

Synthesis of Color Filter Array Pattern in Digital Images

Matthias Kirchner and Rainer Böhme

Technische Universität Dresden, Department of Computer Science, 01062 Dresden, Germany

ABSTRACT

We propose a method to synthetically create or restore typical color filter array (CFA) pattern in digital images. This can be useful, inter alia, to conceal traces of manipulation from forensic techniques that analyze the CFA structure of images. For continuous signals, our solution maintains optimal image quality, using a quadratic cost function; and it can be computed efficiently. Our general approach allows to derive even more efficient approximate solutions that achieve linear complexity in the number of pixels. The effectiveness of the CFA synthesis as tamper-hiding technique and its superior image quality is backed with experimental evidence on large image sets and against state-of-the-art forensic techniques. This exposition is confined to the most relevant ‘Bayer’-grid, but the method can be generalized to other layouts as well.

1. INTRODUCTION

Tamper detection in digital images is a standard problem of digital image forensics,^{1,2} and the arsenal of available methods to unveil image manipulation has grown rapidly over the past couple of years. One option to check the integrity of color images is to test the existence and completeness of characteristic local correlation pattern that originates from the color filter array (CFA) interpolation in typical image acquisition devices.³⁻⁵

Advances in and increasing relevance of image forensics have also raised questions about the reliability of forensic results, and stirred research into counter-technologies.^{6,7} So-called “tamper-hiding” techniques try to conceal traces of manipulation, i.e., mislead tamper detection methods in a way that they produce false negatives.⁸ This paper proposes a method to synthesize CFA pattern in arbitrary, possibly manipulated, color images while minimizing the distortion due to such a ‘restoration’ of characteristic CFA pattern.

We shall proceed as follows: Sect. 3 recalls the basics of CFA interpolation and its detection. Sect. 4 formulates the minimization problem in CFA synthesis. As finding the general solution to this problem requires about $\mathcal{O}(m^3n^3)$ operations, $m \times n$ being the image size in pixels, we also show how more efficient solutions can be found by exploiting the special structure of the underlying least squares problem. In the limit case, that is ignoring boundary conditions, our method can find approximate solutions in $\mathcal{O}(mn)$ steps. We present experimental results on the effectiveness of our method in Sect. 5 and conclude in Sect. 6.

2. NOTATION

Throughout this paper we denote matrices and vectors with uppercase and lowercase boldface symbols, respectively. A vector \mathbf{x} holds elements x_k , $k \in \{1, 2, \dots\}$, and likewise an $m \times n$ matrix \mathbf{A} consists of elements $A_{i,j}$, $(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\}$. When necessary we specify the dimension of a matrix explicitly as $\mathbf{A}^{(m \times n)}$. Matrix coordinates $\mathbf{c} = (i, j)$ can be mapped to vector indices by a family of functions $\text{vec}(\cdot, m)$, where m is the number of rows of the matrix (column-major order),

$$\text{vec}(\mathbf{c}, m) = i + m(j - 1). \quad (1)$$

The inverse mapping is defined as

$$\text{vec}^{-1}(k, m) = (\text{row}(k, m), \text{col}(k, m)) \quad \text{with } \text{row}(k, m) = k - m \left\lfloor \frac{k-1}{m} \right\rfloor, \text{col}(k, m) = \left\lfloor \frac{k-1}{m} \right\rfloor + 1. \quad (2)$$

To emphasize a special (block) structure of certain matrices, we write $\mathcal{A}^{(m \times n)}$ for a block matrix built of $m \times n$ blocks $\mathbf{A}_{i,j}$, which all share the same dimension. To simplify some matrix algebra, we make the following conventions, $\mathbf{A}^\times = \mathbf{A}'\mathbf{A}$ and $\mathbf{A}^+ = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'$ (the ‘pseudo-inverse’ of \mathbf{A}).

Further author information: {rainer.boehme,matthias.kirchner}@tu-dresden.de, Telephone: +49 351 463 38247

3. DETECTION OF CFA PATTERN

Most digital cameras capture color images with a single sensor and an array of color filters. As a result, only about one third of all pixels in an RGB image contain genuine information from a sensor element. The remaining pixels are interpolated. Although a variety of filter layouts has been proposed in the literature, our analysis in this paper is restricted to the most frequently used Bayer layout⁹ (cf. Fig. 6 of Ref. 9).

In Ref. 3, Popescu and Farid have proposed a forensic tool which can detect traces of CFA interpolation in color images in form of local correlation between pixels. They assume a mixture model as image generating process, according to which each pixel is generated from one of two processes:

1. Pixels can be values of a linear function \mathcal{M}_1 of pixels in the spatial neighbourhood, weighted with a vector of linear weights $\boldsymbol{\alpha}$, or
2. realisations of a random function \mathcal{M}_2 , which is assumed to be a uniform distribution.

The corresponding mixture equation for a single pixel is given as follows,

$$y_i = p_i \mathcal{M}_1(\mathbf{y}, \boldsymbol{\alpha}, i) + (1 - p_i) \mathcal{M}_2 + \varepsilon_i , \quad (3)$$

where i is an index to the pixel in vector \mathbf{y} , which holds the vectorized intensity values of the two-dimensional image lattice. $p_i \in [0, 1]$ is the probability that the i -th pixel is generated by process \mathcal{M}_1 . Vector \mathbf{p} is called ‘ p -map’ and can be estimated jointly with the linear weights $\boldsymbol{\alpha}$ from a given image using the iterative expectation-maximization algorithm,¹⁰ a standard technique for estimating mixture models. The systematic alternation of linear dependent and idiosyncratic pixels in CFA-interpolated color images causes a high frequency periodic pattern in the p -map. This feature is most visible after a Fourier transform to the frequency domain, where distinct characteristic peaks can be interpreted as indications of intact CFA pattern by a human forensic investigator.

Examples and performance evaluations of this detector for never-compressed and JPEG images can be found in Ref. 3. This type of CFA detector can also benefit from our recently proposed method¹¹ to substantially reduce the computational effort of detectors based on linear predictor residue (to which the above-described method can be transformed with simple calculus).

Gallahger and Chen’s recent publication⁵ puts forward an alternative detection method. Nevertheless we stick to Ref. 3 as reference detector in our research, since the alternative detector is based on similar ideas—the linear predictor is constrained to form a special case of the above described more general method—and is deemed equivalent for images that are large enough to estimate the predictor coefficients reliably.

4. SYNTHESIS OF CFA PATTERN

Synthesis of CFA pattern boils down to restoring the specific linear dependencies between spatially related pixels according to the emulated CFA layout. An obvious solution to achieve this goal is a *naive re-interpolation*, i.e., the subsampling of a $m \times n \times 3$ full color image to a $m \times n$ intensity image followed by a CFA re-interpolation of this ‘genuine’ image. Naive re-interpolation, however, would overwrite about two thirds of all pixels in a color image with new (interpolated) values and thus goes along with information loss comparable to linear kernel smoothing.¹² Our objective in this paper is to restore the CFA pattern in a minimal invasive way. That is to minimize the distortion between the input image and the output image, which contains the desired pattern. A quadratic cost function, such as PSNR, serves as distortion measure.

4.1. General approach

For our synthesis approach, we assume that images with CFA interpolation artefacts emerge from the following linear equation:

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{x} , \quad (4)$$

where \mathbf{x} is a vectorized 2D intensity lattice as captured by the sensor, \mathbf{H} is the matrix of CFA interpolation weights, which depends on the interpolation method and the color channel, and $\hat{\mathbf{y}}$ is a vectorized 2D lattice of a single color channel. Even though the linearity assumption might not hold for (some regions of) images produced by today’s digital cameras, it has been successfully applied to different CFA related forensic approaches^{3, 4, 13, 14} and is therefore believed to form a valid starting point. Analogously to Eq. (4), an image \mathbf{y} without or with incomplete CFA artefacts can be modelled as emerging from a similar process plus an additive residual signal ϵ , which represents the component of \mathbf{y} that is ‘incompatible’ with the CFA interpolation,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \epsilon . \quad (5)$$

Minimal distortion synthesis of a CFA pattern compatible with \mathbf{H} corresponds to finding a possible sensor signal \mathbf{x} such that $\|\epsilon\| = \|\mathbf{y} - \hat{\mathbf{y}}\| \rightarrow \min$, where $\|\cdot\|$ denotes the L_2 -norm of a signal. This is an ordinary least squares problem with its well-known solution

$$\mathbf{x} = (\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'\mathbf{y} = \mathbf{H}^+\mathbf{y} . \quad (6)$$

For some classes of \mathbf{H} , both \mathbf{x} and \mathbf{H} have to be decimated so that no indeterminable equations (all terms zero) remain in the system. This is equivalent to removing all samples that do not affect the color channel of interest.

One possible interpretation of Eq. (6) is that of a linear pre-filtering of the signal \mathbf{y} before performing the re-interpolation onto the Bayer grid from the subsampled ‘genuine’ intensity lattice,

$$\hat{\mathbf{y}} = \mathbf{H}(\mathbf{H}^+\mathbf{y}) . \quad (7)$$

In this respect, it is similar to the concept of low-pass filtering before downsampling a signal. Our approach to minimal distortion synthesis then aims at finding the optimal pre-filter coefficients in a least squares sense.

4.2. Efficiency improvements exploiting the structure of the least squares problem

From an analytical point of view, Eq. (6) solves our problem— $\hat{\mathbf{y}}$ can be obtained from Eq. (4) given \mathbf{x} and \mathbf{H} —and we are done. However, in practice, a straight-forward implementation of the least squares problem has to cope with two limiting factors and is very likely become intractable even for moderate image dimensions. First, the complexity of matrix inversion and multiplication grows cubic with the number of pixels. Second, working with the huge matrices generally requires a vast amount of memory.

However, it is easy to see that \mathbf{H} is typically sparse and has a regular structure. In the following we will discuss ways to exploit the special properties of \mathbf{H} to find more efficient solutions to the general minimization problem. We make the simplifying (but not limiting) assumption of bi-linear interpolation kernels

$$\mathbf{R} = \mathbf{B} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix} \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 1 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix} , \quad (8)$$

where \mathbf{R}, \mathbf{G} and \mathbf{B} denote the filter masks for the red, green and blue channel, respectively. We deal with each color channel separately and present explicit solutions to Eq. (6). By making use of efficient inversion algorithms, exact results can be obtained with reasonable complexity. Finally, we will present approximate solutions of complexity $\mathcal{O}(mn)$ which estimate \mathbf{x} by applying a linear filter to the input image \mathbf{y} .

4.2.1. Red channel

Let $\mathbf{y} = (y_1, y_2, \dots, y_{mn})$ denote the vectorized $m \times n$ 2D intensity lattice of the red color channel. Without loss of generality, assume that $m = 2k - 1$ and $n = 2l - 1$, with k, l integer, so that the Bayer grid layout defines $\mathbf{y}_R = (y_j \mid j = 2i + \lfloor \frac{i-1}{k} \rfloor(m-1))$, $i = 1, 2, \dots, kl$, as ‘genuine’, i.e., non-interpolated, samples. In the following

example ($m = n = 3$), the original samples that form vector \mathbf{x} are annotated with an asterisk superscript,

$$\begin{matrix} y_1^* & y_4 & y_7^* \\ y_2 & y_5 & y_8 \\ y_3^* & y_6 & y_9^* \end{matrix} \quad \text{and the } mn \times kl \text{ matrix } \mathbf{H} \text{ is given by } \quad \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For a generic notation of \mathbf{H} , we define two $(2\nu - 1) \times \nu$ matrices of type \mathbf{O} and \mathbf{E} with elements

$$O_{i,j} = \begin{cases} 1 & \text{if } i = 2j - 1 \\ 0 & \text{else,} \end{cases} \quad \text{and} \quad E_{i,j} = \begin{cases} 1 & \text{if } i \in \{2j, 2j - 2\} \\ 0 & \text{else,} \end{cases} \quad (9)$$

respectively. We further denote a matrix of type \mathbf{H}_α , $\alpha \in \mathbb{R}$, as

$$\mathbf{H}_\alpha = \mathbf{O} + \alpha \mathbf{E} \quad (10)$$

and observe that the linear interpolation coefficient matrix \mathbf{H} for the red channel is given by

$$\mathbf{H} = \mathbf{H}_{1/2}^{(n \times l)} \otimes \mathbf{H}_{1/2}^{(m \times k)}. \quad (11)$$

With \mathbf{H} written as a Kronecker product in the form of Eq. (11), we can derive the following expression for \mathbf{H}^+ (see Appendix A):

$$\mathbf{H}^+ = \left(\mathbf{H}_{1/2}^{(n \times l)} \right)^+ \otimes \left(\mathbf{H}_{1/2}^{(m \times k)} \right)^+. \quad (12)$$

From the definition of the Kronecker product, Eq. (6) can now be re-formulated as

$$x_i = \sum_{j=1}^{mn} H_{i,j}^+ y_j = \sum_{j=1}^{mn} y_j \left(H_{1/2}^{(n \times l)} \right)_{u,v}^+ \left(H_{1/2}^{(m \times k)} \right)_{w,z}^+ \quad (13)$$

where the matrices $\left(\mathbf{H}_{1/2}^{(n \times l)} \right)^+$ and $\left(\mathbf{H}_{1/2}^{(m \times k)} \right)^+$ are indexed by $(w, u) = \text{vec}^{-1}(i, k)$ and $(z, v) = \text{vec}^{-1}(j, m)$. Thus, in the 2D lattice, (z, v) corresponds to the coordinates of pixel y_j , whereas (w, u) equals the position of pixel x_i in the subsampled ‘genuine’ red image. Note that we only have to invert two matrices of dimension $k \times k$ and $l \times l$ instead of one matrix with dimension $kl \times kl$. Due to the very sparse and regular structure of \mathbf{H}_α , its pseudo-inverse can be expressed solely in terms of $(\mathbf{H}_\alpha^\times)^{-1}$,

$$(H_\alpha)_{p,q}^+ = \begin{cases} (H_\alpha^\times)_{p,r}^{-1} & \text{for } q = 2r - 1 \\ \alpha (H_\alpha^\times)_{p,r}^{-1} + \alpha (H_\alpha^\times)_{p,r+1}^{-1} & \text{for } q = 2r, \end{cases} \quad (14)$$

which allows a very efficient computation of the estimated original sensor signal \mathbf{x} .

Inversion of \mathbf{H}_α^\times From Eqs. (9) and (10) follows that \mathbf{H}_α^\times is symmetric tridiagonal and takes the form

$$\mathbf{H}_\alpha^\times = \begin{bmatrix} 1 + \alpha^2 & \alpha^2 & 0 & \dots & 0 & 0 \\ \alpha^2 & 1 + 2\alpha^2 & \alpha^2 & \ddots & 0 & 0 \\ 0 & \alpha^2 & 1 + 2\alpha^2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 1 + 2\alpha^2 & \alpha^2 \\ 0 & 0 & 0 & \dots & \alpha^2 & 1 + \alpha^2 \end{bmatrix}. \quad (15)$$

We use the method by Huang and McColl¹⁵ to find an analytical solution for the inverse of this matrix and so obtain conditional linear weights for optimal CFA synthesis of the red channel. For their inversion algorithm, Huang and McColl use second-order linear recurrences ζ_i and v_j , which, for a $k \times k$ matrix \mathbf{H}_α^\times , are given by

$$\zeta_i = (1 + 2\alpha^2)\zeta_{i-1} - \alpha^4\zeta_{i-2} \quad i = 2, \dots, k \quad (16)$$

$$v_{k+1-i} = (1 + 2\alpha^2)v_{k+2-i} - \alpha^4v_{k+3-i} = \zeta_i \quad (17)$$

The initial values are set to be $\zeta_0 = v_{k+1} = 1$ and $\zeta_1 = v_k = 1 + \alpha^2$. Now, by using the beforehand defined recurrences, the inversion of \mathbf{H}_α^\times can be written as¹⁵

$$(H_\alpha^\times)_{j,j}^{-1} = (\text{diag}(\mathbf{H}_\alpha^\times)_j - \alpha^4(\xi_{j-1}^{-1} + \gamma_{j+1}^{-1}))^{-1} \quad j = 1, 2, \dots, k \quad (18)$$

$$(H_\alpha^\times)_{i,j}^{-1} = \begin{cases} -\frac{\alpha^2}{\xi_i} (H_\alpha^\times)_{i+1,j}^{-1} & \text{for } i < j \\ -\frac{\alpha^2}{\gamma_i} (H_\alpha^\times)_{i-1,j}^{-1} & \text{for } i > j \end{cases} \quad (19)$$

where the ratios ξ_i and γ_i are given by

$$\xi_i = \frac{\zeta_i}{\zeta_{i-1}} \quad \text{and} \quad \gamma_i = \frac{v_i}{v_{i+1}}. \quad (20)$$

Huang and McColl note that their algorithm has only complexity $\mathcal{O}(k^2)$.

Approximate solution From Eqs.(13) and (14), we can see that \mathbf{x} is basically a filtered version of \mathbf{y} , where the filter coefficients are sampled from the inverse of $\mathbf{H}_{1/2}^\times$. The inverse itself is completely determined by the ratios ξ_i which turn out to have the following asymptotic behavior:¹⁶

$$\lim_{i \rightarrow \infty} \xi_i = q_1. \quad (21)$$

Here, $q_1, q_2, |q_1| \geq |q_2|$, denote the roots of the polynomial $x^2 - (1 + 2\alpha^2)x + \alpha^4 = 0$.

Since $\xi_i = v_{k+1-i}$, the diagonal elements of $(\mathbf{H}_\alpha^\times)^{-1}$ asymptotically approach a limit towards the center,

$$\lim_{\substack{j \rightarrow k/2 \\ k \rightarrow \infty}} \text{diag}((\mathbf{H}_\alpha^\times)^{-1})_j = ((1 + 2\alpha^2) - 2\alpha^4q_1^{-1})^{-1} \quad (22)$$

Hence, for an infinitely large image without border conditions, the filter coefficients $(H_\alpha^\times)_{i,j}^{-1}$ depend only on the above limit, i.e., $(\mathbf{H}_\alpha^\times)^{-1}$ is asymptotically symmetric Toeplitz. It further follows from Eq.(19) that, asymptotically, whenever $|\alpha^2/q_1| < 1$, the column elements of $(\mathbf{H}_\alpha^\times)^{-1}$ decay exponentially from the diagonal elements. This is indeed the case for $\alpha = 1/2$, where

$$q_1 = 3/4 + \sqrt{1/2}. \quad (23)$$

For an approximate solution of the OLS problem, cf. Eq. (13), we can therefore reasonably cut the summation for matrix elements with indices $|u - \lceil v/2 \rceil| > T$ and $|w - \lceil z/2 \rceil| > T$, where T is a threshold for the maximal distance from the diagonal.

Note that the threshold T can be directly related to the horizontal and vertical distances of pixels x_i and y_j in the 2D lattice: Whenever the position of pixel y_j corresponds to that of pixel x_i , we have $u - \lceil v/2 \rceil = 0$ (and equivalently for w and z), i.e., the filter coefficients are drawn from the diagonal. For a pixel y_{j_1} with $\text{vec}^{-1}(j_1) = \text{vec}^{-1}(j) + (2K - 1, 2L - 1)$, the filter weights are the product of $(H_\alpha^\times)_{u,u+K}^{-1}$ and $(H_\alpha^\times)_{w,w+L}^{-1}$. Similarly, for an even shift vector $(2K, 2L)$ we obtain the filter coefficients from a mixture of the columns $K, K + 1$ and $L, L + 1$. Thus, by considering only off-diagonal elements up to a certain degree, we effectively limit the size of the spatial neighborhood which is taken as filter input.

For the case without border conditions we can therefore derive the following linear filter $F_{k,l}$ as approximate solution to our initially stated OLS problem,

$$F_{k,l} = \frac{f_k f_l}{2} \quad \text{with} \quad -\frac{T+1}{2} \leq k, l \leq \frac{T+1}{2}, \quad (24)$$

where the vertical (and equivalently the horizontal) component is given by

$$f_k = \begin{cases} \left(\frac{-1}{4q_1}\right)^r & \text{for } |k| = 2r \\ 1/2 \left(\frac{-1}{4q_1}\right)^r + 1/2 \left(\frac{-1}{4q_1}\right)^{r+1} & \text{for } |k| = 2r - 1 \end{cases} \quad (25)$$

This leads to the following numeric configuration of the asymptotic kernel for low-pass filtering before CFA re-interpolation (a ‘*’ denotes coefficients $|F_{k,l}| < 10^{-3}$):

*	*	*	*	*	*	*	-0.001	*	*	*	*	*	*	*
*	*	*	*	*	*	-0.001	-0.003	-0.001	*	*	*	*	*	*
*	*	*	*	*	-0.001	0.003	0.006	0.003	-0.001	*	*	*	*	*
*	*	*	*	-0.001	-0.003	0.006	0.015	0.006	-0.003	-0.001	*	*	*	*
*	*	*	-0.001	0.003	0.006	0.015	-0.036	-0.036	0.006	0.003	-0.001	*	*	*
*	*	-0.001	-0.003	0.006	0.015	-0.036	-0.036	-0.036	-0.036	0.015	0.006	-0.003	-0.001	*
-0.001	-0.003	0.006	0.015	-0.036	-0.086	0.207	0.500	0.207	-0.086	-0.036	0.015	0.006	-0.003	-0.001
*	-0.001	0.003	0.006	-0.015	-0.036	0.086	0.207	0.086	-0.036	-0.015	0.006	0.003	-0.001	*
*	*	-0.001	-0.003	0.006	0.015	-0.036	-0.086	-0.036	0.015	0.006	-0.003	-0.001	*	*
*	*	*	-0.001	0.003	0.006	-0.015	-0.036	-0.015	0.006	0.003	-0.001	*	*	*
*	*	*	*	-0.001	-0.003	0.006	0.015	0.006	-0.003	-0.001	*	*	*	*
*	*	*	*	*	-0.001	0.003	0.006	0.003	-0.001	*	*	*	*	*
*	*	*	*	*	*	-0.001	-0.003	-0.001	*	*	*	*	*	*
*	*	*	*	*	*	*	*	-0.001	*	*	*	*	*	*

4.2.2. Green channel

Prior to the analysis of the minimum distortion CFA synthesis problem for the green channel, we note that the diagonal structure of the green pixels in the Bayer grid requires a special treatment for border pixels. To ensure that every pixel is interpolated with the same CFA filter coefficients, we introduce imaginary ‘original’ margin pixels. More specifically, we expand an $m \times n$ signal, with $m = 2k$ and $n = 2l$ and k, l integer, by $n + m + 2$ pixels so that the ‘genuine’ green samples \mathbf{y}_G are given by (with $k' = m/2 + 1$)

$$(\mathbf{y}_G)_i = y_j \quad \text{where} \quad j = \begin{cases} i & \text{for } 1 \leq i \leq k' \\ 2(i - k') - \text{col}(i - k', k') + k' & \text{for } k' < i \leq (n + 1)k' \\ i + \frac{mn}{2} & \text{for } i > (n + 1)k' \end{cases} \quad (26)$$

The following example ($m = n = 4$) illustrates the expansion. Samples belonging to \mathbf{y}_{CFA} are marked with an asterisk and samples introduced by the expansion are printed on grey background.

	y_4^*		y_{14}^*		y_{24}^*
y_1^*	y_5	y_9^*	y_{15}	y_{19}^*	
	y_6^*	y_{10}	y_{16}^*	y_{20}	y_{25}^*
y_2^*	y_7	y_{11}^*	y_{17}	y_{21}^*	
	y_8^*	y_{12}	y_{18}^*	y_{22}	y_{26}^*
y_3^*		y_{13}^*		y_{23}^*	

We obtain the following general block matrix from for the CFA interpolation matrix

$$\mathcal{H} = \begin{bmatrix} \mathbf{I}^{(m/2+1)} & & & & & & & & & & \\ \mathbf{A}_1 & \mathbf{B} & \mathbf{A}_1 & & & & & & & & \\ & \mathbf{A}_2 & \mathbf{B} & \mathbf{A}_2 & & & & & & & \\ & & \mathbf{A}_1 & \mathbf{B} & \mathbf{A}_1 & & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & \mathbf{A}_1 & \mathbf{B} & \mathbf{A}_1 & & & & \\ & & & & & & & \mathbf{I}^{(m/2+1)} & & & \end{bmatrix}, \quad (27)$$

where \mathbf{I} is the identity matrix, blank blocks denote matrices with all coefficients zero ($\mathbf{0}$), and \mathbf{A}_1 , \mathbf{A}_2 and \mathbf{B} are $(m+1) \times (m/2+1)$ matrices which hold the interpolation weights as follows:

$$(A_1)_{i,j} = \begin{cases} 1/4 & \text{if } i = 2j \\ 0 & \text{else,} \end{cases} \quad (A_2)_{i,j} = \begin{cases} 1/4 & \text{if } i = 2j - 2 \\ 0 & \text{else,} \end{cases} \quad (B_1)_{i,j} = \begin{cases} 1 & \text{if } i = 2j - 1 \\ 1/4 & \text{if } i \in \{2j, 2-2\} \\ 0 & \text{else,} \end{cases} . \quad (28)$$

Due to the non-separable interpolation kernel, there exists no trivial decomposition of \mathcal{H} into horizontal and vertical components like for the red channel, cf. Eq. (11). However, the signal \mathbf{y} and \mathcal{H} can be reordered such that $\tilde{\mathbf{y}} = [\mathbf{y}_G \ \mathbf{y}_{\text{CFA}}]$ and thus

$$\tilde{\mathcal{H}} = \begin{bmatrix} \mathbf{I} \\ \mathcal{A} \end{bmatrix} \quad \text{and} \quad \mathbf{x} = [(\tilde{\mathcal{H}}^\times)^{-1} \ (\tilde{\mathcal{H}}^\times)^{-1} \mathcal{A}'] \tilde{\mathbf{y}}. \quad (29)$$

Here, \mathbf{y}_G and \mathbf{y}_{CFA} denote the signal \mathbf{y} subsampled at ‘genuine’ green and CFA interpolated positions, respectively. Block matrix \mathcal{A} has dimension $n \times (n+2)$ and is composed of blocks of size $m/2 \times (m/2+1)$,

$$\mathcal{A} = \begin{bmatrix} \overline{\mathbf{A}}_1 & \mathbf{C} & \overline{\mathbf{A}}_1 & & & \\ & \underline{\mathbf{A}}_2 & \mathbf{C} & \underline{\mathbf{A}}_2 & & \\ & & \overline{\mathbf{A}}_1 & \mathbf{C} & \overline{\mathbf{A}}_1 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}. \quad (30)$$

The sub-matrices $\overline{\mathbf{A}}_1$ and $\underline{\mathbf{A}}_2$ hold the non-zero rows of \mathbf{A}_1 and \mathbf{A}_2 , respectively. Similarly, matrix \mathbf{C} contains only those rows of \mathbf{B} with interpolation weights $\neq 1$, hence

$$(\overline{\mathbf{A}}_1)_{i,j} = (A_1)_{2i-1,j}, \quad (\underline{\mathbf{A}}_2)_{i,j} = (A_2)_{2i,j} \quad \text{and} \quad \mathbf{C} = \overline{\mathbf{A}}_1 + \underline{\mathbf{A}}_2. \quad (31)$$

Using some matrix algebra,¹⁷ $(\tilde{\mathcal{H}}^\times)^{-1}$ can now be written as

$$(\tilde{\mathcal{H}}^\times)^{-1} = (\mathbf{I} + \mathcal{A}\mathcal{A}')^{-1} = \mathbf{I} - \mathcal{A}' (\mathbf{I} + \mathcal{A}\mathcal{A}')^{-1} \mathcal{A}, \quad (32)$$

with $\mathbf{I} + \mathcal{A}\mathcal{A}'$ having the property of being block tridiagonal Toeplitz (see Appendix B),*

$$\mathbf{I} + \mathcal{A}\mathcal{A}' = \begin{bmatrix} \mathbf{D} & \mathbf{E} & & & & \\ \mathbf{E}' & \mathbf{D} & \mathbf{E} & & & \\ & \mathbf{E}' & \mathbf{D} & \mathbf{E} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{E}' & \mathbf{D} & \mathbf{E} \\ & & & & \mathbf{E}' & \mathbf{D} \end{bmatrix}. \quad (33)$$

Let the $(n/2 \times n/2)$ block matrix Φ be the inverse of $\mathbf{I} + \mathcal{A}\mathcal{A}'$ and let $\Phi_{i,j}$ be the (i,j) -th $m \times m$ block of Φ . To simplify the notation of boundary conditions, we define ‘outer blocks’ $\Phi_{i,j} = \mathbf{0} \ \forall (i,j) \in \{0, \dots, n/2+1\}^2 \setminus$

*We invert $\mathbf{I} + \mathcal{A}\mathcal{A}'$ instead of $\mathbf{I} + \mathcal{A}'\mathcal{A}$ because the latter is not invertible with Huang and McColl’s method.¹⁵

$\{1, \dots, n/2\}^2$. Then, by making use of the sparse structure of \mathcal{A} , we can write the (i, j) -th $(m+2) \times (m+2)$ block of $(\tilde{\mathcal{H}}^\times)^{-1}$ as

$$(\tilde{\mathcal{H}}^\times)^{-1}_{i,j} = \mathbf{I}_{i,j} - (\mathbf{V}'\Phi_{i-1,j-1} + \mathbf{U}'\Phi_{i,j-1})\mathbf{V} - (\mathbf{V}'\Phi_{i-1,j} + \mathbf{U}'\Phi_{i,j})\mathbf{U} \quad 1 \leq i, j \leq n/2 + 1 \quad (34)$$

with the 2×2 block matrices \mathbf{U} and \mathbf{V} defined as

$$\mathbf{U} = \begin{bmatrix} \bar{\mathbf{A}}_1 & \mathbf{C} \\ \mathbf{0} & \underline{\mathbf{A}}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} \bar{\mathbf{A}}_1 & \mathbf{0} \\ \mathbf{C} & \underline{\mathbf{A}}_2 \end{bmatrix}. \quad (35)$$

Finally, by noting that

$$\left((\tilde{\mathcal{H}}^\times)^{-1} \mathcal{A}' \right)_{i,j} = (\tilde{\mathbf{H}}^\times)^{-1}_{i,j} \mathbf{U}' + (\tilde{\mathbf{H}}^\times)^{-1}_{i,j+1} \mathbf{V}' \quad 1 \leq i \leq n/2 + 1, 1 \leq j \leq n/2 \quad (36)$$

we have all building blocks for an explicit solution of the OLS estimation of the original sensor pixels for the green channel. To shorten our notation, we introduce $\Psi = (\tilde{\mathcal{H}}^\times)^{-1}$. The estimate \mathbf{x} can now be obtained from

$$x_i = \sum_{\kappa=1}^{(n/2+1)(m+2)} (y_G)_\kappa (\Psi_{p,q})_{r,s} + \sum_{\lambda=1}^{mn/2} (y_{\text{CFA}})_\lambda [(\Psi_{p,u} \mathbf{U}') + (\Psi_{p,u+1} \mathbf{V}')]_{r,v}. \quad (37)$$

Similar to the solution for the red channel, the (block) indices of the inverse matrix Ψ can be directly related to the 2D lattice coordinates of x_i and y_j ,

$$(r, p) = \text{vec}^{-1}(i, m+2), \quad (s, q) = \text{vec}^{-1}(\kappa, m+2), \quad \text{and} \quad (v, u) = \text{vec}^{-1}(\lambda, m). \quad (38)$$

Here, (r, p) and (s, q) correspond to the spatial coordinates of the pixels x_i and y_κ in the subsampled $(m+2) \times (n/2+1)$ ‘genuine’ green image. Accordingly, (v, u) is the 2D position of pixel y_λ in the $m \times n/2$ subsampled ‘interpolated’ image. The matrix products in Eq. (37) can be implemented very efficiently due to the specific structure of $\bar{\mathbf{A}}_1$ and $\underline{\mathbf{A}}_2$.

Inversion of $\mathbf{I} + \mathcal{A}\mathcal{A}'$ For the inversion of a general block tridiagonal matrix, Huang and McColl¹⁵ use second-order matrix recurrences similar to those of the non-block case. For the block Toeplitz matrix $\mathbf{I} + \mathcal{A}\mathcal{A}'$ the corresponding matrices \mathbf{Z}_i and \mathbf{Y}_i are defined as

$$\mathbf{E}\mathbf{Z}_i = \mathbf{D}\mathbf{Z}_{i-1} - \mathbf{E}'\mathbf{Z}_{i-2} \quad i = 2, \dots, n/2 \quad (39)$$

$$\mathbf{E}'\mathbf{Y}_j = \mathbf{D}\mathbf{Y}_{j+1} - \mathbf{E}\mathbf{Y}_{j+2} \quad j = n/2 - 1, \dots, 1 \quad (40)$$

with initial matrices set to be $\mathbf{Z}_0 = \mathbf{I}$, $\mathbf{Z}_1 = \mathbf{E}^{-1}\mathbf{D}$ and $\mathbf{Y}_{n/2+1} = \mathbf{I}$, $\mathbf{Y}_{n/2} = \mathbf{E}'^{-1}\mathbf{D}$, respectively. The matrix Φ is then given by

$$\Phi_{j,j} = (\mathbf{D} - \mathbf{E}'\mathbf{Z}_{j-2}\mathbf{Z}_{j-1}^{-1} - \mathbf{E}\mathbf{Y}_{j+2}\mathbf{Y}_{j+1}^{-1})^{-1} \quad (41)$$

$$\Phi_{i,j} = \begin{cases} -\mathbf{Z}_{i-1}\mathbf{Z}_i^{-1}\Phi_{i+1,j} & \text{for } i < j \\ -\mathbf{Y}_{i+1}\mathbf{Y}_i^{-1}\Phi_{i-1,j} & \text{for } i > j \end{cases}. \quad (42)$$

Approximate solution Compared to the red channel, the computation of the estimate \mathbf{x} for the green channel is much more expensive. Although the inversion on a block level makes the CFA synthesis more tractable than the trivial approach, we still have to deal with matrix multiplications instead of scalar multiplications. An approximate solution is such most valuable here, as it can avoid considerable complexity.

From Eqs. (41) and (42) we can see that the inverse matrix Φ is completely determined by the terms $\mathbf{Z}_{i-1}\mathbf{Z}_i^{-1}$ and $\mathbf{Y}_j\mathbf{Y}_{j-1}^{-1}$, respectively. In order to formulate an approximate solution based on the asymptotic behavior of the ‘ratios’ of second order matrix recurrences we are in the need of an analytical expression of the matrices \mathbf{Z}_i

and \mathbf{Y}_j . Unfortunately, we are unable to give a closed-form expression for the coefficients of the inverse matrix. This is mainly due to the non-commuting matrices $\mathbf{E}^{-1}\mathbf{D}$ and $\mathbf{E}^{-1}\mathbf{E}'$ which preclude an analytical solution of the general form of the polynomial $\mathbf{X}\mathbf{X} - \mathbf{E}^{-1}\mathbf{D}\mathbf{X} + \mathbf{E}^{-1}\mathbf{E}' = \mathbf{0}$ (and corresponding for the recurrences \mathbf{Y}_j). Nevertheless, numerical experiments suggest that the ratios approach finite limits \mathbf{Q}_Z and \mathbf{Q}_Y with

$$\mathbf{Q}_Z = \lim_{i \rightarrow \infty} \mathbf{Z}_{i-1} \mathbf{Z}_i^{-1} \quad \text{and} \quad \mathbf{Q}_Y = \lim_{j \rightarrow 1} \mathbf{Y}_j \mathbf{Y}_{j-1}^{-1}. \quad (43)$$

More specifically, we found for a variety of matrix dimensions $m \times m$ that $\mathbf{Z}_{i-1} \mathbf{Z}_i^{-1}$ and $\mathbf{Y}_j \mathbf{Y}_{j-1}^{-1}$ approach constant matrices \mathbf{Q}_Z and \mathbf{Q}_Y , respectively, after only a very small number of iterations.

Assuming the existence of \mathbf{Q}_Z and \mathbf{Q}_Y we can proceed analogously to the red channel and define \mathbf{Q}_D as the asymptotic diagonal block of Φ ,

$$\mathbf{Q}_D = \lim_{\substack{j \rightarrow n/4 \\ n \rightarrow \infty}} \Phi_{j,j} = (\mathbf{D} - \mathbf{E}'\mathbf{Q}_Z + \mathbf{E}\mathbf{Q}_Y)^{-1}. \quad (44)$$

The off-diagonal elements can then be approximated by

$$\Phi_{i,j}^\infty = \begin{cases} (-\mathbf{Q}_Z)^{j-i} \mathbf{Q}_D & \text{for } i \leq j \\ (-\mathbf{Q}_Y)^{i-j} \mathbf{Q}_D & \text{for } i \geq j \end{cases}. \quad (45)$$

Since we lack an analytical description of the matrices \mathbf{Q}_Z and \mathbf{Q}_Y , we compute the approximate filter kernel numerically based on a pixel in the center row of an image without border conditions. Therefore, we compute the asymptotic matrix Ψ^∞ from the matrices \mathbf{Q}_D , \mathbf{Q}_Y and \mathbf{Q}_Z . The filter coefficients $F_{k,l}$ can then be obtained from the following general properties, cf. Eqs. (37) and (38):

1. The center coefficient $F_{0,0}$ is applied to a pixel $(y_G)_\kappa$. Without loss of generality, we assume that $\text{col}(y_G)_\kappa, m+2) = 2j+1$, $j \leq 0$. Matrix Ψ^∞ is indexed by $p=q$ and $r=s$.
2. Filter coefficients $F_{k,l}$ with $(k,l) \in \{2K, 2L\} \cup \{2K+1, 2L+1\}$ correspond to ‘genuine’ green pixels. Since two columns of the image are captured in one block of the matrix Ψ^∞ , the filter coefficients are given as elements of the sub-matrix $\Psi_{p,p+L}^\infty$. For an even column shift $2L$, we remain in an even column of the image and therefore $s = r + K$. For an odd shift $2L+1$, however, we change from an even to an odd column and $s = r + K + (m/2 + 1)$.
3. Filter coefficients $F_{k,l}$ with $(k,l) \in \{2K, 2L+1\} \cup \{2K+1, 2L\}$ correspond to CFA interpolated pixels. The filter coefficients for these pixels are the (r,v) -th element of the matrix $[(\Psi_{p,u}^\infty \mathbf{U}') + (\Psi_{p,u+1}^\infty \mathbf{V}')]'$. Since the sub-images y_G and y_{CFA} are displaced by one column, we have $u = p - 1 + \lceil l/2 \rceil$. Similar to the previous case, we change from even to odd column for an odd column shift $2L+1$, thus $v = r + \lceil k-1/2 \rceil + m/2$. For an even shift $2L$, v is given by $r + \lceil k-1/2 \rceil$.

By computing the corresponding matrix elements at the above indices for sufficiently large dimensions m and n , we obtain the following numeric configuration of the asymptotic kernel for low-pass filtering before CFA re-interpolation (a ‘*’ denotes coefficients $|F_{k,l}| < 10^{-3}$):

*	*	-0.001	0.001	0.001	0.001	-0.001	*	*
*	-0.001	0.003	0.005	-0.004	0.005	0.003	-0.001	*
-0.001	0.003	0.009	-0.022	-0.029	-0.022	0.009	0.003	-0.001
0.001	0.005	-0.022	-0.072	0.165	-0.072	-0.022	0.005	0.001
0.001	-0.004	-0.029	0.165	0.835	0.165	-0.029	-0.004	0.001
0.001	0.005	-0.022	-0.072	0.165	-0.072	-0.022	0.005	0.001
-0.001	0.003	0.009	-0.022	-0.029	-0.022	0.009	0.003	-0.001
*	-0.001	0.003	0.005	-0.004	0.005	0.003	-0.001	*
*	*	-0.001	0.001	0.001	0.001	-0.001	*	*

4.2.3. Blue channel

Due to the equality in Eq. (8), the approach described for the red channel applies to the blue channel as well.

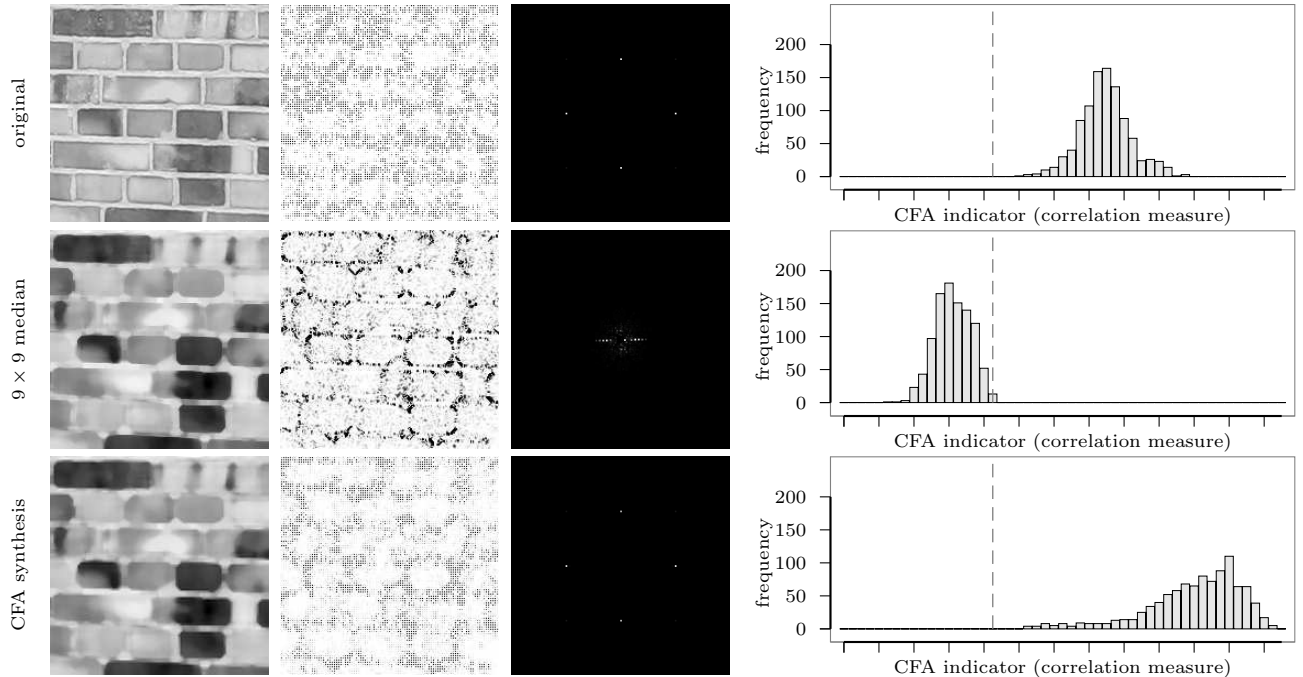


Figure 1. CFA pattern detection for the red channel of a single genuinely interpolated RGB image (left part, top row), its 9×9 median filtered version (middle row), and filtered version after CFA synthesis (bottom row); p -maps (second column) and their Fourier spectra (third column). The p -map after CFA synthesis is indistinguishable from the original one—unlike the one after filtering. Right part: quantitative results from 1000 images, aggregated as histograms over the CFA indicator, a measure of the strength of the CFA artifacts; identical processing as in the respective row on the left.

5. EXPERIMENTAL RESULTS

As we pointed out in Ref. 18, attacks against digital image forensics should be evaluated both in terms of undetectability and resulting image quality. In order to demonstrate the appropriateness of our suggested synthesis method of CFA interpolation artifacts, we have to provide evidence that a state-of-the-art CFA detector cannot distinguish between genuinely CFA interpolated and our synthesized images. Furthermore, a minimal distortion synthesis should yield measurable higher image quality than naive re-interpolation.

For a quantitative analysis, we use Andrew Ker’s ‘gold standard’ image set for steganalysis benchmarks.¹⁹ It contains 1500×2000 full resolution never-compressed images taken with a Minolta DiIMAGE A1 digital camera and stored in RAW format. Minolta’s standard software was used for the conversion to RGB TIFFs.

Detectability We measure detectability using our fast version¹¹ of Popescu and Farid’s state-of-the-art detector,³ cf. Sect. 3. An image is considered to contain traces of CFA interpolation whenever the spectrum of its p -map exhibits clear peaks in either the highest horizontal, vertical or diagonal frequencies. For automated detection, Popescu and Farid propose to use a correlation measure between the p -map of the image under investigation and an ‘ideal’ p -map with CFA artifacts. See Ref. 3 for a detailed description of this indicator. Here, it is only important to note that larger positive values correspond to a stronger CFA pattern.

The left part of Fig.1 illustrates the effectiveness of our proposed CFA synthesis for a test image which was generated using linear CFA interpolation (top row), filtered with a 9×9 median filter to mimic tampering (middle row), and finally processed by our CFA synthesis method (bottom row). The three gray scale images per row depict the red channel of the image, its p -map and the Fourier spectrum of the p -map. Observe the distinct peaks for the CFA interpolated image, indicating an authentic image. As CFA interpolation is similar to upsampling by a factor of 2, it yields characteristic peaks at the highest horizontal and vertical (and, for some grids, diagonal) frequencies.¹¹ In order to increase the visibility of the peaks, we enlarge the p -map by a factor of

Table 1. Gain in image quality, ΔPSNR [dB], of minimal distortion CFA synthesis as opposed to naive re-interpolation; red and green color channel. Quartiles and inter-quartile ranges (IQR) over 1000 images sized 512×512 .

red channel				green channel			
Q_{25}	Q_{50}	Q_{75}	IQR	Q_{25}	Q_{50}	Q_{75}	IQR
1.07	1.18	1.28	0.21	0.89	0.94	0.99	0.10

2 prior to the Fourier transform, which corresponds to doubling the cycle length of the periodic pattern. Hence, the peaks are shifted to half of their original frequency.³

Since strong median filtering is likely to destroy the specific linear dependencies in interpolated images,¹⁸ it is not surprising that the spectrum of the filtered image’s p -map shows no characteristic peaks and thus appears as ‘forgery’ due to the absence of intact CFA pattern. However, after applying our CFA synthesis procedure (bottom row), the ‘correct’ peaks re-appear, i.e., a human investigator would not be able to visually differentiate between the detector response of this processed image and a genuinely CFA interpolated image.

The reported existence of the CFA peaks in the latter case is an inevitable consequence of the re-interpolation of the image: it introduces exactly those characteristics the detector was designed to find. It is therefore not really a question whether we can mislead the detector in terms of the existence of distinct peaks in the p -map’s spectrum. However, we have to bear in mind that, in typical digital cameras, CFA interpolation is by far not the last step in ever more complex image processing chains. As a consequence, we will not find as perfect pattern in original images as in our synthesized images.

This is illustrated in the right part of Fig. 1. It depicts the histograms of the red channel detector responses from about 1000 images (cropped to 512×512) of the three types: original camera images, 9×9 median filtered images, and CFA synthesized images. First of all, in line with our expectations, observe that the detector can perfectly distinguish between original images and those processed with a median filter. However, if the detector threshold is set in such a way that it is not subject to false positives (dashed grey line), it cannot decide whether an image was produced by a camera or is actually synthesized. Nevertheless, it is also evident that the synthesized images, on average, yield a higher detector response due to the absence of (non-linear) post-processing. Obviously, a more sophisticated detector would label too perfect traces of CFA interpolation as suspicious, too. An ultimate evaluation of detectability is therefore beyond the scope of this paper and believed to be non-trivial. While strength and type of CFA artifacts strongly depend on the in-camera processing and therefore differ between camera types,[†] a more sophisticated ‘attack’ would try to mimic typical camera post-processing and thus smooth out perfect interpolation traces. Nevertheless, a procedure for optimal CFA synthesis, as described in this paper, forms an essential building block for more refined synthesizers of combined processing artefacts.

Image quality To assess the visual quality of CFA synthesized images we employ the popular measure of peak-signal-to-noise ratio, PSNR. Here, it is important to note that the most influential factor is the image content itself. For example, highly textured images will generally produce low PSNR values between original and re-interpolated version. For a comparison of two different processing chains, we therefore measure the difference of PSNR values between the two processed version \mathbf{y}_1 and \mathbf{y}_2 and one reference image \mathbf{y}_0 ,

$$\Delta \text{PSNR}(\mathbf{y}_1, \mathbf{y}_2; \mathbf{y}_0) = \text{PSNR}(\mathbf{y}_1, \mathbf{y}_0) - \text{PSNR}(\mathbf{y}_2, \mathbf{y}_0), \quad (46)$$

rather than interpreting plain PSNR values.

Table 1 compares the visual quality after the application of minimal distortion synthesis and naive re-interpolation. The listed quartiles for the red and green channel were obtained from approximately 1000 images, which have been 9×9 median filtered prior to CFA synthesis. The figures provide sound evidence that our

[†]This is suggested by the successful classification of different camera types and brands based on CFA artefacts.^{13,14,20}

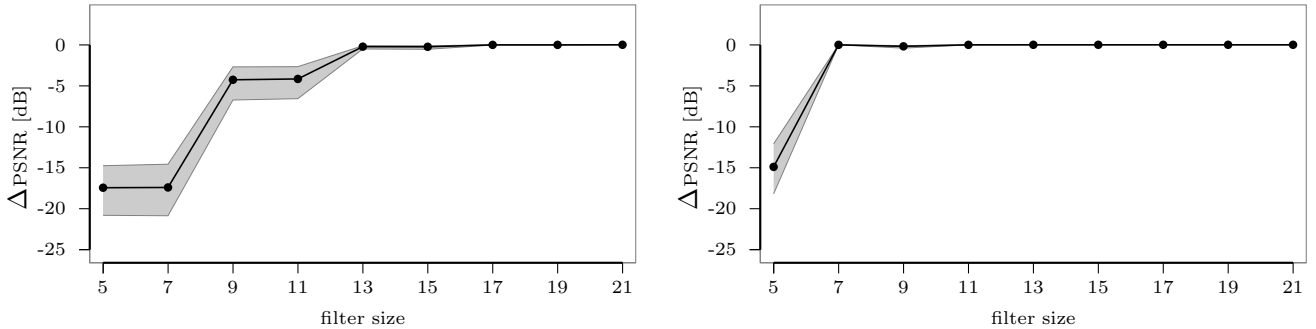


Figure 2. Loss in image quality, median Δ PSNR [dB], of approximate as opposed to exact solution; red (left) and green (right) channel for different filter sizes. The shaded areas denote inter-quartile ranges over 1000 images sized 512×512 .

proposed approach indeed yields better quality than naive re-interpolation, as both median and lower quartile estimates of Δ PSNR are strictly positive and not negligible.

Besides the exact solutions to the originally stated OLS problem (Eq. (7)), we also proposed approximate solutions of complexity $\mathcal{O}(mn)$. Figure 2 compares the visual quality of the exact and the approximate solutions for different window sizes of the approximated filter. As can be seen from the curves, the approximate solution for both the red and the green channel approach the image quality of the exact solution with increasing filter size.[‡] This is due to the exponential decay of the off-diagonal elements in the inverse matrices $(\mathbf{H}^\times)^{-1}$. In line with the different kernel sizes for a fixed cut-off value, the filter size required for a satisfactory approximation of the red channel is at least 13×13 , while for the green channel already a 7×7 filter produces good results.

6. CONCLUDING REMARKS

We have formulated optimal CFA synthesis in color images as least squares problem and presented novel methods to exploit its special structure to find the optimal solution efficiently and even more efficient near-optimal (approximate) solutions with linear complexity (in the number of pixels). This method can serve as building block for more sophisticated tamper hiding techniques to defeat forensic analyses based on CFA artifacts.

Other applications include steganography: the existence of a secret message embedded in a cover image can be further concealed by restoring the CFA pattern of the resulting stego image, which such will exhibit ‘plausible’ or ‘natural’ dependencies between pixels in a local neighborhood. Future research has to tell how secure these stego images are against steganalyzers that typically exploit such dependencies.

While this paper is confined to bi-linear interpolation, obvious next steps are generalizations to other, more sophisticated, interpolation methods. It is an open question whether similar efficient methods can be found to synthesize pattern of signal-adaptive CFA interpolation algorithms with minimal distortion. Related to this, further open questions emerge on the reaction of forensic techniques that try to identify the type, brand or model of the acquisition device based on CFA artifacts^{13,14,20,21} to our synthesized images. It is up to future experiments to study if the synthesis procedure can be tuned to mimic various acquisition devices deceptively.

As to the method’s limitations, one has to bear in mind that CFA interpolation is not necessarily the last step in ever more complex image processing chains of digital image acquisition devices. For example, we found raw images with (locally) incomplete CFA pattern. So forgeries that exhibit ‘perfect’ CFA artefacts may cause suspicion if traces from further processing are not appropriately restored, too.

Finally, while the proposed method finds a continuous optimum efficiently, its solution after rounding to integers does not necessarily concur with the discrete optimum of intensity values rounded to integers (or quantized integers in a transformed domain, such as JPEG). It is possible to conceive new heuristic algorithms that improve our optimum further based on an iterate analysis of the residuals. Finding the discrete optimum, however, is NP-hard and thus hardly manageable for real images.

[‡]Mind that the detectability is roughly the same for both approaches.

10. A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B* **39**(1), pp. 1–38, 1977.
11. M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *MM&Sec'08, Proceedings of the Multimedia and Security Workshop 2008, September 22-23, 2008, Oxford, UK*, pp. 11–20, ACM Press, (New York, NY, USA), 2008.
12. Y. Huang, "Can digital image forgery detection be unevadable? A case study: color filter array interpolation statistical feature recovery," in *Proceedings of SPIE: Visual Communications and Image Processing*, S. Li, F. Pereira, H.-Y. Shum, and A. G. Tescher, eds., **SPIE 5960**, 59602W, 2005.
13. A. Swaminathan, M. Wu, and K. J. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security* **2**(1), pp. 91–106, 2007.
14. S. Bayram, H. T. Sencar, and N. Memon, "Classification of digital camera-models based on demosaicing artifacts," *Digital Investigation* **5**, pp. 46–59, 2008.
15. Y. Huang and W. McColl, "Analytical inversion of general tridiagonal matrices," *Journal of Physics A: Mathematical and General* **30**, pp. 7919–7933, 1997.
16. P. Kiss and Z. Sinka, "On the ratios of the terms of second order linear recurrences," *Periodica Mathematica Hungaria* **23**(2), pp. 139–143, 1991.
17. S. R. Searle, *Matrix Algebra Useful for Statistics*, Wiley Interscience, Hoboken, NJ, 2006.
18. M. Kirchner and R. Böhme, "Hiding traces of resampling in digital images," *IEEE Transactions on Information Forensics and Security* **3**(4), pp. 582–592, 2008.
19. A. D. Ker and R. Böhme, "Revisiting weighted stego-image steganalysis," in *Proceedings of SPIE-IS&T Electronic Imaging: Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, E. J. Delp, P. W. Wong, N. D. Memon, and J. Dittmann, eds., **SPIE 6819**, 681905, 2008.
20. Y. Long and Y. Huang, "Image based source camera identification using demosaicking," in *Proceedings of the 8th IEEE Workshop on Multimedia Signal Processing*, pp. 419–424, 2006.
21. C. McKay, A. Swaminathan, H. Gou, and M. Wu, "Image acquisition forensics: Forensic analysis to identify imaging source," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, pp. 1657–1660, 2008.