the noise in the CFA video sequence has been reduced. Thus CDM can be subsequently performed to reproduce the full color video sequence. We first apply an initial CDM to each frame, and then exploit the spatial-temporal redundancy to reduce the CDM errors.

### 3.1 Initial Color Demosaicking

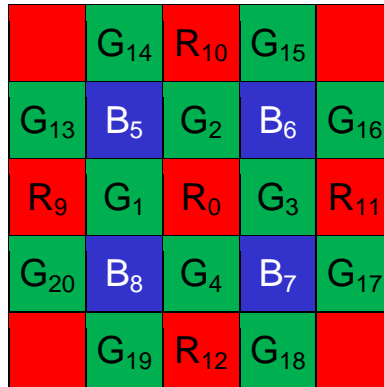


Fig. 3. A CFA block centered on a red pixel.

In [2], Adam and Hamilton proposed the well-known second order Laplacian correction (SOLC) filtering algorithm for CDM. In SOLC, the horizontal and vertical gradients of each pixel are first calculated. If the horizontal gradient is less than that in the vertical direction, then CDM is performed along the horizontal direction by using the SOLC filter, vice versa. Such a strategy avoids interpolating across edges. In this section, we improve the SOLC scheme by fusing the filtering outputs at horizontal and vertical directions, instead of keeping only one of them.

The missing green color components are first interpolated. Refer to Figure 3, considering a CFA block centered on a red pixel  $R_0$ , where the green color is to be estimated. (The missing green colors on blue pixels can be similarly interpolated.) By using the SOLC filter, we can interpolate the missing green color  $G_0$  along horizontal direction as  $\hat{G}_0^h = \frac{1}{2}(G_1 + G_3) + \frac{1}{4}(2 \cdot R_0 - R_9 - R_{11})$ . Similarly, we can interpolate  $G_0$  along vertical direction as  $\hat{G}_0^v = \frac{1}{2}(G_2 + G_4) + \frac{1}{4}(2 \cdot R_0 - R_{10} - R_{12})$ . The two estimates  $\hat{G}_0^h$  and  $\hat{G}_0^v$  can be fused for a more accurate estimation of  $G_0$  as follows:

$$\hat{G}_0 = w_h \hat{G}_0^h + w_v \hat{G}_0^v \quad (3-1)$$

where  $w_h + w_v = 1$ . To determine the weights  $w_h$  and  $w_v$ , we first calculate the gradients along horizontal and vertical directions within the CFA block as follows

$$d_h = |G_1 - G_3| + |2 \cdot R_0 - R_9 - R_{11}| + \frac{1}{2}|B_5 - B_6| + \frac{1}{2}|B_7 - B_8| + \frac{1}{2}|2 \cdot G_2 - G_{13} - G_{16}| + \frac{1}{2}|2 \cdot G_4 - G_{17} - G_{20}| \quad (3-2)$$

$$d_v = |G_2 - G_4| + |2 \cdot R_0 - R_{10} - R_{12}| + \frac{1}{2}|B_5 - B_8| + \frac{1}{2}|B_6 - B_7| + \frac{1}{2}|2 \cdot G_1 - G_{14} - G_{19}| + \frac{1}{2}|2 \cdot G_3 - G_{15} - G_{18}| \quad (3-3)$$

In general, a bigger gradient in one direction means more variations in that direction. Therefore, if the horizontal gradient is bigger than the vertical gradient, it should contribute less to the final estimate; vice versa. We set the weights to be

$$w_h = d_v / (d_v + d_h); w_v = d_h / (d_v + d_h) \quad (3-4)$$

Once all the missing green colors are estimated, we can further interpolate the missing blue (red) colors at red (blue) positions. For example, at position  $R_0$ , the missing blue color can be estimates as



$\hat{B}_0 = \frac{1}{4} \sum_{i=5}^8 (B_i - \hat{G}_i) + \hat{G}_0$ . Finally, the missing red and blue colors at green positions can be

interpolated. For example, the missing red color at  $G_1$  can be interpolated as

$\hat{R}_1 = \frac{1}{4} (R_0 + R_9 + \hat{R}_5 + \hat{R}_8) - \frac{1}{4} (\hat{G}_0 + \hat{G}_9 + \hat{G}_5 + \hat{G}_8) + \hat{G}_1$ . Then the initial CDM is completed.

### 3.2 Spatial-temporal Post-processing

After initial CDM, a full color video sequence is obtained. However, there can be many CDM errors in high variation areas due to the lack of spatial redundancy. Therefore, a post-processing is necessary to reduce the CDM errors and enhance the video quality. Inspired by the non-local means and collaborative filtering techniques [16-18] in image denoising, for a given pixel to be enhanced, we search for the similar pixels to it within the spatial-temporal neighborhood and then let the enhanced pixel be the weighted average of them.

Denote by  $(R_0, G_0, B_0)$  the triplet of the current pixel to be enhanced. We denote by  $\mathbf{W}_0$  a local 3D window (e.g.  $5 \times 5 \times 3$ ) centered on  $(R_0, G_0, B_0)$ . For a spatial-temporal neighborhood triplet of  $(R_0, G_0, B_0)$ , denoted by  $(R_i, G_i, B_i)$ , its local 3D window is denoted by  $\mathbf{W}_i$ . The similarity between  $(R_0, G_0, B_0)$  and  $(R_i, G_i, B_i)$  can be measured by the distance between the two blocks  $\mathbf{W}_0$  and  $\mathbf{W}_i$ :  $d_i = \|\mathbf{W}_0 - \mathbf{W}_i\|$ .

Let  $\mathbf{W}_0^c$  and  $\mathbf{W}_i^c$ ,  $c \in \{r, g, b\}$ , be the red, green or blue channels of  $\mathbf{W}_0$  and  $\mathbf{W}_i$ , we can readily have

$$\begin{aligned} d_i &= \|\mathbf{W}_0^r - \mathbf{W}_i^r\| + \|\mathbf{W}_0^g - \mathbf{W}_i^g\| + \|\mathbf{W}_0^b - \mathbf{W}_i^b\| \\ &= d_i^r + d_i^g + d_i^b \end{aligned} \quad (3-5)$$

where  $d_i^c = \|\mathbf{W}_0^c - \mathbf{W}_i^c\|$ ,  $c \in \{r, g, b\}$ .

In general, the smaller the distance  $d_i$  is, the higher the similarity between  $(R_0, G_0, B_0)$  and  $(R_i, G_i, B_i)$  is. We choose the first  $l$  most similar pixels to  $(R_0, G_0, B_0)$ , including itself, in the enhancement of it. Suppose  $(R_i, G_i, B_i)$ ,  $i=0, 1, \dots, l-1$ , are the selected most similar pixels to  $(R_0, G_0, B_0)$ . The enhanced pixel is the weighted average of them:

$$\hat{R}_0 = \sum_{i=0}^{l-1} w_i^r R_i, \quad \hat{G}_0 = \sum_{i=0}^{l-1} w_i^g G_i, \quad \hat{B}_0 = \sum_{i=0}^{l-1} w_i^b B_i \quad (3-6)$$

where the weights  $w_i^r$ ,  $w_i^g$  and  $w_i^b$  are set as

$$w_i^c = \frac{1}{C_c} \exp(-d_i^c / \lambda), \quad c \in \{r, g, b\} \quad (3-7)$$

$\lambda$  is a parameter to control the decay speed of weights w.r.t  $d_i^c$ , and constant  $C_c = \sum_{i=0}^{l-1} \exp(-d_i^c / \lambda)$  is to normalize the weights. Clearly, the higher the distance  $d_i^c$  is, the lower the weight  $w_i^c$  is.

Like in the spatial-temporal denoising process in Section 2, in the spatial-temporal post-processing of CDM there is an implicit ME process. The ME is replaced by a similar patch matching process in both current frame and adjacent frames. Different from the traditional ME that usually outputs a best-matched reference block (per reference frame) to the given block, the proposed scheme find many similar blocks to the given one for noise reduction and image enhancement. Such an implicit ME could effectively exploit more spatial-temporal redundancies in couple with the subsequent denoising and post-processing procedures. One advantage of it is that it is robust to patch matching errors. Even if some of the matched blocks are inaccurately found, they would not degrade much the final results.

#### 4. Experimental Results

The proposed spatial-temporal CDM method for noisy CFA videos is tested by using both simulated CFA sequences and real CFA sequences. We compare it with the following state-of-the-art schemes<sup>2</sup>:

- (1) The single frame joint denoising and color demosaicking (JDC) schemes [19] and [20].
- (2) Demosaicking first and denoising later schemes. We first use the advanced CDM algorithms [4] and [5] to demosaick the noisy CFA video and then use the KSVD [15] and benchmark BM3D [17] algorithms to denoise the demosaicked video.
- (3) Denoising first and demosaicking later scheme. We first use the CFA denoising scheme in [25] to denoise each frame and then use the temporal CDM scheme in [9] to demosaick the video.

Among the above schemes, the proposed method has the highest complexity. This is mainly because it involves a PCA transformation stage, which needs to calculate the covariance matrix of the dataset and

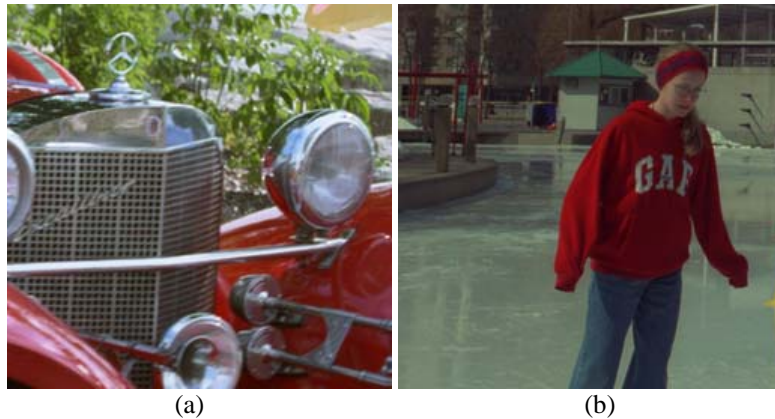
---

<sup>2</sup> We thank the authors of [19], [5], [15] and [17] for sharing their codes.

apply singular value decomposition (SVD). Apart from the part of PCA transformation, the proposed method has similar complexity to the BM3D scheme.

We use two simulated CFA sequences and two real CFA sequences in the following experiments. The parameters in the proposed method are set as follows. Nine adjacent frames are used in the spatial-temporal denoising and demosaicking. In the denoising stage, the size of variable block is  $m=6\times6=36$  and  $n=150$  sample blocks are used for PCA training. In the spatial-temporal post-processing,  $l$  (refer to Eq. (3-6)) is 10 and  $\lambda$  is 4 (refer to Eq. (3-7)). Due to the limitation of space, only partial experimental results are shown here. The code and reproduced videos of the proposed algorithm can be downloaded at <http://www4.comp.polyu.edu.hk/~cslzhang/CFA-Video-RP.htm>.

#### 4.1 Experiments on Simulated CFA Videos



**Fig. 4. Scenes of the two video sequences. (a) The Car sequence; (b) the Skater sequence.**

Two full color video sequences<sup>3</sup>, which were first captured by a film video camera and then scanned by a scanner, are used in this experiment. We down-sampled them according to the Bayer pattern to simulate the CFA sequences. The original spatial resolution of the videos is  $1024\times768$ . We cropped the most difficult  $256\times256$  portion of them and used 24 frames in the experiment. Figure 4 shows the scenes of the two cropped sequences. To simulate noisy CFA sequence, we added Gaussian white noise to each color channel. For a fair comparison with other methods, two cases were considered. In the first case, the noise levels in the three channels are set to be the same:  $\sigma_r = \sigma_g = \sigma_b = 15$ . In the

<sup>3</sup> We thank the IMAX Corporation, Toronto, Canada, for sharing the video sequences with us.

second case, the noise levels are channel-dependent, and we let  $\sigma_r = 19, \sigma_g = 14, \sigma_b = 15$  for the car sequence and  $\sigma_r = 18, \sigma_g = 19, \sigma_b = 14$  for the skater sequence. Since the denoising methods KSVD [15] and BM3D [17] do not use a channel-dependent noise model, we apply an equivalent noise level  $\sigma = \frac{1}{2}\sqrt{\sigma_r^2 + 2\sigma_g^2 + \sigma_b^2}$  to each channel when using them in the second case.

**Table I. PSNR (dB) results on the *Car* sequence.**

Demosaicking & denoising methods	$\sigma_r = \sigma_g = \sigma_b = 15$			$\sigma_r = 19, \sigma_g = 14, \sigma_b = 15$		
	R	G	B	R	G	B
JDC [19]	26.6	26.4	23.7	26.2	26.4	23.6
JDC [20]	28.3	28.9	25.9	27.8	29.0	25.9
[5] + KSVD [15]	29.1	29.7	26.1	28.3	29.8	26.3
[4] + KSVD [15]	29.8	29.7	26.5	28.2	29.7	25.9
[25] + [9]	30.1	29.9	26.7	29.4	30.0	26.7
[5] + BM3D [17]	29.2	30.8	27.1	28.8	31.1	27.2
[4] + BM3D [17]	30.2	<b>31.2</b>	27.7	29.3	<b>31.4</b>	27.6
Proposed (before stage 3)	30.5	30.9	27.4	30.2	30.9	27.4
Proposed (after stage 3)	<b>31.9</b>	<b>31.2</b>	<b>28.2</b>	<b>31.3</b>	31.3	<b>28.1</b>

**Table II. PSNR (dB) results on the *Skater* sequence.**

Demosaicking & denoising method	$\sigma_r = \sigma_g = \sigma_b = 15$			$\sigma_r = 18, \sigma_g = 19, \sigma_b = 14$		
	R	G	B	R	G	B
JDC [19]	30.6	31.6	32.3	30.1	31.0	32.1
JDC [20]	31.5	32.6	32.7	30.6	31.3	32.5
[5] + KSVD [15]	31.8	33.6	33.7	29.9	31.3	31.7
[4] + KSVD [15]	32.6	33.6	34.1	31.0	32.0	33.5
[25] + [9]	32.6	32.9	33.4	31.7	31.6	32.8
[5] + BM3D [17]	32.1	34.3	33.9	31.3	33.0	33.6
[4] + BM3D [17]	33.1	34.5	34.5	32.2	33.2	34.6
Proposed (before stage 3)	33.0	34.3	35.0	32.2	32.9	34.4
Proposed (after stage 3)	<b>34.2</b>	<b>34.6</b>	<b>35.7</b>	<b>33.3</b>	<b>33.3</b>	<b>35.4</b>

Tables I and II list the average PSNR results of the reconstructed color video sequences by different schemes. We can see that the proposed method achieves the best PSNR results. Particularly, the proposed method reconstructs much better the R and B channels than the other schemes. This is because the sampling frequency of R and B channels is half of that of G, and hence they are more difficult to reproduce than the G channel. If we apply CDM first, there will be more noise-caused

CDM errors in R and B channels than in G, and even the benchmark BM3D scheme may not denoise R and B very well. By using PCA based denoising on the CFA video, which is an adaptive and multivariate filtering technique, the proposed method reduces many noise-caused CDM artifacts, and hence it can reproduce R and B much better than other schemes. It can also be found that the scheme “LMMSE\_CDM+BM3D\_Denoising” (i.e. “[4]+BM3D [17]”) works the second best. This also validates that BM3D is powerful in denoising and it can suppress many noise-caused CDM artifacts.

Figures 5 ~ 6 show one frame of the reproduced full color sequences by different schemes. It can be seen that the result by the proposed method has the best visual perception, producing the fewest color artifacts and preserving well the edge structures. The “LMMSE\_CDM+BM3D\_Denoising” scheme also yields reasonably good results. However, it shows more color artifacts, which are mostly generated in the CDM process due to the corruptive noise. The reproduced videos by the proposed method and the “LMMSE\_CDM+BM3D\_Denoising” method can be downloaded at website <http://www4.comp.polyu.edu.hk/~cslzhang/CFA-Video-RP.htm>.

## 4.2 Experiments on Real CFA Videos

This section presents the experimental results on two real CFA sequences (*Bear* and *Notebook*), which were captured by a single CCD video camera (model: Flea2 by Point Grey Research, Inc.) at a frame rate of 30fps. The original spatial size of the two CFA videos is 1280×960 and we cropped a 360×360 portion of them in the experiment. 60 frames of the *Bear* sequence and 45 frames of the *Notebook* sequence were used. The noise levels in the sequences are estimated as follows. We divide the  $N \times M$  CFA frame into four  $N/2 \times M/2$  sub-images (two green sub-images, one red sub-image and one blue sub-image), and then apply one-stage orthogonal wavelet transform to each sub-image. The noise level is estimated as  $\sigma = \text{Median}(\mathbf{w})/0.6475$  [11], where  $\mathbf{w}$  is the diagonal sub-band at the first stage. For the green channel, the noise level is the average of the values of  $\sigma$  from the two green sub-images. The estimated average noise levels for the *Bear* and *Notebook* sequences are  $\sigma_r = 13, \sigma_g = 12, \sigma_b = 26$  and  $\sigma_r = 15, \sigma_g = 12, \sigma_b = 26$ , respectively.

To save space, we only present the results by the proposed method and the “LMMSE\_CDM+BM3D\_Denoising” method in Figure 7. As in Section 4.1, we see that the proposed method achieves better color reproduction by reducing much the noise-caused color artifacts and preserving well the image edge details. The reproduced videos by the two methods can be downloaded at website <http://www4.comp.polyu.edu.hk/~cslzhang/CFA-Video-RP.htm>.

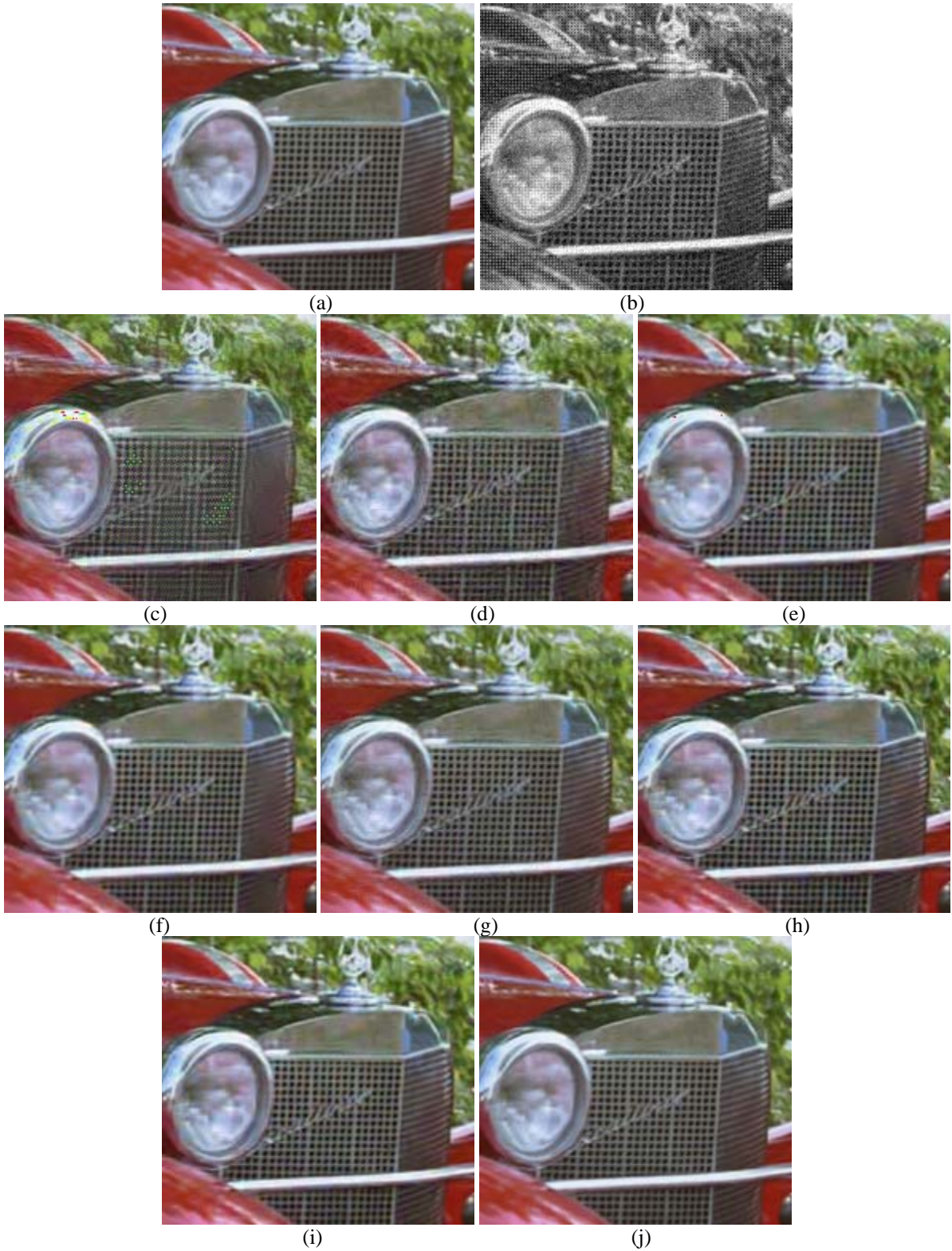
## 5. Conclusion

This paper presented a spatial-temporal color video reproduction algorithm from the noisy color filter array (CFA) sequence. The proposed method has three steps: principal component analysis based spatial-temporal CFA denoising, initial spatial color demosaicking (CDM) and spatial-temporal enhancement. The spatial-temporal redundancies existing in the CFA sequence are effectively exploited to reproduce and enhance the color videos. Two simulated and two real noisy CFA sequences were used to evaluate the proposed method in comparison with state-of-the-art denoising and CDM schemes, including the benchmark BM3D algorithm. The experimental results showed that the proposed method achieves the best color video reproduction in terms of both PSNR and subjective quality, preserving well the image edge structures and suppressing effectively the color artifacts.

## Reference

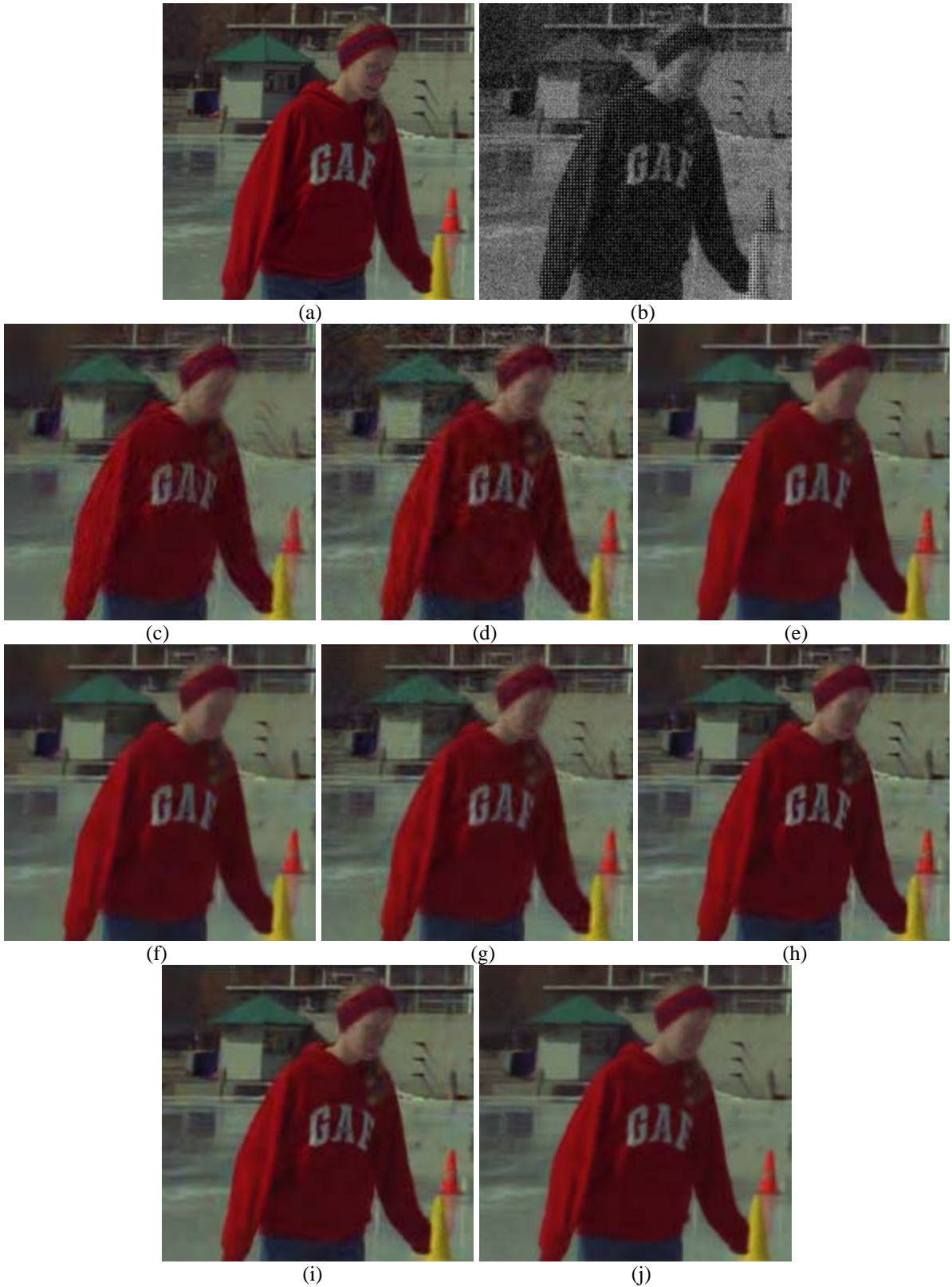
- [1] B. E. Bayer and Eastman Kodak Company, “Color Imaging Array,” US patent 3 971 065, 1975.
- [2] J. E. Adams and J. F. Hamilton Jr., “Adaptive color plane interpolation in single color electronic camera,” U. S. Patent, 5 506 619, 1996.
- [3] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer and R. M. Mersereau, “Demosaicking: Color filter array interpolation in single-chip digital cameras,” *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp.44-54, Jan. 2005.
- [4] Lei Zhang and X. Wu, “Color demosaicking via directional linear minimum mean square-error estimation,” *IEEE Trans. on Image Processing*, vol. 14, no. 12, pp. 2167-2178, Dec. 2005.
- [5] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm”, *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 360-369, Mar. 2005.
- [6] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, and A.N. Venetsanopoulos, “Vector filtering for color imaging,” *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, vol. 22, no. 1, pp. 74-86, Jan. 2005.
- [7] Xin Li, “Demosaicing by successive approximation,” *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 370-379, Mar. 2005.
- [8] Xin Li, B. Gunturk, L. Zhang, “Image demosaicing: a systematic survey,” *Visual Communications and Image Processing 2008*, Proceedings of the SPIE, Volume 6822, pp. 68221J-68221J-15 (2008). San Jose, CA, USA.
- [9] X. Wu and L. Zhang, “Improvement of color video demosaicking in temporal domain,” *IEEE*

- Trans. on Image Processing*, vol. 15, Oct. 2006.
- [10] X. Wu and L. Zhang, "Color video demosaicking via motion estimation and data fusion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, pp. 231-240, Feb. 2006.
  - [11] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425-455, 1994.
  - [12] A. Pizurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 654-665, Mar. 2006.
  - [13] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. on Image Processing*, vol. 12, no. 11, pp. 1338 - 1351, Nov. 2003.
  - [14] D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," *ICIP 2003*, vol. 1, pp. I101-I104, 14-17 Sep. 2003.
  - [15] M. Aharon, M. Elad, and A.M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, November 2006.
  - [16] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multisc. Model. Simulat.*, vol. 4, no. 2, pp. 490-530, 2005.
  - [17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. on Image Processig*, vol. 16, no. 8, pp. 2080-2095, Aug. 2007.
  - [18] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering" *Proc. SPIE Electronic Imaging 2008*, vol. 6812, no. 6812-1D, San Jose (CA), USA, January 2008.
  - [19] K. Hirakawa and T. W. Parks, "Joint demosaicking and denoising," *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 2146 - 2157, Aug. 2006.
  - [20] L. Zhang, X. Wu and D. Zhang, "Color reproduction from noisy CFA data of single sensor digital cameras," *IEEE Trans. on Image Processing*, vol. 16, no. 9, pp. 2184-2197, Sept. 2007.
  - [21] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy Data," *Int'l J. of Imaging Systems and Technology, Special Issue on Applied Color Image Processing*, vol. 17, pp. 105-122, 2007.
  - [22] D. Paliy, M. Trimeche, V. Katkovnik, S. Alenius, "Demosaicing of noisy data: spatially adaptive approach," *Proceedings of the SPIE*, vol. 6497, pp. 64970K, 2007.
  - [23] C. C. Koh, J. Mukherjee and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1448-1456, Nov. 2003.
  - [24] K. Hirakawa, X.-L. Meng, P.J. Wolfe, "A framework for wavelet-based analysis and processing of color filter array images with applications to denoising and demosaicing," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. I-597-600, April 2007.
  - [25] L. Zhang, R. Lukac, X. Wu and D. Zhang, "PCA-based Spatial Adaptive Denoising of CFA Images for Single-Sensor Digital Cameras," *IEEE Trans. on Image Processing*, vol. 18, no. 4, pp. 797-812, April 2009.
  - [26] L. Zhang, W. Dong, D. Zhang and G. Shi, "Two-stage Image Denoising by Principal Component Analysis with Local Pixel Grouping," to appear in *Pattern Recognition*.
  - [27] A. Foi, S. Alenius, V. Katkovnik, and K. Egiazarian, "Noise measurement for raw-data of digital imaging sensors by automatic segmentation of nonuniform targets," *IEEE Sensors Journal*, vol. 7, pp. 1456-1461, 2007.
  - [28] A. Foi, V. Katkovnik, D. Paliy, K. Egiazarian, M. Trimeche, S. Alenius, R. Bilcu, and M. Vehvilainen, "Apparatus, method, mobile station and computer program product for noise estimation, modeling and filtering of a digital image," U.S. Patent Application No. 11/426,128, 2006.



**Fig. 5.** One reconstructed full color frame of the *Car* sequence. (a) Original full-color image; (b) noisy CFA image ( $\sigma_r = \sigma_g = \sigma_b = 15$ ); reconstructed images by methods (c) [19]; (d) [20]; (e) [5] + KSVD [15]; (f) [4] + KSVD; (g) [25] + [9]; (h) [5] + BM3D [17]; (i) [4] + BM3D [17]; and (j) proposed.





**Fig. 6.** One reconstructed full color frame of the *Skater* sequence. (a) Original full-color image; (b) noisy CFA image ( $\sigma_r = \sigma_g = \sigma_b = 15$ ); reconstructed images by methods (c) [19]; (d) [20]; (e) [5] + KSVD [15]; (f) [4] + KSVD; (g) [25] + [9]; (h) [5] + BM3D [17]; (i) [4] + BM3D [17]; and (j) proposed.



**Fig. 7.** Left column: one frame of the reconstructed *Bear* sequence from the raw CFA sequence (estimated noise level:  $\sigma_r = 13, \sigma_g = 12, \sigma_b = 26$ ). Right column: one frame of the reconstructed *Notebook* sequence from the raw CFA sequence (estimated noise level:  $\sigma_r = 15, \sigma_g = 12, \sigma_b = 26$ ). From top to bottom: original noisy CFA image, reconstructed full color image by [4]+BM3D [17] and the proposed method.