Additive Spread-Spectrum Watermark Detection in Demosaicked Images

Peter Meerwald and Andreas Uhl
Department of Computer Sciences
University of Salzburg
Jakob-Haringer-Str. 2, A-5020 Salzburg, Austria
{pmeerw,uhl}@cosy.sbg.ac.at

ABSTRACT

In this paper we investigate watermarking of digital camera raw images and blind detection of spread-spectrum watermarks in demosaicked images. We propose straightforward watermark embedding in sensor data combined with a novel detector. To this end, we extend a detection approach which adaptively combines the components of the demosaicked image to take advantage of the interpolated and correlated image structure within and between color channels. Experimental results confirm the benefits of the novel detection approach. Further, we experimentally assess the impact of several demosaicking methods on the detection performance.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*; K.4.4 [**Computer and Society**]: Electronic Commerce—*Security*

General Terms

Algorithms, Security

Keywords

Watermarking detection, demosaicking, color filter array, raw images, copyright protection

1. INTRODUCTION

Digital cameras are in ubiquitous use. Most popular digital cameras use a single, monochrome image sensor with a color filter array (CFA) on top, often arranged in the Bayer pattern, see Figure 1. In order to provide a full-resolution RGB image, the sensor data has to be interpolated – a process called demosaicking – as well as color, gamma and white point corrected [29]. Many different demosaicking techniques exist, see [16, 2] for an overview, yet the basic processing steps are shared by most camera implementations. While the JPEG image format is widely use to store the processed image data, most cameras also permit to store the unprocessed, raw sensor data. The latter can be considered the most valuable image asset and the digital equivalent of the analog film negative.

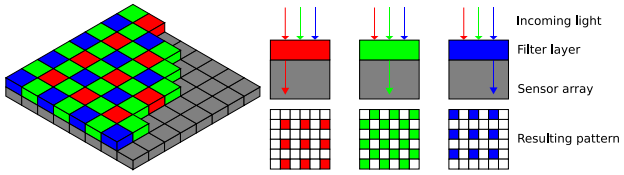

Figure 1: Color filter array (CFA) arranged in the popular Bayer pattern

The digital nature of the recorded images allows for easy duplication and manipulation and poses challenges when these images are to be used as evidence in court or when resolving ownership claims. Active techniques, such as watermarking methods [7] that imperceptibly embed a pseudo-random signal in the image data, as well as passive or forensic approaches have been suggested to address image integrity verification, camera identification and ownership resolution. Many different forensic techniques have been proposed to detect image forgeries by exploiting camera characteristics to link an image to a specific camera or to confirm that certain processing artefacts are preserved. For example, Chen et al. [5] exploit the inherent Photo-Response Non-Uniformity (PRNU) noise of the image sensor for camera identification and image integrity verification. Interpolation artefacts due to demosaicking are used by Popescu et al. [28] to verify the integrity of the image. Passive techniques have the disadvantage that camera characteristics such as PRNU have to be estimated before use.

Surprisingly, watermarking is generally not integrated in the early stages of the image acquisition processes but added later-on e.g. during JPEG compression [3]. Although the raw image data is probably the most valuable asset, very limited research has been published on watermark protection of the sensor data. One reason might be that the image data volume and constrained power resources of digital cameras demand efficient processing, favoring simple and hardware-based solutions. In addition, it is not clear how the image processing pipeline and the demosaicking step in particular affect a watermark embedded in the sensor data. Nelson et al. [25] propose a CMOS image sensor with watermarking

MM&Sec'09, September 7–8, 2009, Princeton, New Jersey, USA.

capabilities that adds pseudo-random noise. Meerwald et al. [22] propose a software-only solution adding an additive, pseudo-random spread-spectrum watermark in camera firmware. Mohanty et al. [24] describe a quantization-based hardware implementation for combined robust and fragile watermarking. Blythe et al. [3] discuss a secure digital camera which uses lossless watermarking to embed a biometric identifier of the photographer together with a cryptographic hash of the image data. Their embedding method efficiently changes the JPEG quantization tables and DCT coefficients but precludes watermarking of raw images.

Digital watermarking has to be applied close to the image acquisition stage in order to protect the copyright of both, the raw and compressed image. In Section 2 we briefly review main directions in image demosaicking and present several available implementations. We consider watermarking the raw CFA sensor data before the demosaicking stage of a typical image processing pipeline [29] as outlined in Section 3. For watermark detection, we adopt a scheme that adaptively combines the polyphase components of an interpolated image [10] and extend the approach for demosaicked images in Section 4, taking advantage also of the correlation between color channels. In Section 5, we analyze the performance of the novel detection approach after JPEG compression. Concluding remarks are offered in Section 6.

2. DEMOSAICKING

Demosaicking (or demosaicing) is a term describing color interpolation techniques to reconstruct the missing color information from image sensor data overlaid with a color filter array (CFA). Most often the 2×2 Bayer RGB pattern is assumed, but most methods are also applicable to other pixel arrangements or color models.

A tremendous amount of research has been published recently on image demosaicking. In the following, we only consider algorithms for which implementations are available from the authors. Although demosaicking can be solved by standard image interpolation techniques (e.g. bilinear or edge-directed interpolation (EDI) [17]), exploiting the *intra* and *inter* channel dependencies can significantly improve the visual appearance of the full resolution image. Note that the actual implementation in a digital cameras is generally not known and might be covered by patents [2].

Table 1 provides an overview of the demosaicking methods considered in this paper. Most demosaicking methods belong to the class of *sequential* interpolation algorithms [16] where first the luminance (green) channel is reconstructed which then aids in recovering the chrominance channels. Edge-directed interpolation of the green component forms the basis of many sequential, spatial domain demosaicking methods: adaptive color plane interpolation (ACPI) [12], primary-consistent soft-decision demosaicking (PCD) [33], directional linear mean square-error estimation (DLMMSE) [34], data-adaptive filtering (DAF) [21], variance of color differences (VCD) method [6], directional filtering (DF) [23], spatial and spectral correlation based demosaicking (SSC) [4]. Alternatively, the missing luminance data can also be recovered by frequency-domain filtering approaches such as filtering based on HVS properties (FHVS) [1], frequency domain filtering (FD) [9], adaptive filtering (AF) [18] or adaptive homogeneity-directed (AHD) filtering [13].

The last scheme is the basis for many open-source implementations such as `dcraw`[1].

Having a full-resolution luminance (green) channel facilitates the recovery of the chrominance channels by enforcing constant hue rules. The assumption is that the ratio of color differences (i.e. R-G, B-G) is constant within image objects. The color-difference signals are interpolated based on the full-resolution green channel and the down-sampled chrominance channels [12]. Later methods include effective color interpolation (ECI) [27] and normalized color-ratio modeling (NCRM) [20].

In order to suppress visually annoying artefacts such as false colors and zipper patterns, several post-processing methods such as median filtering [19], iterative methods based on alternating projections (AP) [11], successive approximation (SA) [15] as well as three-step demosaicking (TSD) [31] have been put forward. Recently, combined denoising and demosaicking techniques have been proposed including the joint denoising and demosaicking (JDD) [35] and local polynomial approximation (LPA) [26] methods. Note that the watermark embedded in the CFA image data is a noise source independent from the image data. Two aspects are of interest: First, the watermark may interfere with the demosaicking approach which makes assumptions on the correlation between color bands to guide the image reconstruction process [4]. To avoid the problem, we embed the watermark in only one CFA component. Second, results indicate that joint denoising and demosaicking is superior to demosaicking followed by denoising [35, 14, 26] and thus the demosaicking stage may suppress noise (such as PRNU [5]) but also the watermark as a noise source.

3. WATERMARKING RAW IMAGE DATA

In this work, we focus on a simple additive, spread-spectrum watermark embedded in the spatial domain such as performed by the CMOS image sensor with watermarking capabilities described by Nelson et al. [25]. Alternatively, the software-only solution embedding the watermark in camera firmware can be employed [22]. After embedding the watermarked raw image can be stored at this point for later post-processing with third party software or, alternatively, the data is upsampled in the demosaicking stage of the camera and stored in the JPEG format. Watermarking the raw image data has the advantage that copyright protection is incorporated at an early point in the image life cycle. The most valuable original sensor data as well as all derived images are protected by the same watermark. On the downside, a watermarked raw image has to withstand many processing steps when turned into a full-resolution RGB image which we explore in Section 5.

Figure 2 depicts a simplified model of the image processing pipeline with the intercalated watermark embedding stage and the following demosaicking and JPEG compression stages. The CFA image data can be written as $x = [R_0\ G_0; G_1\ B_0]$ where R_0, G_0, G_1 and B_0 denote the CFA components' pixels according to the 2×2 Bayer pattern. Each CFA component is comprising N pixels. In the embedding stage we select one CFA component $c \in \{R_0, G_0, G_1, B_0\}$ and generate a pseudo-random bipolar spread-spectrum watermark w with the same size as c. w is derived from a secret seed value k identifying the copyright owner. The ad-

[1] `http://www.cybercom.net/~dcoffin/dcraw/`

Table 1: Classification of demosaicking methods

Intra	Sequential Inter Channel Interpolation	
	Spatial Domain	Frequency Domain
bilinear, EDI [17]	ACPI [12], DAF [21], DF [23], DLMMSE [34], ECI [27], JDD [35], LPA [26], NCRM [20], NM [19], PCD [33], SA [15], SSC [4], TSD [31], VCD [6]	AF [18], AHD [13], AP [11], FD [9], FHVS [1]

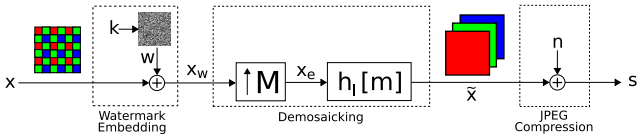

Figure 2: Watermarking embedding and image processing pipeline

ditive embedding operation can then be written as $c_w[\mathbf{m}] = c[\mathbf{m}]+\alpha\cdot w[\mathbf{m}]$ where $\mathbf{m} \in \mathbb{N}^2$ denotes pixel indices and $\alpha > 0$ controls the embedding strength. In the following we consider two cases: embedding in B_0, and in G_1, respectively, to construct the watermarked CFA data x_w.

The choice of the blue channel is motivated by the fact that the human visual system is least sensitive to blue color stimuli, thus embedding in the blue channel causes the least perceptual distortion. Most demosaicking methods first estimate the overall luminance from the green CFA pixels and then interpolate the chrominance components based on the estimate because there are twice as many green CFA pixels in the Bayer pattern (compare Fig. 1). Further, the green channel mainly contributes to the luminance information best preserved by JPEG compression, therefore we expect higher watermark robustness for green channel embedding. From the viewpoint of constrained processing resources (e.g. in software-only implementation [22]), computational effort can be saved by just processing one color band.

The schematic demosaicking stage comprises an upsampling operation and an interpolation filter which turns the watermarked data into a full-resolution RGB image $\tilde{x}$. Note that post-processing (e.g. median filtering [19]) or denoising [35] may also affect the image but are not detailed here. The watermark detector does not know which demosaicking algorithm and post-processing operations have been applied on the watermarked raw image. Nevertheless, we can approximate the effect of the demosaicking step with an expansion of the data with a matrix $M = [2\ 0; 0\ 2]$ which yields an image x_e twice the size in each dimension and interpolation with a low-pass filter $h_I = [1/4\ 1/2\ 1/4; 1/2\ 1\ 1/2; 1/4\ 1/2\ 1/4]$ resulting in an upsampled image $\tilde{x}$. Finally, we roughly model the impact of the post-processing and JPEG compression stage as an additive noise source n.

Demosaicking basically upsamples the CFA image data and interpolates missing pixels. A framework for blind watermark detection in noisy, interpolated images has been proposed by Giannoula et al. [10]. In [22], this framework has been successfully applied to demosaicked images, irrespective of a particular interpolation technique. Only three demosaicking implementations have been tested. In this work we extend the results to 22 different algorithms (see Table 1).

The actual implementation of the demosaicking, post- processing and compression stage is unknown. However, we can make assumptions on the interpolation and demosaicking step and utilize the interpolated structure of the demosaicked image for efficient watermark detection. In this work, we separately assess the impact of demosaicking on a watermark embedded in the blue color channel and the green color channel.

4. WATERMARK DETECTION FROM THE DEMOSAICKED IMAGE

The received RGB image data is denoted by $s = [R\ G\ B]$. The pixels of each color band $s^b \in \{R, G, B\}$ can be further separated into its noisy polyphase components s_i^b where $0 \leq i \leq 3$ refers to one of the four components [32] and b indicates the color band. The simplest detector uses just the low-resolution watermarked polyphase component s_0^b from the demosaicked image and employs a linear correlator where b selects the watermarked CFA color channel. However, for blind watermark detection the host signal has to be considered as noise interferes with the detection of the signal. Direct linear correlation (LC) detection gives poor results, so we ignore this option and seek to reduce the interference in the following. It is well known that detection performance can be significantly improved by applying a whitening filter (e.g. $h_{wf} = [1/4\ -2/4\ 1/4;\ -2/4\ 1\ -2/4; 1/4\ -2/4\ 1/4]$) on the received image and thus reducing correlation between neighboring pixels and host signal interference [8]. The detection statistic is given by

$$\rho_{pf} = \frac{1}{N}\sum_{\mathbf{m}}(h_{wf} * s_0)[\mathbf{m}]\cdot w[\mathbf{m}] \qquad (1)$$

where the $*$ operator represents convolution. We denote the resulting pre-filtering detector with PF-LC. We now look into reducing host signal interference by taking advantage of the demosaicked image structure.

4.1 Intra color band fusion

Relying on the assumptions about the image processing pipeline given in Section 3, we can adapt the watermark detection strategy proposed by Giannoula et al. [10] for interpolated, noisy images. While the watermark is embedded in the low-resolution raw data, watermark detection takes place using the high-resolution data of the demosaicked and usually compressed image. We aim at exploiting the additional watermark information being spread out due to interpolation.

Figure 3 illustrates the process for the blue color band, i.e. $s^b = B$. To simplify notation, we do not explicitly denote the color band. s_0 thus represents the low-resolution watermarked blue CFA data (remember that $c_w = B_0+\alpha\cdot w$), cor-

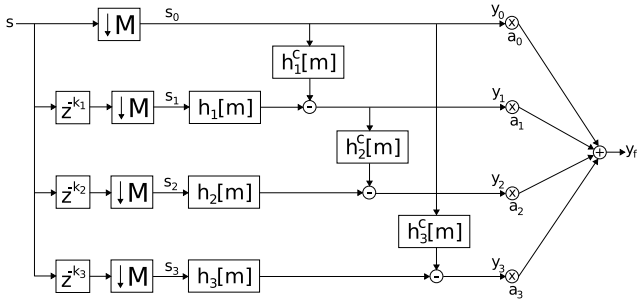

Figure 3: Polyphase fusion of the received image with four components

rupted by a noise component n_0, $y_0[\mathbf{m}] = s_0[\mathbf{m}] = c_w[\mathbf{m}] + n_0[\mathbf{m}]$. With the help of two linear filters for estimation and interference cancellation,

$$h_i[\mathbf{m}] = \beta \cdot h_I[\mathbf{m}] \text{ and } h_i^c[\mathbf{m}] = \beta \cdot h_I[\mathbf{m}] * h_I[\mathbf{m}] - \delta[\mathbf{m}], \quad (2)$$

respectively, further noisy estimates of c_w are computed, such that

$$y_i[\mathbf{m}] = c_w[\mathbf{m}] + n_i[\mathbf{m}] = h_i[\mathbf{m}] * s_i[\mathbf{m}] - h_i^c[\mathbf{m}] * s_0[\mathbf{m}]. \quad (3)$$

The components $s_i, i \neq 0$ correspond to the reconstructed pixels of the blue color band. The scaling factor β is adjusted such that $h_i^c[0] = 0$ for $1 \leq i \leq 3$, $\delta[\mathbf{m}]$ is the Kronecker delta. Finally, the components y_i are *fused* according to optimal weight factors $a_i \in [0, 1]$, $\sum_i a_i = 1$, depending on the estimated noise variance $\sigma_{n_i}^2$ of each component, $y_f[\mathbf{m}] = \sum_i a_i \cdot y_i[\mathbf{m}]$ where

$$(a_0, ..., a_3) = \left(\frac{1}{\sigma_{n_0}^2 \sum_i \frac{1}{\sigma_{n_i}^2}}, ..., \frac{1}{\sigma_{n_3}^2 \sum_i \frac{1}{\sigma_{n_i}^2}} \right). \quad (4)$$

Giannoula et al. [10] suggest to estimate the noise variance $\sigma_{n_i}^2$ by filtering the initial component samples s_0 and subtracting the result from s_i, i.e.

$$\hat{\sigma}_{n_i}^2 = \text{var}\left(s_i[\mathbf{m}] - h_I[\mathbf{m}] * s_0[\mathbf{m}] \right). \quad (5)$$

The *fused* image y_f is suitable for watermark detection.

So far we have assumed watermark embedding occurs in the blue CFA component (i.e. $c = B_0$). For the case where the watermark is embedded in one of the green CFA components (either $c = G_0$ or $c = G_1$), the polyphase component fusion approach described above is constrained to only three components as only half of the pixels in the green color band are interpolated.

4.2 Inter color band fusion

As discussed in Section 2, most demosaicking methods perform *sequential inter* band interpolation and therefore the watermark signal is carried over to the other color bands as well. Optimal fusion of polyphase components of one color band was discussed in the previous Section. In a similar way, we can try to exploit inter band correlation by fusing polyphase components from all color bands. Considering the case where the watermark is embedded in the blue CFA component, we note that the red and green pixels in the full-resolution image have to be reconstructed at those locations where the CFA data has a blue pixel. The original pixel values contribute to in the reconstruction and so the watermark signal is transfered to all bands.

In case of blue CFA component embedding, the *color fused* image is computed as

$$y_{cf}[\mathbf{m}] = a_0^R \cdot y_0^R[\mathbf{m}] + \sum_{i=0}^{3} a_i^B \cdot y_i^B[\mathbf{m}] + a_0^G \cdot y_0^G[\mathbf{m}] \quad (6)$$

where $y_0^B = s_0^B$, otherwise $y_i^b[\mathbf{m}] = h_i[\mathbf{m}] * s_i^b[\mathbf{m}] - h_i^c[\mathbf{m}] * s_0^B[\mathbf{m}]$. The fusion weights $a_i^b, \sum_{i,b} a_i^b = 1$ are determined as before depending on the individual components' estimated noise variances

$$\hat{\sigma}_{n_i^b}^2 = \text{var}\left(s_i^b[\mathbf{m}] - h_I[\mathbf{m}] * s_0^B[\mathbf{m}] \right). \quad (7)$$

When the watermark is embedded in the green CFA component, component fusion is confined to three *intra* and two *inter* components.

4.3 Detection problem

The watermark detection problem can be formulated as a binary hypothesis test on the received image:

$$\begin{aligned} &\mathcal{H}_0 : \text{test image contains no or other watermark} \\ &\mathcal{H}_1 : \text{test image is watermarked with } w. \end{aligned} \quad (8)$$

We apply the widely-used linear correlation (LC) detector on the *pre-filtered* (PF-LC), *fused* (F-LC), *color fused* (CF-LC) and *pre-filtered*, *color fused* image (PFCF-LC). The detection statistic of the F-LC detector can be expressed as

$$\rho_f = \frac{1}{N} \sum_{\mathbf{m}} y_f[\mathbf{m}] \cdot w[\mathbf{m}] \quad (9)$$

and follows a Gaussian distribution under both, $\mathcal{H}_0$ and $\mathcal{H}_1$. In order to decide on the valid hypothesis, ρ_f is compared against a suitable threshold T selected for a given probability of false-alarm (P_f) under the Neyman-Pearson criterion and the performance of the detector can be expressed in terms of probability of miss (P_m). See [10] for a detailed analysis of the *fused* detector.

Based on the *color fused* image, we can define the CF-LC detector with detection statistic

$$\rho_{cf} = \frac{1}{N} \sum_{\mathbf{m}} y_{cf}[\mathbf{m}] \cdot w[\mathbf{m}] \quad (10)$$

and the PFCF-LC detector operating on the *pre-filtered*, *color fused* image

$$\rho_{pfcf} = \frac{1}{N} \sum_{\mathbf{m}} (h_{wf} * y_{cf})[\mathbf{m}] \cdot w[\mathbf{m}]. \quad (11)$$

Similarly, the detection statistic of ρ_{pf}, ρ_{cf} and ρ_{pfcf} follow a Gaussian law under the assumption that the host signal is i.i.d. and the central limit theorem can be invoked. The parameters of the detection statistics can be estimated experimentally under $\mathcal{H}_0$ and $\mathcal{H}_1$ by performing numerous detection attempts, which we undertake in the next section.

5. RESULTS

For our experiments we choose 12 landscape color images from the Kodak Photo-CD set which are commonly used for demosaicking experiments. The images have been scanned from film. Artificial CFA image data is obtained by selecting pixels in the Bayer pattern and discarding the remaining pixels. Next, we generate two sets of watermarked

Figure 4: Test images from the Kodak Photo-CD set (768×512 pixels)

images by embedding 330 randomly generated bipolar, additive spread-spectrum watermarks in the blue component pixels ($c = B_0$), or, for the other set of watermarked images, in half of the green component pixels ($c = G_1$). The watermark embedding strength is set to $\alpha = 4$.

The full resolution RGB images are then restored by demosaicking the watermarked CFA image data with 22 different demosaicking methods, resulting in approximately 185 GB of data. We also include a set of images where the watermark is embedded directly in the full-resolution image for reference.

Watermark detection is performed on the full-resolution images. The Python source code of the watermark detectors is available at `http://www.wavelab.at/sources`. To simulate the impact of noise, the images are subjected to JPEG compression with the quality factor varying from 20 up to 100 with increments of 5. At the detector side, we crop the center 256×256 pixels from the received images. The cropping mitigates the problem of incorrect border handling of some of the demosaicking implementations and provides for a challenging detection scenario.

We compare the detection performance of the four watermark detection methods described in Section 4: the *fused* (F-LC) and *color fused* (CF-LC) detector, the linear-correlation detector on the high-pass pre-filtered image (PF-LC) and a LC detector operating on the high-pass pre-filtered *color fused* image (PFCF-LC). In total, 6272640 detection attempts have been performed to estimate the parameters of the detection statistic under $\mathcal{H}_0$ and $\mathcal{H}_1$ for all cases.

In Table 2 the percentage of detection errors (false positives and false negatives) is given for the watermark embedded in the blue CFA component and for the green CFA component for a probability of false-alarm (P_f) of 10^{-4}. The errors have been summed for each demosaicking scheme over all 12 images and 18 JPEG compression settings (including no compression). The lowest error rates have been marked in bold. We note that overall significantly more errors occur detecting the blue channel watermark versus the green channel watermark which is due to the different processing of the color channels during JPEG compression and demosaicking.

In case of the blue channel watermark (left column of Table 2), the error rates are very high for the images generated by *intra* only demosaicking schemes (and the reference schemes denoted by 'none' where the watermark is embedded in the full-resolution image): not surprisingly, PF-LC results in the best detection performance as no gain can be expected from taking other color bands into account. For all other demosaicking approaches, the lowest error rates are obtained by PFCF-LC, yet the advantage over PF-LC is very slim. Pre-filtering leads to a significant reduction in error rate. The watermark embedded in green CFA component spreads out during demosaicking as discussed in Section 2 and is therefore best picked up by the CF-LC and PFCF-LC detector (see right column of Table 2).

We now take a closer look at the probability of missing the watermark for select images and compression rates for a given false-alarm rate $P_f = 10^{-9}$. Table 3 compares the detection results obtained by the four detectors for the blue channel watermark using the *Girl* image (left column) after JPEG compression with quality factor $Q = 80$ and the *Archipel* image (right column) with $Q = 70$. It is evident that pre-filtering significantly improves the detection performance. Detection on the *color fused* image further enhances the results. In Table 4 we present the detection results for the green channel watermarks using the *Motorbike* and *Valley* image after JPEG compression with $Q = 70$ and $Q = 30$, respectively. We observe that the CF-LC detectors performs best for most instances. Interestingly, also for the *intra* only demosaicked images, the *color fused* detector is superior and even outperforms detectors relying on pre-filtering. We speculate that the host interference is reduced by incorporating all color channels in the fusion process.

From the results we see that the demosaicking and compression step have an impact on the watermark embedded in the raw image data. A watermark detector can benefit from the resulting interpolated image structure and correlation between color channels. The particular demosaicking method used is of secondary importance; neither spatial nor frequency domain demosaicking hinders watermark detection. AP demosaicking is troublesome for the blue channel watermark, while ECI demosaicking gives worst results for the green channel watermark detection. The two joint denoising and demosaicking methods, JDD and LPA, do not show severe adverse effects related to watermark detection although their noise estimator conditioned on the characteristics of the additive, spread-spectrum watermark.

In this work we have focused on simple additive, spread-spectrum embedding and linear correlation detection after pre-processing the received image in order to study the effect of demosaicking. Perceptual shaping of the watermark and more accurate modelling of the host signal (e.g. assuming a Cauchy model [30]) might improve the detection performance.

6. CONCLUSION

Digital watermarking has to be applied close to the image acquisition stage in order to protect the copyright of both, the raw and compressed image. We have investigated embedding of a simple additive, spread-spectrum watermark in the CFA sensor data which can be applied even under very constrained processing resources. We have proposed watermark detection schemes that allow to exploit the intra and inter color band correlation in the full-resolution demosaicked image which leads to improved detection performance compared to using the low-resolution component only. The impact of different demosaicking methods on watermark detection performance was evaluated separately for embedding in the blue and green CFA component of the raw image.

Table 2: Percentage of detection errors (false positive and false negative), $P_f = 10^{-4}$

Method	Blue Channel Watermark				Green Channel Watermark			
	F-LC	CF-LC	PF-LC	PFCF-LC	F-LC	CF-LC	PF-LC	PFCF-LC
bilinear	38.92	39.97	**26.50**	26.60	5.63	3.24	1.79	**1.72**
EDI	32.57	37.77	**31.68**	32.21	1.13	**0.82**	0.91	0.91
none	44.07	44.37	**42.25**	42.27	15.39	9.97	9.34	**8.98**
ACPI	16.21	15.78	7.09	**6.95**	1.75	**0.52**	0.85	0.76
DAF	20.90	17.21	3.57	**3.38**	4.45	2.12	1.38	**1.25**
DF	11.38	9.79	3.84	**3.64**	2.47	**0.83**	1.40	1.24
DLMMSE	13.24	11.34	4.08	**3.95**	2.09	**0.62**	1.11	0.98
ECI	13.26	12.6	5.55	**5.33**	1.57	**0.52**	0.84	0.77
JDD	16.67	14.62	5.29	**5.16**	4.61	**2.04**	2.66	2.45
SA	7.79	5.20	2.54	**2.39**	3.64	**1.53**	1.82	1.66
LPA	11.93	10.08	3.97	**3.83**	2.47	**0.79**	1.36	1.21
NCRM	10.41	11.54	4.23	**3.97**	5.18	2.48	1.21	**1.11**
NM	15.93	15.13	6.23	**6.05**	1.56	**0.46**	0.80	0.71
PCD	19.17	17.42	5.51	**5.33**	3.08	1.03	0.98	**0.87**
SSC	7.00	6.93	3.35	**3.16**	3.71	1.59	1.46	**1.33**
TSD	9.39	7.21	3.01	**2.84**	2.89	**1.11**	1.55	1.42
VCD	24.47	20.31	3.62	**3.49**	7.23	3.93	1.35	**1.25**
AF	10.78	8.88	3.71	**3.50**	2.77	**1.05**	1.53	1.40
AHD	14.83	13.78	5.97	**5.79**	1.09	**0.27**	0.60	0.53
AP	38.38	39.77	24.92	**24.65**	11.72	6.17	2.61	**2.49**
FD	8.60	6.39	2.72	**2.54**	0.54	**0.25**	0.43	0.35
FHVS	9.81	7.33	2.80	**2.55**	2.75	**1.10**	1.73	1.63

Table 3: Probability of missing the blue channel watermark, $P_f = 10^{-9}$

Method	Girl (Q=80)				Archipel (Q=70)			
	F-LC	CF-LC	PF-LC	PFCF-LC	F-LC	CF-LC	PF-LC	PFCF-LC
bilinear	1.00	1.00	$\mathbf{9.1\,e^{-1}}$	$9.2\,e^{-1}$	1.00	1.00	$\mathbf{9.6\,e^{-1}}$	$\mathbf{9.6\,e^{-1}}$
EDI	1.00	1.00	$\mathbf{9.9\,e^{-1}}$	$\mathbf{9.9\,e^{-1}}$	1.00	1.00	1.00	1.00
none	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ACPI	$9.8\,e^{-1}$	$9.2\,e^{-1}$	$2.4\,e^{-5}$	$\mathbf{9.6\,e^{-6}}$	$9.8\,e^{-1}$	$1.3\,e^{-1}$	$3.7\,e^{-4}$	$\mathbf{1.6\,e^{-4}}$
DAF	1.00	1.00	$1.5\,e^{-18}$	$\mathbf{2.5\,e^{-20}}$	1.00	1.00	$1.0\,e^{-11}$	$\mathbf{3.3\,e^{-13}}$
DF	$6.0\,e^{-1}$	$1.9\,e^{-1}$	$8.9\,e^{-19}$	$\mathbf{4.4\,e^{-21}}$	$6.4\,e^{-1}$	$3.6\,e^{-6}$	$1.1\,e^{-11}$	$\mathbf{2.3\,e^{-13}}$
DLMMSE	$8.3\,e^{-1}$	$4.8\,e^{-1}$	$3.3\,e^{-14}$	$\mathbf{1.2\,e^{-15}}$	$9.2\,e^{-1}$	$1.5\,e^{-3}$	$3.9\,e^{-10}$	$\mathbf{2.0\,e^{-11}}$
ECI	$8.4\,e^{-1}$	$5.1\,e^{-1}$	$4.7\,e^{-10}$	$\mathbf{5.5\,e^{-11}}$	$5.9\,e^{-1}$	$3.9\,e^{-4}$	$3.0\,e^{-7}$	$\mathbf{5.3\,e^{-8}}$
JDD	1.00	$9.9\,e^{-1}$	$4.9\,e^{-4}$	$\mathbf{3.0\,e^{-4}}$	1.00	$8.2\,e^{-1}$	$2.4\,e^{-2}$	$\mathbf{1.5\,e^{-2}}$
LAP	$8.2\,e^{-1}$	$4.8\,e^{-1}$	$2.7\,e^{-15}$	$\mathbf{1.0\,e^{-16}}$	$7.7\,e^{-1}$	$8.1\,e^{-5}$	$7.3\,e^{-11}$	$\mathbf{2.7\,e^{-12}}$
NCRM	$3.2\,e^{-1}$	$8.9\,e^{-1}$	$6.6\,e^{-12}$	$\mathbf{5.4\,e^{-13}}$	$9.7\,e^{-1}$	$7.4\,e^{-2}$	$3.7\,e^{-10}$	$\mathbf{3.1\,e^{-11}}$
NM	$9.7\,e^{-1}$	$9.0\,e^{-1}$	$2.9\,e^{-7}$	$\mathbf{6.9\,e^{-8}}$	$9.7\,e^{-1}$	$8.7\,e^{-2}$	$3.2\,e^{-5}$	$\mathbf{9.0\,e^{-6}}$
PCD	1.00	1.00	$3.8\,e^{-10}$	$\mathbf{5.7\,e^{-11}}$	$6.4\,e^{-1}$	$7.3\,e^{-1}$	$1.6\,e^{-6}$	$\mathbf{2.4\,e^{-7}}$
SA	$1.4\,e^{-1}$	$4.2\,e^{-3}$	$1.8\,e^{-28}$	$\mathbf{8.5\,e^{-32}}$	$2.3\,e^{-1}$	$6.9\,e^{-14}$	$2.3\,e^{-18}$	$\mathbf{5.4\,e^{-21}}$
SSC	$1.1\,e^{-2}$	$4.4\,e^{-1}$	$1.2\,e^{-20}$	$\mathbf{5.3\,e^{-23}}$	$1.6\,e^{-2}$	$4.7\,e^{-8}$	$1.5\,e^{-13}$	$\mathbf{2.1\,e^{-15}}$
TSD	$3.4\,e^{-1}$	$4.4\,e^{-2}$	$1.5\,e^{-23}$	$\mathbf{2.3\,e^{-26}}$	$4.2\,e^{-1}$	$1.5\,e^{-9}$	$1.2\,e^{-15}$	$\mathbf{6.5\,e^{-18}}$
VCD	1.00	1.00	$3.8\,e^{-18}$	$\mathbf{2.8\,e^{-19}}$	1.00	1.00	$9.8\,e^{-13}$	$\mathbf{5.5\,e^{-14}}$
AF	$6.4\,e^{-1}$	$2.4\,e^{-1}$	$1.2\,e^{-16}$	$\mathbf{1.5\,e^{-18}}$	$6.8\,e^{-1}$	$6.7\,e^{-6}$	$3.8\,e^{-11}$	$\mathbf{9.7\,e^{-13}}$
AHD	$9.4\,e^{-1}$	$7.3\,e^{-1}$	$1.4\,e^{-8}$	$\mathbf{1.3\,e^{-9}}$	$9.5\,e^{-1}$	$2.1\,e^{-2}$	$7.0\,e^{-6}$	$\mathbf{1.4\,e^{-6}}$
AP	1.00	1.00	$7.2\,e^{-1}$	$\mathbf{6.5\,e^{-1}}$	1.00	1.00	$\mathbf{9.6\,e^{-1}}$	$\mathbf{9.6\,e^{-1}}$
FD	$1.7\,e^{-1}$	$9.3\,e^{-3}$	$1.9\,e^{-27}$	$\mathbf{4.5\,e^{-31}}$	$2.3\,e^{-1}$	$5.4\,e^{-12}$	$7.0\,e^{-18}$	$\mathbf{1.1\,e^{-20}}$
FHVS	$6.1\,e^{-1}$	$2.4\,e^{-1}$	$2.8\,e^{-29}$	$\mathbf{4.4\,e^{-32}}$	$4.9\,e^{-1}$	$8.5\,e^{-8}$	$8.4\,e^{-17}$	$\mathbf{1.7\,e^{-19}}$

Table 4: Probability of missing the green channel watermark, $P_f = 10^{-9}$

Method	Motorbike (Q=70)				Valley (Q=30)			
	F-LC	CF-LC	PF-LC	PFCF-LC	F-LC	CF-LC	PF-LC	PFCF-LC
bilinear	$8.8\,e^{-8}$	$\mathbf{2.0\,e^{-11}}$	$5.7\,e^{-8}$	$3.7\,e^{-8}$	$3.0\,e^{-14}$	$\mathbf{9.4\,e^{-21}}$	$1.7\,e^{-10}$	$4.7\,e^{-12}$
EDI	$8.0\,e^{-10}$	$\mathbf{5.6\,e^{-15}}$	$3.3\,e^{-10}$	$1.8\,e^{-10}$	$1.3\,e^{-19}$	$\mathbf{1.5\,e^{-28}}$	$4.0\,e^{-14}$	$2.1\,e^{-16}$
none	$2.3\,e^{-5}$	$\mathbf{4.2\,e^{-8}}$	$1.7\,e^{-5}$	$1.4\,e^{-5}$	$2.7\,e^{-11}$	$\mathbf{4.3\,e^{-17}}$	$3.0\,e^{-8}$	$1.1\,e^{-9}$
ACPI	$2.9\,e^{-1}$	$1.2\,e^{-2}$	$\mathbf{2.0\,e^{-4}}$	$2.1\,e^{-4}$	$1.2\,e^{-2}$	$\mathbf{2.8\,e^{-6}}$	$9.4\,e^{-5}$	$4.7\,e^{-5}$
DAF	$6.4\,e^{-5}$	$\mathbf{7.1\,e^{-9}}$	$1.0\,e^{-7}$	$7.1\,e^{-8}$	$6.1\,e^{-6}$	$\mathbf{6.5\,e^{-14}}$	$5.7\,e^{-10}$	$2.9\,e^{-11}$
DF	$4.1\,e^{-6}$	$\mathbf{1.3\,e^{-9}}$	$3.5\,e^{-7}$	$2.4\,e^{-7}$	$3.0\,e^{-8}$	$\mathbf{1.9\,e^{-16}}$	$8.3\,e^{-10}$	$2.1\,e^{-11}$
DLMMSE	$3.8\,e^{-6}$	$\mathbf{5.8\,e^{-9}}$	$2.8\,e^{-7}$	$2.4\,e^{-7}$	$8.0\,e^{-11}$	$\mathbf{1.1\,e^{-15}}$	$3.9\,e^{-9}$	$1.6\,e^{-10}$
ECI	$6.7\,e^{-1}$	$\mathbf{3.8\,e^{-1}}$	$4.8\,e^{-1}$	$4.9\,e^{-1}$	$2.5\,e^{-1}$	$\mathbf{5.6\,e^{-2}}$	$2.2\,e^{-1}$	$1.4\,e^{-1}$
JDD	$1.5\,e^{-5}$	$\mathbf{3.8\,e^{-9}}$	$1.1\,e^{-7}$	$7.3\,e^{-8}$	$1.2\,e^{-8}$	$\mathbf{1.5\,e^{-15}}$	$8.5\,e^{-11}$	$1.8\,e^{-12}$
LPA	$6.4\,e^{-7}$	$\mathbf{4.3\,e^{-10}}$	$1.6\,e^{-7}$	$1.4\,e^{-7}$	$5.5\,e^{-9}$	$\mathbf{4.6\,e^{-12}}$	$1.0\,e^{-7}$	$1.1\,e^{-8}$
NCRM	$7.0\,e^{-5}$	$\mathbf{2.3\,e^{-8}}$	$2.1\,e^{-7}$	$1.7\,e^{-7}$	$3.6\,e^{-7}$	$\mathbf{5.4\,e^{-12}}$	$1.8\,e^{-9}$	$6.2\,e^{-11}$
NM	$7.0\,e^{-6}$	$\mathbf{1.8\,e^{-8}}$	$9.0\,e^{-7}$	$7.5\,e^{-7}$	$1.0\,e^{-11}$	$\mathbf{3.9\,e^{-17}}$	$2.7\,e^{-9}$	$1.0\,e^{-10}$
PCD	$8.8\,e^{-8}$	$\mathbf{2.5\,e^{-11}}$	$2.5\,e^{-8}$	$1.4\,e^{-8}$	$3.2\,e^{-16}$	$\mathbf{1.5\,e^{-22}}$	$6.5\,e^{-12}$	$1.4\,e^{-13}$
SA	$\mathbf{1.2\,e^{-7}}$	$3.2\,e^{-7}$	$4.3\,e^{-5}$	$3.6\,e^{-5}$	$1.0\,e^{-14}$	$\mathbf{6.2\,e^{-20}}$	$5.9\,e^{-14}$	$9.2\,e^{-14}$
SSC	$1.4\,e^{-2}$	$\mathbf{6.4\,e^{-5}}$	$9.9\,e^{-5}$	$\mathbf{6.4\,e^{-5}}$	$2.7\,e^{-4}$	$\mathbf{3.6\,e^{-10}}$	$4.2\,e^{-7}$	$2.4\,e^{-7}$
TSD	$6.0\,e^{-7}$	$\mathbf{2.9\,e^{-10}}$	$2.5\,e^{-7}$	$1.9\,e^{-7}$	$4.5\,e^{-12}$	$\mathbf{1.5\,e^{-18}}$	$8.2\,e^{-9}$	$2.0\,e^{-10}$
VCD	$3.8\,e^{-5}$	$\mathbf{2.6\,e^{-8}}$	$3.9\,e^{-7}$	$3.3\,e^{-7}$	$3.2\,e^{-9}$	$\mathbf{3.5\,e^{-12}}$	$2.1\,e^{-9}$	$6.2\,e^{-11}$
AF	$1.2\,e^{-5}$	$\mathbf{2.4\,e^{-8}}$	$4.9\,e^{-7}$	$4.0\,e^{-7}$	$4.2\,e^{-9}$	$\mathbf{4.0\,e^{-14}}$	$1.9\,e^{-8}$	$9.5\,e^{-10}$
AHD	$8.5\,e^{-8}$	$\mathbf{9.1\,e^{-12}}$	$9.3\,e^{-9}$	$5.3\,e^{-9}$	$8.7\,e^{-16}$	$\mathbf{9.1\,e^{-23}}$	$3.9\,e^{-13}$	$4.9\,e^{-15}$
AP	$2.0\,e^{-9}$	$\mathbf{6.3\,e^{-11}}$	$1.3\,e^{-6}$	$3.3\,e^{-7}$	$1.0\,e^{-20}$	$\mathbf{1.2\,e^{-26}}$	$1.3\,e^{-11}$	$3.7\,e^{-14}$
FD	$6.3\,e^{-4}$	$\mathbf{1.2\,e^{-8}}$	$1.1\,e^{-7}$	$8.8\,e^{-8}$	$8.7\,e^{-6}$	$\mathbf{1.1\,e^{-14}}$	$4.1\,e^{-9}$	$3.0\,e^{-10}$
FHVS	$5.0\,e^{-5}$	$\mathbf{3.3\,e^{-7}}$	$6.7\,e^{-6}$	$7.4\,e^{-6}$	$2.0\,e^{-8}$	$\mathbf{6.4\,e^{-14}}$	$2.0\,e^{-8}$	$2.1\,e^{-9}$

Acknowledgments

Supported by Austrian Science Fund project FWF-P19159-N13. Thanks to Colin M. L. Burnett for the graphics used in Figure 1.

7. REFERENCES

[1] D. Alleysson, S. Susstrunk, and J. Herault. Linear demosaicing inspired by the human visual system. *IEEE Transactions on Image Processing*, 14(4):439–449, Apr. 2005.

[2] S. Battiato, M. Guarnera, G. Messina, and V. Tomaselli. Recent patents on color demosaicing. *Recent Patents on Computer Science*, 1(3):194–207, Nov. 2008.

[3] P. Blythe and J. Fridrich. Secure digital camera. In *Digital Forensic Research Workshop*, Baltimore, MD, USA, Aug. 2004.

[4] L. Chang and Y.-P. Tan. Effective use of spatial and spectral correlations for color filter array demosaicking. *IEEE Transactions on Consumer Electronics*, 50(1):355–365, Feb. 2004.

[5] M. Chen, J. Fridrich, M. Goljan, and J. Lukas. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Security and Forensics*, 3(1):74–90, Mar. 2008.

[6] K.-H. Chung and Y.-H. Chan. Color demosaicing using variance of color differences. *IEEE Transactions on Image Processing*, 15(10):2944–2955, Oct. 2006.

[7] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007.

[8] G. Depovere, T. Kalker, and J.-P. Linnartz. Improved watermark detection reliability using filtering before correlation. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '98*, volume 1, pages 430–434, Chicago, IL, USA, Oct. 1998.

[9] E. Dubois. Frequency domain methods for demosaicking of bayer-sampled color images. *IEEE Signal Processing Letters*, 12(12):847–850, Dec. 2005.

[10] A. Giannoula, N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis. Watermark detection for noisy interpolated images. *IEEE Transactions on Circuits and Systems*, 53(5):359–363, May 2006.

[11] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013, Sept. 2002.

[12] J. Hamilton and J. Adams. Adaptive color plane interpolation in single sensor color electronic camera, May 1997. US Patent 5,629,734.

[13] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing*, 14(3):360–369, Mar. 2005.

[14] K. Hirakawa and T. W. Parks. Joint demosaicing and denoising. *IEEE Transactions on Image Processing*, 15(8):2146–2157, Aug. 2006.

[15] X. Li. Demosaicing by successive approximation. *IEEE Transactions on Image Processing*, 14(3):370–379, Mar. 2005.

[16] X. Li, B. Gunturk, and L. Zhang. Image demosaicing: A systematic survey. In *Proceedings of SPIE, Visual Communications and Image Processing, VCIP '08*, volume 6822, pages 68221J–68221J–15, San Jose, CA, USA, Jan. 2008. SPIE.

[17] X. Li and M. Orchard. New edge-directed

interpolation. *IEEE Transactions on Image Processing*, 10(10):1521–1527, Oct. 2001.

[18] N. Lian, V. Zagorodnov, and Y.-P. Tan. Adaptive filtering for color filter array demosaicking. *IEEE Transactions on Image Processing*, 16(10):2515–2525, Oct. 2007.

[19] W. Lu and Y.-P. Tan. Color filter array demosaicing: new method and performance measures. *IEEE Transactions on Image Processing*, 12(10):1194–1210, Oct. 2003.

[20] R. Lukac and K. N. Plataniotis. Normalized color-ratio modeling for CFA interpolation. *IEEE Transactions on Consumer Electronics*, 50(2):737–745, May 2004.

[21] R. Lukac and K. N. Plataniotis. Data-adaptive filters for demosaicking: A framework. *IEEE Transactions on Consumer Electronics*, 51(2):560–570, May 2005.

[22] P. Meerwald and A. Uhl. Watermarking of raw digital images in camera firmware: embedding and detection. In *Advances in Image and Video Technology: Proceedings of the 3rd Pacific-Rim Symposium on Image and Video Technology, PSIVT '09*, volume 5414 of *Lecture Notes in Computer Science*, pages 340–348, Tokyo, Japan, Jan. 2009. Springer.

[23] D. Menon, S. Andriani, and G. Calvagno. Demosaicing with directional filtering and a posteriori decision. *IEEE Transactions on Image Processing*, 16(1):132–141, Jan. 2007.

[24] S. P. Mohanty, E. Kougianos, and N. Ranganathan. VLSI architecture and chip for combined invisible robust and fragile watermarking. *IET Computers & Digital Techniques*, 1(5):600–611, June 2007.

[25] G. R. Nelson, G. A. Julien, and O. Yadid-Pecht. CMOS image sensor with watermarking capabilities. In *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS '05*, volume 5, pages 5326–5329. IEEE, May 2005.

[26] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian. Spatially adaptive color filter array interpolation for noiseless and noisy data. *International Journal of Imaging Systems and Technology, Special Issue on Applied Color Image Processing*, 17(3):105–122, Oct. 2007.

[27] S.-C. Pei and I.-K. Tam. Effective color interpolation in CCD color filter array using signal correlation. *IEEE Transactions on Circuit Systems and Video Technology*, 13(6):503–513, June 2003.

[28] A. C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, Oct. 2005.

[29] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew. Color image processing pipeline: A general survey of digital still camera processing. *IEEE Signal Processing Magazine*, 22(1):34–43, Jan. 2005.

[30] E. Sayrol, J. Vidal, S. Cabanillas, and S. Santamaría. Optimum watermark detection in color images. In *Proceedings of the IEEE International Conference on Image Processing, ICIP '99*, volume 2, pages 231–235, Kobe, Japan, Oct. 1999.

[31] C.-Y. Su. Low-complexity hybrid demosaicing for color filter arrays. *Journal of the Chinese Institute of Engineers*, 31(1):173–179, Jan. 2008.

[32] P. P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial. *Proceedings of the IEEE*, 78(1):56–93, Jan. 1990.

[33] X. Wu and N. Zhang. Primary-consistent soft-decision color demosaicking for digital cameras. *IEEE Transactions on Image Processing*, 13(9):1263–1274, Sept. 2004.

[34] L. Zhang and X. Wu. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Transactions on Image Processing*, 14(12):2167–2178, Dec. 2005.

[35] L. Zhang, X. Wu, and D. Zhang. Color reproduction from noisy CFA data of single sensor digital cameras. *IEEE Transactions on Image Processing*, 16(9):2184–2197, Sept. 2007.