Exposing Digital Forgeries in Color Filter Array Interpolated Images

Alin C. Popescu and Hany Farid

***Abstract*—With the advent of low-cost and high-resolution digital cameras, and sophisticated photo editing software, digital images can be easily manipulated and altered. Although good forgeries may leave no visual clues of having been tampered with, they may, nevertheless, alter the underlying statistics of an image. Most digital cameras, for example, employ a single sensor in conjunction with a color filter array (CFA), and then interpolate the missing color samples to obtain a three channel color image. This interpolation introduces specific correlations which are likely to be destroyed when tampering with an image. We quantify the specific correlations introduced by CFA interpolation, and describe how these correlations, or lack thereof, can be automatically detected in any portion of an image. We show the efficacy of this approach in revealing traces of digital tampering in lossless and lossy compressed color images interpolated with several different CFA algorithms.**

***Index Terms*—Digital forensics, digital tampering.**

I. Introduction

WHEN it first surfaced, the digital image of Jane Fonda and Senator John Kerry sharing a stage at an anti-war rally was explosive.[1] It was also a fake. The image was created by digitally splicing together two separate images. Because of the ease with which digital media can be manipulated, these types of digital forgeries are becoming more common. As a result, photographs no longer hold the unique stature as a definitive recording of events. Digital watermarking has been proposed as a means by which an image can be authenticated (see, for example, [1] and [2] for general surveys). The major drawback of this approach is that a watermark must be inserted at the time of recording, which would limit this approach to specially equipped digital cameras. This method also relies on the assumption that the digital watermark cannot be easily removed and reinserted—it is not yet clear whether this is a reasonable assumption (e.g., [3]). In contrast to these approaches, we and others have proposed techniques for detecting tampering in digital images that work in the absence of any digital watermark or signature [4]–[10]. These techniques work on the assumption that although digital forgeries may leave no visual clues of having been tampered with, they may, nevertheless, alter the underlying statistics of an image. Most digital cameras, for example, capture color images using a single sensor in conjunction with an array of color filters. As a result, only one third of the samples in a color image are captured by the camera—the other two thirds being interpolated. This interpolation introduces specific correlations between the samples of a color image. When creating a digital forgery, these correlations may be destroyed or altered. We describe the form of these correlations and propose a method that quantifies and detects them in any portion of an image.[2] We show the general effectiveness of this technique in detecting traces of digital tampering and analyze its sensitivity and robustness to simple image distortions (compression, noise, and gamma correction).

Manuscript received June 24, 2004; revised January 18, 2005. This work was supported by an Alfred P. Sloan Fellowship, a National Science Foundation CAREER Award (IIS-99-83806), a departmental National Science Foundation Infrastructure Grant (EIA-98-02068), and under Award no. 2000-DT-CX-K001 from the Office for Domestic Preparedness, U.S. Department of Homeland Security (points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security). The associate editor coordinating the review of this manuscript and approving it for publication was Guest Editor Dr. Ingemar J. Cox.

A. C. Popescu is with the Computer Science Department, Dartmouth College, Hanover, NH 03755 USA.

H. Farid is with the 6211 Sudikoff Lab, Computer Science Department, Dartmouth College, Hanover, NH 03755 USA (e-mail: farid@cs.dartmouth.edu).

Digital Object Identifier 10.1109/TSP.2005.855406

[1]The actress Jane Fonda was well known for her strong views against the involvement of the United States in the Vietnam war. After serving in the Vietnam war, John Kerry spoke out against the war. The photograph of Fonda and Kerry appeared as Kerry was running for President of the United States and came at a time when his involvement in the antiwar movement was under attack.

II. Color Filter Array Interpolation Algorithms

A digital color image consists of three channels containing samples from different bands of the color spectrum, e.g., red, green, and blue. Most digital cameras, however, are equipped with a single charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) sensor and capture color images using a color filter array (CFA). The most frequently used CFA, known as the Bayer array [12], employs three color filters: red, green, and blue. The red and blue pixels are sampled on rectilinear lattices, whereas the green pixels are sampled on a quincunx lattice; see Fig. 1. Since only a single color sample is recorded at each pixel location, the other two color samples must be estimated from the neighboring samples in order to obtain a three-channel color image. Let $S(x,y)$ denote the CFA image in Fig. 1 and $\tilde{R}(x,y), \tilde{G}(x,y), \tilde{B}(x,y)$ denote the red, green, and blue channels constructed from $S(x,y)$ as follows:

$$\tilde{R}(x,y) = \begin{cases} S(x,y), & \text{if } S(x,y) = r_{x,y} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\tilde{G}(x,y) = \begin{cases} S(x,y), & \text{if } S(x,y) = g_{x,y} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\tilde{B}(x,y) = \begin{cases} S(x,y), & \text{if } B(x,y) = b_{x,y} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where (x,y) span an integer lattice. A complete color image, with channels $R(x,y), G(x,y)$, and $B(x,y)$, needs to be

[2]Portions of this work also appear in the Ph.D. dissertation [11].

$r_{1,1}$	$g_{1,2}$	$r_{1,3}$	$g_{1,4}$	$r_{1,5}$	$g_{1,6}$	
$g_{2,1}$	$b_{2,2}$	$g_{2,3}$	$b_{2,4}$	$g_{2,5}$	$b_{2,6}$	
$r_{3,1}$	$g_{3,2}$	$r_{3,3}$	$g_{3,4}$	$r_{3,5}$	$g_{3,6}$	
$g_{4,1}$	$b_{4,2}$	$g_{4,3}$	$b_{4,4}$	$g_{4,5}$	$b_{4,6}$	$\cdots$
$r_{5,1}$	$g_{5,2}$	$r_{5,3}$	$g_{5,4}$	$r_{5,5}$	$g_{5,6}$	
$g_{6,1}$	$b_{6,2}$	$g_{6,3}$	$b_{6,4}$	$g_{6,5}$	$b_{6,6}$	
			$\vdots$			$\ddots$

Fig. 1. Top-left portion of a CFA image obtained from a Bayer array. The red $r_{2i+1,2j+1}$ and blue $b_{2i,2j}$ pixels are sampled on rectilinear lattices, whereas the green $g_{2i+1,2j}$ and $g_{2i,2j+1}$ pixels are sampled twice as often on a quincunx lattice. Notice that at each pixel location only a single color sample is recorded.

estimated. These channels take on the nonzero values of $\tilde{R}(x,y), \tilde{G}(x,y)$, and $\tilde{B}(x,y)$ and replace the zeros with estimates from neighboring samples. The estimation of the missing color samples is referred to as CFA interpolation or demosaicking. CFA interpolation has been extensively studied, and many methods have been proposed (see, for example, [13] for a survey and [14]–[16] for more recent methods). We will briefly describe seven demosaicking techniques for the Bayer array that are used in our studies. In selecting these seven algorithms, we have tried to cover a range of CFA algorithms, from the simple (bilinear), to ones that are employed in commercially available digital cameras (gradient-based), to more recent algorithms that may one day be employed in commercial digital cameras.

A. Bilinear and Bicubic

The simplest methods for demosaicking are kernel-based interpolation methods that act on each channel independently. These methods can be efficiently implemented as linear filtering operations on each color channel:

$$R(x,y) = \sum_{u,v=-N}^{N} h_r(u,v)\tilde{R}(x-u, y-v)$$
$$G(x,y) = \sum_{u,v=-N}^{N} h_g(u,v)\tilde{G}(x-u, y-v)$$
$$B(x,y) = \sum_{u,v=-N}^{N} h_b(u,v)\tilde{B}(x-u, y-v)$$

where $\tilde{R}(\cdot), \tilde{G}(\cdot), \tilde{B}(\cdot)$ are defined in (1)–(3), and $h_r(\cdot), h_g(\cdot), h_b(\cdot)$ are linear filters of size $(2N+1) \times (2N+1)$. Different forms of interpolation (nearest neighbor, bilinear, bicubic [17], etc.) differ in the form of the interpolation filter used. For the Bayer array, the bilinear and bicubic filters for the red and blue channels are separable. The one–dimensional (1-D) filters are given by

$$h_l = [1/2 \quad 1 \quad 1/2]$$
$$h_c = [-1/16 \quad 0 \quad 9/16 \quad 1 \quad 9/16 \quad 0 \quad -1/16]$$

and the two–dimensional (2-D) filters are given by $h_l^T h_l$ and $h_c^T h_c$. The bilinear and bicubic filters for the green channel, however, are no longer separable and take the form

$$h_l = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
$$h_c = \frac{1}{256}\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 1 & 0 & 81 & 256 & 81 & 0 & 1 \\ 0 & -9 & 0 & 81 & 0 & -9 & 0 \\ 0 & 0 & -9 & 0 & -9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Using bilinear interpolation, for example, the color samples at pixel locations (3, 3), (3, 4), and (4, 4), Fig. 1 are given by

$$R(3,3) = r_{3,3} \quad G(3,3) = \frac{g_{2,3} + g_{3,2} + g_{3,4} + g_{4,3}}{4}$$
$$B(3,3) = \frac{b_{2,2} + b_{2,4} + b_{4,2} + b_{4,4}}{4}$$
$$G(3,4) = g_{3,4} \quad B(3,4) = \frac{b_{2,4} + b_{4,4}}{2}$$
$$R(3,4) = \frac{r_{3,3} + r_{3,5}}{2}$$
$$B(4,4) = b_{4,4} \quad R(4,4) = \frac{r_{3,3} + r_{3,5} + r_{5,3} + r_{5,5}}{4}$$
$$G(4,4) = \frac{g_{3,4} + g_{4,3} + g_{4,5} + g_{5,4}}{4}.$$

B. Smooth Hue Transition

Smooth hue transition CFA interpolation [18] is based on the assumption that color hue varies smoothly in natural images and can be assumed to be constant in a small neighborhood. Hue is defined here as the ratio between a chrominance component (red or blue) and the luminance component (green). The algorithm interpolates the green channel $\tilde{G}(\cdot)$ [see (2)] using bilinear interpolation (see Section II-A) to yield $G(\cdot)$. To estimate the missing red samples, the ratio $\tilde{R}(\cdot)/G(\cdot)$ is bilinearly interpolated and then pointwise multiplied by $G(\cdot)$. For example, the missing red values at locations (3, 4), (4, 3), and (4, 4) in Fig. 1 are given by

$$R(3,4) = G(3,4) \cdot \frac{1}{2}\left(\frac{r_{3,3}}{G(3,3)} + \frac{r_{3,5}}{G(3,5)}\right)$$
$$R(4,3) = G(4,3) \cdot \frac{1}{2}\left(\frac{r_{3,3}}{G(3,3)} + \frac{r_{5,3}}{G(5,3)}\right)$$
$$R(4,4) = G(4,4) \cdot \frac{1}{4}\left(\frac{r_{3,3}}{G(3,3)} + \frac{r_{3,5}}{G(3,5)} + \frac{r_{5,3}}{G(5,3)} + \frac{r_{5,5}}{G(5,5)}\right).$$

The missing blue samples are estimated in a similar manner as the red samples.

C. Median Filter

Median filter-based CFA interpolation [19] is a two-step process. First, the $\tilde{R}(\cdot), \tilde{G}(\cdot), \tilde{B}(\cdot)$ channels (1)–(3) are each interpolated using bilinear interpolation (see Section II-A).

Next, the pairwise differences of the interpolated channels (i.e., red minus green, red minus blue, and green minus blue) are median filtered. Let $M_{rg}(\cdot), M_{gb}(\cdot)$, and $M_{rb}(\cdot)$ denote the median-filtered differences. At each pixel location, the missing color samples are estimated as the sum or difference between the color samples from the CFA image (see Fig. 1) and the corresponding median filtered differences. For example, the missing samples at locations (3, 3), (3, 4), and (4, 4) are estimated as follows:

$$\begin{aligned} G(3,3) &= r_{3,3} - M_{rg}(3,3) \quad B(3,3) = r_{3,3} - M_{rb}(3,3) \\ R(3,4) &= g_{3,4} + M_{rg}(3,4) \quad B(3,4) = g_{3,4} - M_{gb}(3,4) \\ R(4,4) &= b_{4,4} + M_{rb}(4,4) \quad G(4,4) = b_{4,4} + M_{gb}(4,4). \end{aligned}$$

D. Gradient-Based

Gradient-based CFA interpolation [20] is a multistep method that preserves edge information by preventing interpolation across edges. The green channel $G(\cdot)$ is first estimated using an adaptive interpolation technique. For example, in order to estimate the green values at location (3, 3) and (4, 4) (see Fig. 1), derivative estimators along the horizontal and vertical directions are first computed:

$$\begin{aligned} H_{3,3} &= |(r_{3,1} + r_{3,5})/2 - r_{3,3}| \\ V_{3,3} &= |(r_{1,3} + r_{5,3})/2 - r_{3,3}| \\ H_{4,4} &= |(b_{4,2} + b_{4,6})/2 - b_{4,4}| \\ V_{4,4} &= |(b_{2,4} + b_{6,4})/2 - b_{4,4}|. \end{aligned}$$

These estimators are referred to as classifiers. The estimated green values are then given by

$$G(3,3) = \begin{cases} \frac{g_{3,2}+g_{3,4}}{2}, & H_{3,3} < V_{3,3} \\ \frac{g_{2,3}+g_{4,3}}{2}, & H_{3,3} > V_{3,3} \\ \frac{g_{2,3}+g_{3,2}+g_{3,4}+g_{4,3}}{4}, & H_{3,3} = V_{3,3} \end{cases}$$

$$G(4,4) = \begin{cases} \frac{g_{4,3}+g_{4,5}}{2}, & H_{4,4} < V_{4,4} \\ \frac{g_{3,4}+g_{5,4}}{2}, & H_{4,4} > V_{4,4} \\ \frac{g_{3,4}+g_{4,3}+g_{4,5}+g_{5,4}}{4}, & H_{4,4} = V_{4,4} \end{cases}.$$

The missing red samples are estimated by taking the difference image between the $\tilde{R}(\cdot)$ and $G(\cdot)$ channels, bilinearly interpolating this difference, then adding back $G(\cdot)$. For example, the missing red samples at (3, 4), (4, 3), and (4, 4) (see Fig. 1) are given by

$$\begin{aligned} R(3,4) &= G(3,4) + \frac{1}{2}((r_{3,3} - G(3,3)) + (r_{3,5} - G(3,5))) \\ R(4,3) &= G(4,3) + \frac{1}{2}((r_{3,3} - G(3,3)) + (r_{5,3} - G(5,3))) \\ R(4,4) &= G(4,4) + \frac{1}{4}((r_{3,3} - G(3,3)) \\ &\quad + (r_{3,5} - G(3,5)) + (r_{5,3} - G(5,3)) \\ &\quad + (r_{5,5} - G(5,5))). \end{aligned}$$

The missing blue samples are estimated in a similar manner as the red samples.

E. Adaptive Color Plane

Adaptive color plane CFA interpolation [21] is a modification of the gradient-based interpolation method. The green channel $G(\cdot)$ is first estimated using adaptive interpolation. For example, in order to estimate the green values at (3, 3) and (4, 4) (Fig. 1), horizontal and vertical classifiers are computed as the sum between first-order derivative estimates from the luminance (green) channel and second-order derivative estimates from the chrominance (red and blue) channels:

$$\begin{aligned} H_{3,3} &= |g_{3,2} - g_{3,4}| + |-r_{3,1} + 2r_{3,3} - r_{3,5}| \\ V_{3,3} &= |g_{2,3} - g_{4,3}| + |-r_{1,3} + 2r_{3,3} - r_{5,3}| \\ H_{4,4} &= |g_{4,3} - g_{4,5}| + |-b_{4,2} + 2b_{4,4} - b_{4,6}| \\ V_{4,4} &= |g_{3,4} - g_{5,4}| + |-b_{2,4} + 2b_{4,4} - b_{6,4}|. \end{aligned}$$

Using these classifiers, the estimated green values are given by

$$G(3,3) = \begin{cases} \frac{g_{3,2}+g_{3,4}}{2} + \frac{-r_{3,1}+2r_{3,3}-r_{3,5}}{4}, & H_{3,3} < V_{3,3} \\ \frac{g_{2,3}+g_{4,3}}{2} + \frac{-r_{1,3}+2r_{3,3}-r_{5,3}}{4}, & H_{3,3} > V_{3,3} \\ \frac{g_{2,3}+g_{3,2}+g_{3,4}+g_{4,3}}{4} & \\ \quad + \frac{-r_{1,3}-r_{3,1}+4r_{3,3}-r_{3,5}-r_{5,3}}{8}, & H_{3,3} = V_{3,3} \end{cases}$$

$$G(4,4) = \begin{cases} \frac{g_{4,3}+g_{4,5}}{2} + \frac{-b_{4,2}+2b_{4,4}-b_{4,6}}{4}, & H_{4,4} < V_{4,4} \\ \frac{g_{3,4}+g_{5,4}}{2} + \frac{-b_{2,4}+2b_{4,4}-b_{6,4}}{4}, & H_{4,4} > V_{4,4} \\ \frac{g_{3,4}+g_{4,3}+g_{4,5}+g_{5,4}}{4} & \\ \quad + \frac{-b_{2,4}-b_{4,2}+4b_{4,4}-b_{4,6}-b_{6,4}}{8}, & H_{4,4} = V_{4,4}. \end{cases}$$

The estimation of the missing chrominance samples depends on their location in the CFA. At locations where the nearest chrominance samples are in the same row or column, a nonadaptive procedure with second-order correction terms is employed. For example, red and blue samples at (3, 4) and (4, 3) are given by

$$\begin{aligned} R(3,4) &= \frac{r_{3,3} + r_{3,5}}{2} + \frac{-G(3,3) + 2G(3,4) - G(3,5)}{2} \\ B(3,4) &= \frac{b_{2,4} + b_{4,4}}{2} + \frac{-G(2,4) + 2G(3,4) - G(4,4)}{2} \\ R(4,3) &= \frac{r_{3,3} + r_{5,3}}{2} + \frac{-G(3,3) + 2G(4,3) - G(5,3)}{2} \\ B(4,3) &= \frac{b_{4,2} + b_{4,4}}{2} + \frac{-G(4,2) + 2G(4,3) - G(4,4)}{2}. \end{aligned}$$

At locations where the nearest chrominance neighbors are diagonally located, an adaptive procedure is employed. For example, in order to estimate the missing chrominance samples at (3, 3) and (4, 4), classifiers based on derivative estimates along the first and second diagonals are first estimated:

$$\begin{aligned} D^1_{3,3} &= |b_{2,2} - b_{4,4}| + |-G(2,2) + 2G(3,3) - G(4,4)| \\ D^2_{3,3} &= |b_{2,4} - b_{4,2}| + |-G(2,4) + 2G(3,3) - G(4,2)| \\ D^1_{4,4} &= |r_{3,3} - r_{5,5}| + |-G(3,3) + 2G(4,4) - G(5,5)| \\ D^2_{4,4} &= |r_{3,5} - r_{5,3}| + |-G(3,5) + 2G(4,4) - G(5,3)|. \end{aligned}$$

Then, using these classifiers, the missing red and blue values are given by the equation shown at the bottom of the next page.

F. Threshold-Based Variable Number of Gradients

Threshold-based variable number of gradients CFA interpolation [22] is a more complex demosaicking method. At each pixel location, eight gradient estimates are computed using samples from a 5×5 neighborhood of the CFA image. These estimates are computed slightly differently, depending on the the color sample at the pixel under consideration: chrominance (red and blue) or luminance (green). For example, the eight gradient estimates at location (3, 3) (Fig. 1) are given by

$$\begin{aligned}
N_{3,3} &= |g_{2,3} - g_{4,3}| + |r_{1,3} - r_{3,3}| + \frac{|b_{2,2} - b_{4,2}|}{2} \\
&\quad + \frac{|b_{2,4} - b_{4,4}|}{2} + \frac{|g_{1,2} - g_{3,2}|}{2} + \frac{|g_{1,4} - g_{3,4}|}{2} \\
E_{3,3} &= |g_{3,2} - g_{3,4}| + |r_{3,3} - r_{3,5}| + \frac{|b_{2,2} - b_{2,4}|}{2} \\
&\quad + \frac{|b_{4,2} - b_{4,4}|}{2} + \frac{|g_{2,3} - g_{2,5}|}{2} + \frac{|g_{4,3} - g_{4,5}|}{2} \\
S_{3,3} &= |g_{2,3} - g_{4,3}| + |r_{3,3} - r_{5,3}| + \frac{|b_{2,2} - b_{4,2}|}{2} \\
&\quad + \frac{|b_{2,4} - b_{4,4}|}{2} + \frac{|g_{3,2} - g_{5,2}|}{2} + \frac{|g_{3,4} - g_{5,4}|}{2} \\
W_{3,3} &= |g_{3,2} - g_{3,4}| + |r_{3,1} - r_{3,3}| + \frac{|b_{2,2} - b_{2,4}|}{2} \\
&\quad + \frac{|b_{4,2} - b_{4,4}|}{2} + \frac{|g_{2,1} - g_{2,3}|}{2} + \frac{|g_{4,1} - g_{4,3}|}{2} \\
NE_{3,3} &= |b_{2,4} - b_{4,2}| + |r_{1,5} - r_{3,3}| + \frac{|g_{2,3} - g_{3,2}|}{2} \\
&\quad + \frac{|g_{3,4} - g_{4,3}|}{2} + \frac{|g_{1,4} - g_{2,3}|}{2} + \frac{|g_{2,5} - g_{3,4}|}{2} \\
SE_{3,3} &= |b_{2,2} - b_{4,4}| + |r_{3,3} - r_{5,5}| + \frac{|g_{2,3} - g_{3,4}|}{2} \\
&\quad + \frac{|g_{3,2} - g_{4,3}|}{2} + \frac{|g_{3,4} - g_{4,5}|}{2} + \frac{|g_{4,3} - g_{5,4}|}{2} \\
NW_{3,3} &= |b_{2,2} - b_{4,4}| + |r_{1,1} - r_{3,3}| + \frac{|g_{1,2} - g_{2,3}|}{2} \\
&\quad + \frac{|g_{2,1} - g_{3,2}|}{2} + \frac{|g_{2,3} - g_{3,4}|}{2} + \frac{|g_{3,2} - g_{4,3}|}{2} \\
SW_{3,3} &= |b_{2,4} - b_{4,2}| + |r_{5,1} - r_{3,3}| + \frac{|g_{2,3} - g_{3,2}|}{2} \\
&\quad + \frac{|g_{3,4} - g_{4,3}|}{2} + \frac{|g_{3,2} - g_{4,1}|}{2} + \frac{|g_{4,3} - g_{5,2}|}{2}
\end{aligned}$$

while those at location (3, 4) are given by

$$\begin{aligned}
N_{3,4} &= |b_{2,4} - b_{4,4}| + |g_{1,4} - g_{3,4}| + \frac{|g_{2,3} - g_{4,3}|}{2} \\
&\quad + \frac{|g_{2,5} - g_{4,5}|}{2} + \frac{|r_{1,3} - r_{3,3}|}{2} + \frac{|r_{1,5} - r_{3,5}|}{2} \\
E_{3,4} &= |r_{3,3} - r_{3,5}| + |g_{3,4} - g_{3,6}| + \frac{|g_{2,3} - g_{2,5}|}{2} \\
&\quad + \frac{|g_{4,3} - g_{4,5}|}{2} + \frac{|b_{2,4} - b_{2,6}|}{2} + \frac{|b_{4,4} - b_{4,6}|}{2} \\
S_{3,4} &= |b_{2,4} - b_{4,4}| + |g_{3,4} - g_{5,4}| + \frac{|g_{2,3} - g_{4,3}|}{2} \\
&\quad + \frac{|g_{2,5} - g_{4,5}|}{2} + \frac{|r_{3,3} - r_{5,3}|}{2} + \frac{|r_{3,5} - r_{5,5}|}{2} \\
W_{3,4} &= |r_{3,3} - r_{3,5}| + |g_{3,2} - g_{3,4}| + \frac{|g_{2,3} - g_{2,5}|}{2} \\
&\quad + \frac{|g_{4,3} - g_{4,5}|}{2} + \frac{|b_{2,2} - b_{2,4}|}{2} + \frac{|b_{4,2} - b_{4,4}|}{2} \\
NE_{3,4} &= |g_{2,5} - g_{4,3}| + |g_{1,6} - g_{3,4}| + |r_{1,5} - r_{3,3}| \\
&\quad + |b_{2,6} - b_{4,4}| \\
SE_{3,4} &= |g_{2,3} - g_{4,5}| + |g_{3,4} - g_{5,6}| + |b_{2,4} - b_{4,6}| \\
&\quad + |r_{3,3} - r_{5,5}| \\
NW_{3,4} &= |g_{2,3} - g_{4,5}| + |g_{1,2} - g_{3,4}| + |b_{2,2} - b_{4,4}| \\
&\quad + |r_{1,3} - r_{3,5}| \\
SW_{3,4} &= |g_{2,5} - g_{4,3}| + |g_{3,4} - g_{5,2}| + |b_{2,4} - b_{4,2}| \\
&\quad + |r_{3,5} - r_{5,3}|.
\end{aligned}$$

Let $\mathcal{G}_{x,y}$ denote the set of gradient estimates at (x, y)

$$\begin{aligned}
\mathcal{G}_{x,y} = \{&N_{x,y}, S_{x,y}, E_{x,y}, W_{x,y}, NE_{x,y} \\
&NW_{x,y}, SE_{x,y}, SW_{x,y}\}. \quad (4)
\end{aligned}$$

Eight average red, green, and blue values (one for each gradient direction) are then computed at each pixel location. At location

$$B(3,3) = \begin{cases} \frac{b_{2,4}+b_{4,2}}{2} + \frac{-G(2,4)+2G(3,3)-G(4,2)}{2}, & D^1_{3,3} > D^2_{3,3} \\ \frac{b_{2,2}+b_{4,4}}{2} + \frac{-G(2,2)+2G(3,3)-G(4,4)}{2}, & D^1_{3,3} < D^2_{3,3} \\ \frac{b_{2,2}+b_{2,4}+b_{4,2}+b_{4,4}}{4} + \frac{-G(2,2)-G(2,4)+4G(3,3)-G(4,2)-G(4,4)}{4}, & D^1_{3,3} = D^2_{3,3} \end{cases}$$

$$R(4,4) = \begin{cases} \frac{r_{3,5}+r_{5,3}}{2} + \frac{-G(3,5)+2G(4,4)-G(5,3)}{2}, & D^1_{4,4} > D^2_{4,4} \\ \frac{r_{3,3}+r_{5,5}}{2} + \frac{-G(3,3)+2G(4,4)-G(5,5)}{2}, & D^1_{4,4} < D^2_{4,4} \\ \frac{r_{3,3}+r_{3,5}+r_{5,3}+r_{5,5}}{4} + \frac{-G(3,3)-G(3,5)+4G(4,4)-G(5,3)-G(5,5)}{4}, & D^1_{4,4} = D^2_{4,4}. \end{cases}$$

(3, 3), for example, the average values corresponding to the different gradient regions are given by

$$\begin{aligned}
R^N_{3,3} &= (r_{1,3} + r_{3,3})/2 \quad B^N_{3,3} = (b_{2,2} + b_{2,4})/2 \\
G^N_{3,3} &= g_{2,3} \\
R^E_{3,3} &= (r_{3,3} + r_{3,5})/2 \quad B^E_{3,3} = (b_{2,4} + b_{4,4})/2 \\
G^E_{3,3} &= g_{3,4} \\
R^S_{3,3} &= (r_{3,3} + r_{5,3})/2 \quad B^S_{3,3} = (b_{4,2} + b_{4,4})/2 \\
G^S_{3,3} &= g_{4,3} \\
R^W_{3,3} &= (r_{3,1} + r_{3,3})/2 \quad B^W_{3,3} = (b_{2,2} + b_{4,2})/2 \\
G^W_{3,3} &= g_{3,2} \\
R^{NE}_{3,3} &= (r_{1,5} + r_{3,3})/2 \quad B^{NE}_{3,3} = b_{2,4} \\
G^{NE}_{3,3} &= (g_{1,4} + g_{2,3} + g_{2,5} + g_{3,4})/4 \\
R^{SE}_{3,3} &= (r_{3,3} + r_{5,5})/2 \quad B^{SE}_{3,3} = b_{4,4} \\
G^{SE}_{3,3} &= (g_{3,4} + g_{4,3} + g_{4,5} + g_{5,4})/4 \\
R^{NW}_{3,3} &= (r_{1,1} + r_{3,3})/2 \quad B^{NW}_{3,3} = b_{2,2} \\
G^{NW}_{3,3} &= (g_{1,2} + g_{2,1} + g_{2,3} + g_{3,2})/4 \\
R^{SW}_{3,3} &= (r_{3,3} + r_{5,1})/2 \quad B^{SW}_{3,3} = b_{4,2} \\
G^{SW}_{3,3} &= (g_{3,2} + g_{4,1} + g_{4,3} + g_{5,2})/4.
\end{aligned}$$

From $\mathcal{G}_{x,y}$ [see (4)], a threshold $T_{x,y}$ is computed as $T_{x,y} = k_1 \min \mathcal{G}_{x,y} + k_2(\max \mathcal{G}_{x,y} - \min \mathcal{G}_{x,y})$, where the values of k_1, k_2 are empirically chosen as 1.5 and 0.5, respectively. Next, the previously computed color averages, from regions whose gradient estimate is less than the threshold, are averaged to yield $R^{\text{avg}}_{x,y}$, $G^{\text{avg}}_{x,y}$, and $B^{\text{avg}}_{x,y}$. For example, at location (3, 3), if the gradient estimates less than $T_{3,3}$ are $S_{3,3}, W_{3,3}, NE_{3,3}$, and $SW_{3,3}$, then

$$\begin{aligned}
R^{\text{avg}}_{3,3} &= 1/4 \cdot \left(R^S_{3,3} + R^W_{3,3} + R^{NE}_{3,3} + R^{SW}_{3,3}\right) \\
G^{\text{avg}}_{3,3} &= 1/4 \cdot \left(G^S_{3,3} + G^W_{3,3} + G^{NE}_{3,3} + G^{SW}_{3,3}\right) \\
B^{\text{avg}}_{3,3} &= 1/4 \cdot \left(B^S_{3,3} + B^W_{3,3} + B^{NE}_{3,3} + B^{SW}_{3,3}\right).
\end{aligned}$$

The final estimates of the missing color samples at location (3, 3) are then given by

$$\begin{aligned}
G(3,3) &= r_{3,3} + \left(G^{\text{avg}}_{3,3} - R^{\text{avg}}_{3,3}\right) \\
B(3,3) &= r_{3,3} + \left(B^{\text{avg}}_{3,3} - R^{\text{avg}}_{3,3}\right).
\end{aligned}$$

The missing color samples at the other pixel locations are estimated in a similar manner.

III. Detecting CFA Interpolation

At each pixel location of a CFA interpolated color image, a single color sample is captured by the camera sensor, whereas the other two colors are estimated from neighboring samples. As a result, a subset of samples, within a color channel, are correlated to their neighboring samples. Since the color filters in a CFA are typically arranged in a periodic pattern, these correlations are periodic. Consider, for example, the red channel $R(x, y)$ that has been sampled on a Bayer array (Fig. 1) and then CFA interpolated using bilinear interpolation; see Section II-A. In this case, the red samples in the odd rows and even columns are the average of their closest horizontal neighbors, the red samples in the even rows and odd columns are the average of their closest vertical neighbors, and the red samples in the even rows and columns are the average of their closest diagonal neighbors:

$$\begin{aligned}
R(2x+1, 2y) &= \frac{R(2x+1, 2y-1)}{2} + \frac{R(2x+1, 2y+1)}{2} \\
R(2x, 2y+1) &= \frac{R(2x-1, 2y+1)}{2} + \frac{R(2x+1, 2y+1)}{2} \\
R(2x, 2y) &= \frac{R(2x-1, 2y-1)}{4} + \frac{R(2x-1, 2y+1)}{4} \\
&\quad + \frac{R(2x+1, 2y-1)}{4} + \frac{R(2x+1, 2y+1)}{4}.
\end{aligned}$$

Note that in this simple case, the estimated samples are perfectly correlated to their neighbors. As such, a CFA-interpolated image can be detected (in the absence of noise) by noticing, for example, that every other sample in every other row or column is perfectly correlated to its neighbors. At the same time, the non-interpolated samples are less likely to be correlated in precisely the same manner. Furthermore, it is likely that tampering will destroy these correlations or that the splicing together of two images from different cameras will create inconsistent correlations across the composite image. As such, the presence or lack of correlations produced by CFA interpolation can be used to authenticate an image or expose it as a forgery.

We begin by assuming a simple linear model for the periodic correlations introduced by CFA interpolation. That is, each interpolated pixel is correlated to a weighted sum of pixels in a small neighborhood centered about itself. While perhaps overly simplistic when compared to the highly nonlinear nature of most CFA interpolation algorithms, this simple model is both easy to parameterize and can reasonably approximate each of the CFA interpolation algorithms presented in Section II. Note that most CFA algorithms estimate a missing color sample from neighboring samples in all three color channels. For simplicity, however, we ignore these interchannel correlations and treat each color channel independently.

If the specific form of the correlations is known (i.e., the parameters of the linear model), then it would be straightforward to determine which samples are correlated to their neighbors. On the other hand, if it was known which samples are correlated to their neighbors, the specific form of the correlations could be easily determined. In practice, of course, neither are known. To simultaneously estimate both, we employ the expectation/maximization (EM) algorithm [23], as described below.

A. Expectation/Maximization Algorithm

Let $f(x, y)$ denote a color channel (red, green, or blue) of a CFA interpolated image. We begin by assuming that each sample in $f(x, y)$ belongs to one of two models: 1) M_1 if the sample is linearly correlated to its neighbors, satisfying

$$f(x,y) = \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) + n(x,y) \qquad (5)$$

where the model parameters are given by the linear coefficients $\vec{\alpha} = \{\alpha_{u,v} \,|\, -N \leq u, v \leq N\}$ (N is an integer and $\alpha_{0,0} = 0$), and $n(x,y)$ denotes independent and identically distributed samples drawn from a Gaussian distribution with zero mean and unknown variance σ^2; 2) M_2 if the sample is not correlated to its neighbors, i.e., is generated by an "outlier process."

The EM algorithm is a two-step iterative algorithm: 1) In the E-step, the probability of each sample belonging to each model is estimated, and 2) in the M-step, the specific form of the correlations between samples is estimated. More specifically, in the E-step, the probability of each sample of $f(x,y)$ belonging to model M_1 is estimated using Bayes' rule:

$$\Pr\{f(x,y) \in M_1 \,|\, f(x,y)\} = \frac{\Pr\{f(x,y) \,|\, f(x,y) \in M_1\}\Pr\{f(x,y) \in M_1\}}{\sum_{i=1}^{2} \Pr\{f(x,y) \,|\, f(x,y) \in M_i\}\Pr\{f(x,y) \in M_i\}} \quad (6)$$

where the prior probabilities $\Pr\{f(x,y) \in M_1\}$ and $\Pr\{f(x,y) \in M_2\}$ are assumed to be equal to $1/2$. The probability of observing a sample $f(x,y)$, knowing it was generated from model M_1, is given by

$$\Pr\{f(x,y) \,|\, f(x,y) \in M_1\} = \frac{1}{\sigma\sqrt{2\pi}} \times \exp\left[-\frac{\left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v)\right)^2}{2\sigma^2}\right].$$

The variance σ^2 of this Gaussian distribution is estimated in the M-step (see the Appendix). A uniform distribution is assumed for the probability of observing a sample generated by the outlier model M_2, i.e., $\Pr\{f(x,y) \,|\, f(x,y) \in M_2\}$ is equal to the inverse of the range of possible values of $f(x,y)$. Note that the E-step requires an estimate of the coefficients $\vec{\alpha}$, which on the first iteration is chosen randomly. In the M-step, a new estimate of $\vec{\alpha}$ is computed using weighted least squares by minimizing the following quadratic error function:

$$E(\vec{\alpha}) = \sum_{x,y} w(x,y) \cdot \left(f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v)\right)^2$$

where the weights $w(x,y) \equiv \Pr\{f(x,y) \in M_1 \,|\, f(x,y)\}$; see (6). This error function is minimized by computing the gradient with respect to $\vec{\alpha}$, setting this gradient equal to zero, and solving the resulting linear system of equations. Setting equal to zero the partial derivative with respect to one of the coefficients $\alpha_{s,t}$ yields

$$\frac{\partial E}{\partial \alpha_{s,t}} = 0$$

$$\sum_{x,y} w(x,y) f(x+s, y+t) \sum_{u,v=-N}^{N} \alpha_{u,v} f(x+u, y+v) = \sum_{x,y} w(x,y) f(x+s, y+t) f(x,y).$$

Reordering the terms on the left-hand side yields

$$\sum_{u,v=-N}^{N} \alpha_{u,v} \left(\sum_{x,y} w(x,y) f(x+s, y+t) f(x+u, y+v)\right) = \sum_{x,y} w(x,y) f(x+s, y+t) f(x,y). \quad (7)$$

This process is repeated for each component $\alpha_{s,t}$ of $\vec{\alpha}$ to yield a system of linear equations that can be solved using standard techniques. The E-step and the M-step are iteratively executed until a stable estimate of $\vec{\alpha}$ is achieved. The final $\vec{\alpha}$ has the property that it maximizes the likelihood of the observed samples.[3] See the Appendix for a detailed algorithm.

IV. Results

We collected 100 images: 50 of resolution 512×512 and 50 of resolution 1024×1024. Each of these images were cropped from a smaller set of 20 1600×1200 images taken with a Nikon Coolpix 950 camera and 20 3034×2024 images taken with a Nikon D100 camera. The Nikon Coolpix 950 employs a four-filter (yellow, cyan, magenta, green) CFA and was set to store the images in uncompressed TIFF format. The Nikon D100 camera employs a Bayer array and was set to store the images in RAW format. To avoid interference with the CFA interpolation of the cameras, each color channel of these images was independently blurred with a 3×3 binomial filter and downsampled by a factor of two in each direction. These downsampled color images, of size 256×256 or 512×512, were then resampled onto a Bayer array and CFA interpolated using the algorithms described in Section II.

The results of running the EM algorithm (Section III-A) on eight 256×256 color images that were CFA interpolated using the algorithms presented in Section II are shown in Figs. 2 and 3. The parameters of the EM algorithm were $N = 1, \sigma_0 = 0.0075$, and $p_0 = 1/256$. Shown in the left column are the images, in the middle column the estimated probability maps from the green channel (the red and blue channels yield similar results), and in the right column the magnitude of the Fourier transforms of the probability maps.[4] Note that although the periodic patterns in the probability maps are not visible at this reduced scale, they are particularly salient in the Fourier domain in the form of localized peaks. Note also that these peaks do not appear in the images that are not CFA interpolated (last row of Figs. 2 and 3).

While the probability maps obtained from the EM algorithm can be employed to detect if a color image is the result of a CFA interpolation algorithm, the linear coefficients $\vec{\alpha}$ returned by the EM algorithm can be used to distinguish between different CFA interpolation techniques. Let $\vec{\alpha}$ denote the 24-D vector of linear

[3]The EM algorithm has been proven to always converge to a stable estimate that is a local maximum of the log-likelihood of the observed data [23].

[4]For display purposes, the probability maps were upsampled by a factor of two before Fourier transforming. The periodic patterns introduced by CFA interpolation have energy in the highest horizontal, vertical, and diagonal frequencies, which corresponds to localized frequency peaks adjacent to the image edges. Upsampling by a factor of two shrinks the support of a probability map's spectrum and shifts these peaks into the midfrequencies, where they are easier to see. In addition, for display purposes, the Fourier transforms of the probability maps were highpass filtered, blurred, scaled to fill the range [0, 1], and gamma corrected with an exponent of 2.0.

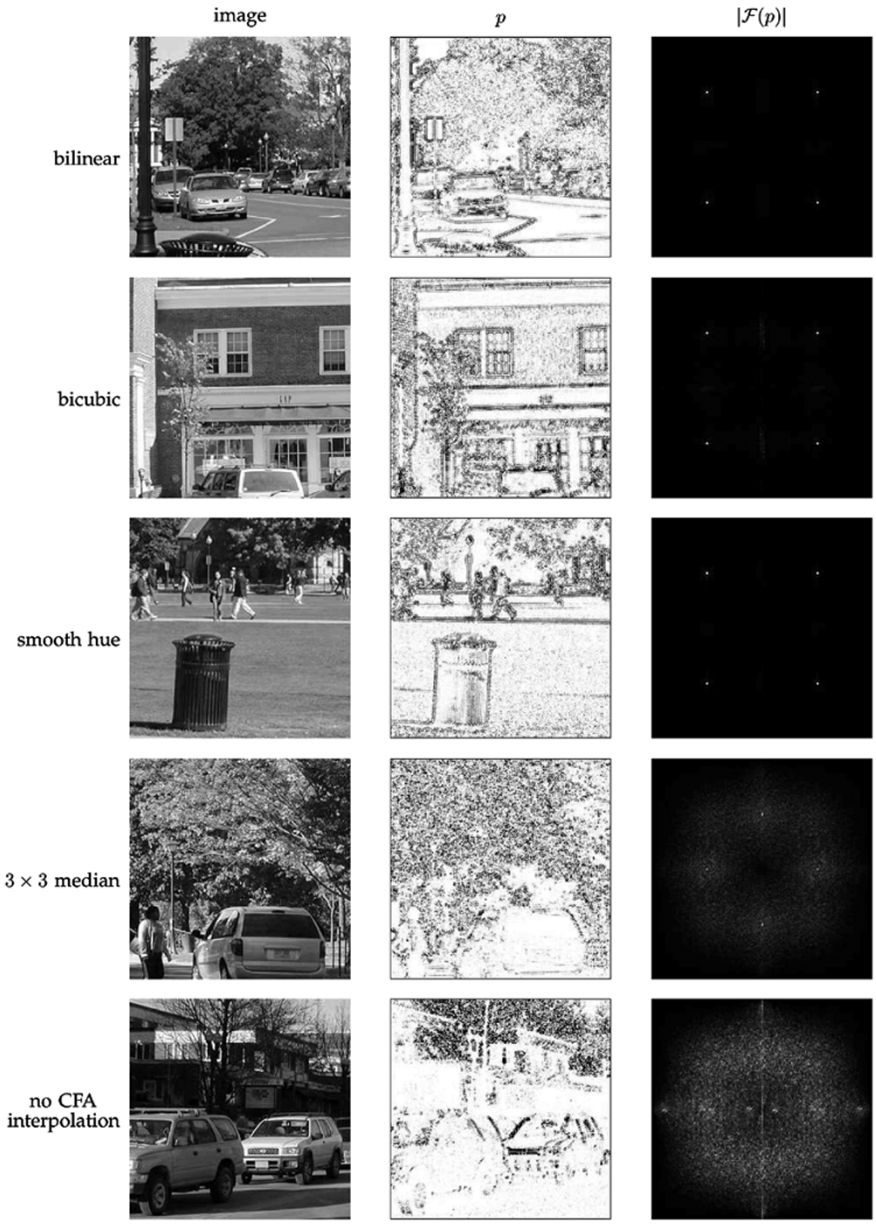

Fig. 2. Image interpolated with the specified algorithm, the probability map of the green channel as output by the EM algorithm, and the magnitude of the Fourier transform of the probability map is shown in each row. Note the peaks in $|\mathcal{F}(p)|$ corresponding to periodic correlations in the CFA interpolated images and the lack of such peaks in the non-CFA interpolated image (last row).

coefficients (i.e., with a neighborhood size of $N = 1$, each of three color channels has eight coefficients). Using principal component analysis (PCA) [24], we computed the principal axes of 800, 24-D vectors obtained from running the EM algorithm on 100 images CFA interpolated with each of eight algorithms described in Section II (two versions of the median filter algorithm were used). The projections onto the first two principal axes are shown in Fig. 4. Note how points corresponding to different CFA interpolation algorithms tend to cluster. Even in this 2-D space, these clusters are, in some cases, clearly separable.

While the different interpolation methods are not perfectly separable in this low-dimensional space, they are pairwise separable in their original 24-D space. Specifically, a Linear Discriminant Analysis (LDA) [25] on all pairs of interpolation algorithms yields nearly perfect separation. For each of 28 pairs of algorithms, we trained an LDA classifier. Using 90% of the data for training and 10% for testing, the average testing accuracy over 10 random training/testing splits and over all pairs of algorithms was 97%. The minimum testing accuracy was 88% and was obtained when separating the 5×5 median filter and the adaptive color plane algorithms. Since digital cameras typically employ different CFA interpolation algorithms, this result could be used to determine if an image was *not* taken with a specific digital camera.

A. Detecting Localized Tampering

Since it is likely that tampering will destroy the periodicity of the CFA correlations, it may be possible to detect and localize tampering in any portion of an image. To illustrate this, consider the left-most image in Fig. 5, taken with a Nikon D100 digital

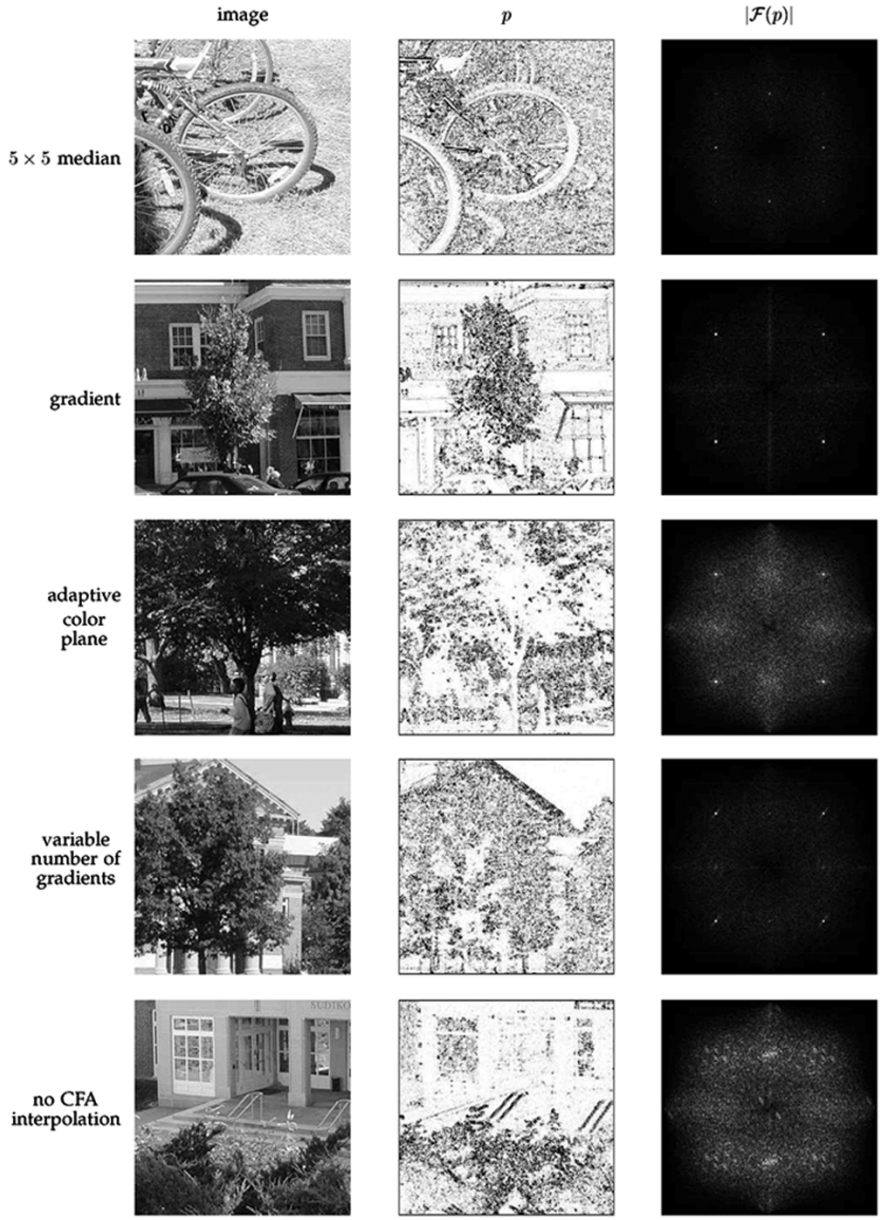

Fig. 3. Image interpolated with the specified algorithm, the probability map of the green channel as output by the EM algorithm, and the magnitude of the Fourier transform of the probability map is shown in each row. Note the peaks in $|\mathcal{F}(p)|$ corresponding to periodic correlations in the CFA interpolated images and the lack of such peaks in the non-CFA interpolated image (last row).

camera and saved in RAW format. Each color channel of this image (initially interpolated using the adaptive color plane technique) was blurred with a 3×3 binomial filter and downsampled by a factor of two in order to destroy the CFA periodic correlations. The 512×512 downsampled image was then resampled on a Bayer array and CFA interpolated. Next, composite images, which were 512×512 pixels in size, were created by splicing, in varying proportions ($1/4$, $1/2$, and $3/4$), the non-CFA interpolated image and the same image CFA interpolated with the bicubic algorithm; see Section II-A. The probability maps obtained from running EM on the red channel of the composite images are shown in Fig. 5. Notice that these probability maps clearly reveal the presence of two distinct regions. The magnitudes of the Fourier transforms of two windows—one from the non-CFA interpolated portion (right) and one from the CFA interpolated portion (left)—are shown below each probability map.[5] Notice the presence of localized frequency peaks in the CFA interpolated portion and the lack of such peaks in the non-CFA interpolated portion.

A perceptually plausible forgery created using Adobe Photoshop (top row, right) is shown in Fig. 6. The tampering consisted of hiding the damage on the car hood using air-brushing, smudging, blurring, and duplication. The original image CFA interpolated using bicubic interpolation (top row, left), the estimated probability map of the tampered image (middle row), and the magnitude of the Fourier transforms of two windows [one from a tampered portion (bottom row, left)

[5]For display purposes, the probability maps were upsampled by a factor of two, and their Fourier transforms were highpass filtered, blurred, scaled to fill the range [0, 1], and gamma corrected with an exponent of 2.0.

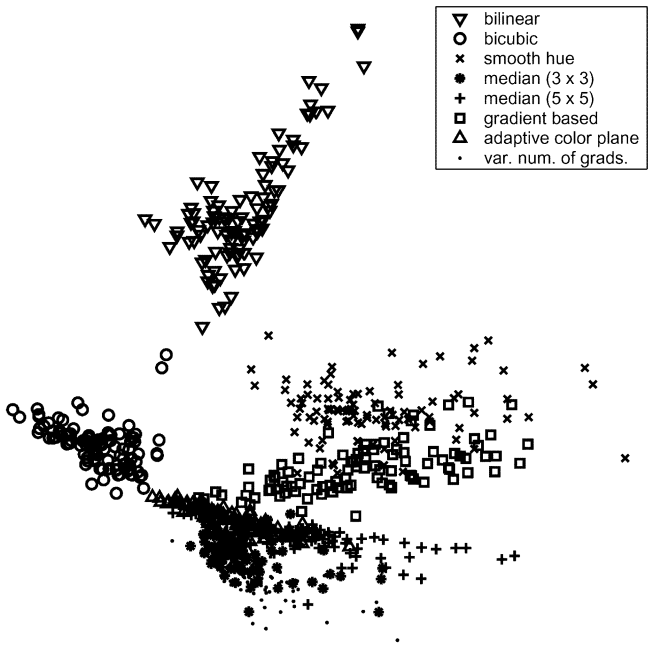

Fig. 4. Estimated interpolation coefficients from 800 CFA-o is interpolated images, with eight different algorithms, are projected onto a 2-D subspace. Even in this reduced space, the algorithms are, in some cases, clearly separable.

and one from an unadulterated portion (bottom row, right)] are also shown. Note that even though the periodic pattern is not visible in the probability map, localized frequency peaks reveal its presence. Note also that the window from the tampered region does not contain frequency peaks.

B. Sensitivity and Robustness

From a digital forensics point of view, it is important to quantify the robustness and sensitivity of our detection technique. It is therefore necessary to devise a quantitative measure for the periodic correlations introduced by CFA interpolation. This is achieved by comparing the estimated probability maps of each color channel of a given image with synthetically generated probability maps. A synthetic map is generated for each color channel as follows:

$$s_r(x,y) = \begin{cases} 0, & S(x,y) = r_{x,y} \\ 1, & \text{otherwise} \end{cases}$$

$$s_g(x,y) = \begin{cases} 0, & S(x,y) = g_{x,y} \\ 1, & \text{otherwise} \end{cases}$$

$$s_b(x,y) = \begin{cases} 0, & S(x,y) = b_{x,y} \\ 1, & \text{otherwise} \end{cases}$$

where $S(x,y)$ is defined as in Section II. Let $p_g(x,y)$ denote the probability map obtained from the green channel of an image. A similarity measure between $p_g(x,y)$ and $s_g(x,y)$ is computed as follows.

1) The probability map $p_g(\cdot)$ is Fourier transformed: $P_g(\omega_x,\omega_y) = \mathcal{F}(p_g(x,y))$.
2) The synthetic map $s_g(\cdot)$ is Fourier transformed: $S_g(\omega_x,\omega_y) = \mathcal{F}(s_g(x,y))$.
3) The measure of similarity between $p_g(x,y)$ and $s_g(x,y)$ is then given by

$$M(p_g,s_g) = \sum_{\omega_x,\omega_y} |P_g(\omega_x,\omega_y)| \cdot |S_g(\omega_x,\omega_y)|$$

where $|\cdot|$ denotes absolute value (note that this similarity measure is phase insensitive).

If the similarity measure $M(p_g,s_g)$ between $p_g(x,y)$ and $s_g(x,y)$ is above a specified threshold, then a periodic pattern is assumed to be present in the probability map, and the channel is labeled as CFA interpolated. Thresholds are empirically determined for each channel to yield a 0% false positive rate, i.e., no color channel from the non-CFA interpolated images is classified as CFA interpolated. The similarity measures between the probability and synthetic maps of the red and blue channels $M(p_r,s_r)$ and $M(p_b,s_b)$ are computed in a similar way.

To simulate tampering, each image is blurred and downsampled in order to destroy the original CFA interpolation correlations. These images are then interpolated with each of the algorithms described in Section II. Given a three-channel color image, we first compute the similarity measures between the probability maps of each color channel and the corresponding synthetic maps. The image is labeled CFA interpolated if at least one of the three similarities is greater than its specified threshold or labeled tampered if all three similarities are less than the thresholds. Using this approach, we have obtained, with a 0% false positive rate, the following classification accuracies averaged over 100 images: bilinear: 100%, bicubic: 100%, smooth hue: 100%, median (3×3): 99% median (5×5): 97%, gradient-based: 100%, adaptive color plane: 97%, and variable number of gradients: 100%.

To be useful in a forensic setting, it is important for our detection method to be robust to simple image distortions. We have tested the sensitivity of our method to typical distortions that may conceal traces of tampering: 1) JPEG compression, 2) additive white Gaussian noise, and 3) nonlinear pointwise gamma correction. Fifty images taken with the Nikon Coolpix 950 camera were processed as described above in order to obtain CFA and non-CFA interpolated images. The detection accuracies (with 0% false positives) for different CFA interpolation algorithms as a function of the JPEG compression quality are shown in Fig. 7. Note that these detection accuracies are close to 100% for quality factors greater than 96 (out of 100) and that they decrease gracefully with decreasing quality factors. This decrease in accuracy is expected since the lossy JPEG compression introduces noise that destroys the correlations introduced by CFA interpolation. The decrease in accuracy depends on the CFA interpolation algorithm, e.g., the accuracy for smooth hue CFA interpolation is 56% at quality 70, whereas the accuracy for adaptive color plane CFA interpolation drops to 6%.

The detection accuracies (with 0% false positives) for different CFA interpolation algorithms as a function of the signal-to-noise ratio (SNR) of additive white Gaussian noise are shown in Fig. 8. In this case, the accuracies drop relatively slowly as the SNR decreases. For example, very good detection rates are achieved for even nonlinear methods: With an SNR of

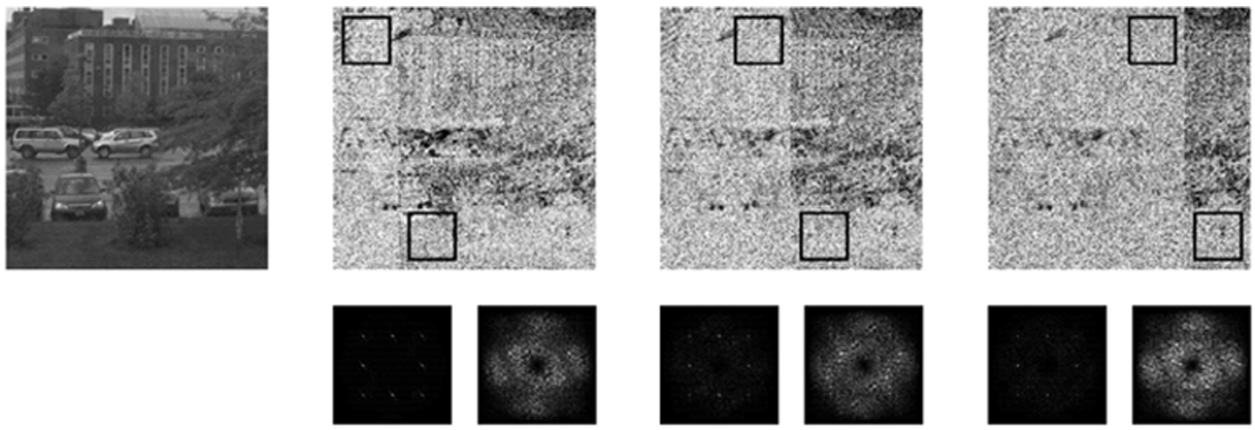

Fig. 5. Image and the probability maps of composite images obtained by splicing together the CFA interpolated image (left portion of each image) and the same image without CFA interpolation (right portion). The magnitudes of the Fourier transforms of windows from the two regions are also shown. Note that windows from the CFA interpolated regions (left) have localized peaks in the Fourier domain, whereas the windows from the "tampered" regions (right) do not.

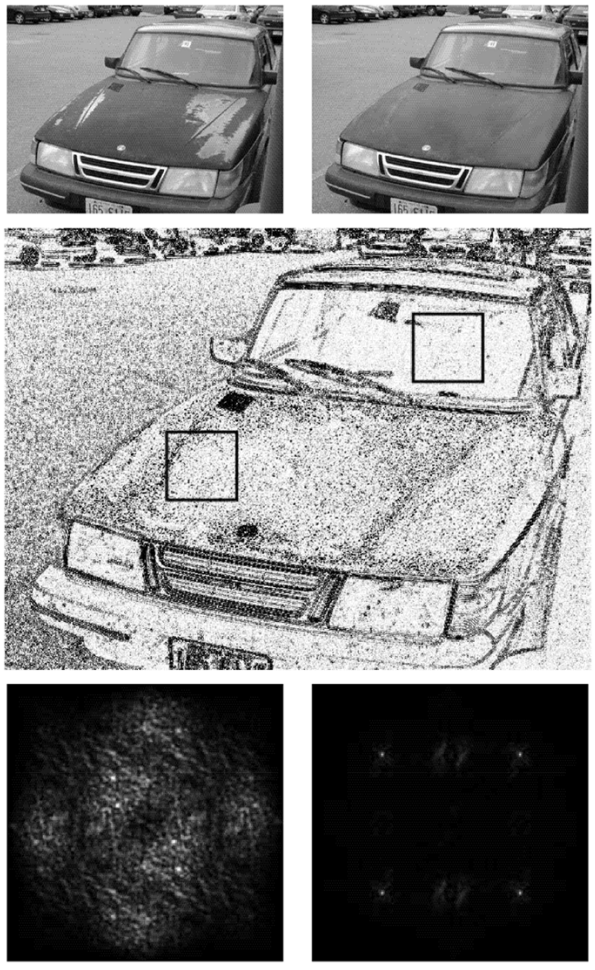

Fig. 6. (Top) Original and a tampered image. (Middle) Probability map of the tampered image's green channel. (Bottom) Magnitude of the Fourier transform of two windows from the probability map. The windows correspond to a tampered and an unadulterated portions of the forgery. Note the lack of peaks in the tampered region signifying the absence of CFA interpolation.

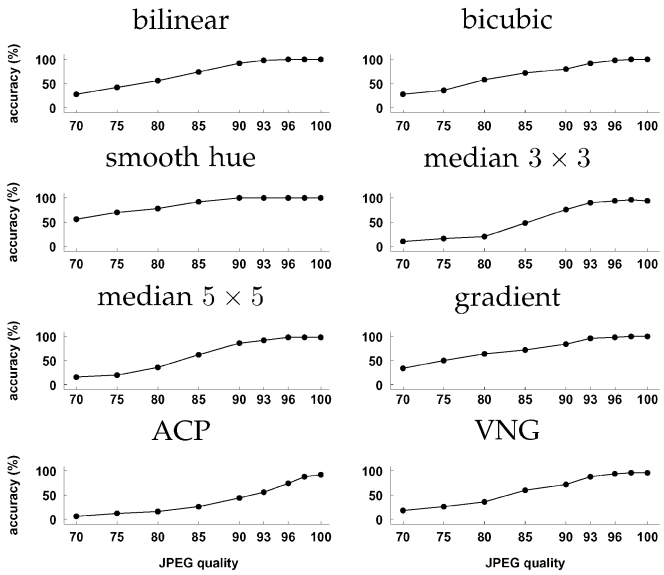

Fig. 7. Detection accuracies (with 0% false positives) for different CFA interpolation algorithms as a function of JPEG compression quality. Each data point corresponds to the average detection accuracy over 50 images.

18 db, 76% of the adaptive color plane and 86% of the variable number of gradients interpolated images are correctly detected.

We have also tested our detection method on CFA-interpolated images that have been gamma corrected with exponents ranging from 0.5 to 1.5, in steps of 0.1. For all CFA-interpolation techniques the accuracies (with 0% false positives) were either 100% or 98% (i.e., one image out of 50 misclassified).

In summary, we have shown that our detection method can reasonably distinguish between CFA and non-CFA interpolated images, even when the images are subjected to JPEG compression, additive noise, or luminance nonlinearities.

C. Commercial Digital Cameras

In the previous sections, the efficacy of our detection method was tested against CFA interpolated images of our creation. Here, we show the efficacy of our method on images acquired from three commercially available digital cameras: Canon Powershot A80 (3.8 megapixels), Nikon D100 (6.1 megapixels), and Kodak DCS 14 N (14 megapixels). Each camera employs an RGB Bayer CFA. The Canon camera was set to capture in maximum-quality JPEG format, and the Nikon and Kodak cameras

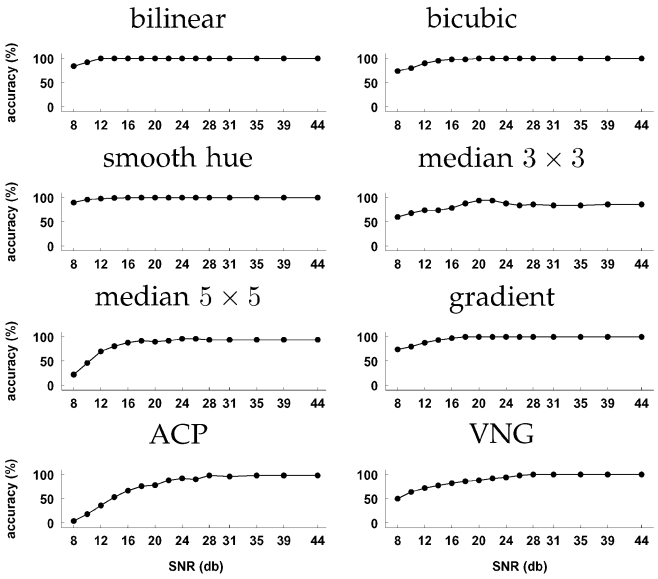

Fig. 8. Detection accuracies (with 0% false positives) for different CFA interpolation algorithms as a function of signal-to-noise ratio of additive white Gaussian noise. Each data point corresponds to the average detection accuracy over 50 images.

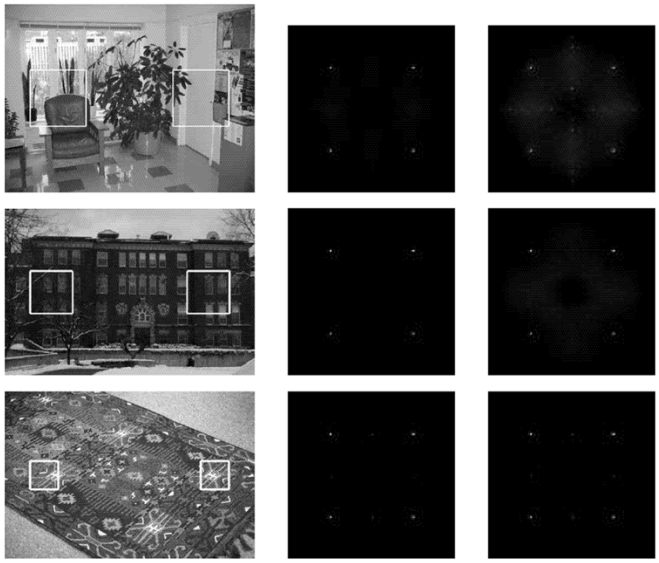

Fig. 9. Three images captured with commercially available Canon, Nikon, and Kodak digital cameras (top to bottom, respectively). Magnitudes of the Fourier transforms of the estimated probability map of 512×512 windows (green channel) are also shown. The peaks in the Fourier transform correspond to the presence of CFA interpolation in the image. These patterns are present in all regions of the image.

were set to capture in uncompressed TIFF format. Nine images were captured (three images per camera).

Each of these images were divided into windows of sizes 256×256 or 512×512, with an overlap of 50% along each direction. This led to, respectively, 192 and 35 windows for the Canon camera, 308 and 60 windows for the Nikon camera, and 748 and 160 windows for the Kodak camera. For each window, the CFA detection algorithm was applied to each color channel to yield three probability maps. The color channels were labeled CFA, or non-CFA, interpolated if the similarity measures between their probability maps and corresponding synthetic maps were above, or below, empirically determined thresholds (the same thresholds, as those used in previous sections, were used). As before, a window was labeled authentic if at least one color channel was labeled as CFA interpolated and labeled tampered if all three color channels were labeled as non-CFA interpolated. With a window size of 256×256, all windows for all three cameras were correctly classified as authentic, with the exception of one of the Kodak images, in which two of the 748 windows were incorrectly classified as tampered. With a window size of 512×512, all windows for all three cameras were correctly classified as authentic. Sample images from each of the three digital cameras are shown in Fig. 9. Also shown is the magnitude of the Fourier transform of the estimated probability map (green channel) from 512×512 windows. Note the presence of the localized peaks that correspond to CFA interpolation. Although difficult to see, the Canon images (captured in JPEG format) contain small peaks along the horizontal and vertical axes. These peaks correspond to the presence of JPEG blocking artifacts but do not interfere with the detection of CFA interpolation.

V. Discussion

Most digital cameras employ a single sensor in conjunction with a color filter array (CFA), where the missing color samples are then interpolated from these recorded samples to obtain a three-channel color image. This interpolation introduces specific correlations that are likely to be destroyed when tampering with an image. As such, the presence or lack of correlations produced by CFA interpolation can be used to authenticate an image or expose it as a forgery. We have shown, for eight different CFA interpolation algorithms, that a simple linear model captures these correlations. We have also shown the efficacy of this approach to detecting traces of digital tampering in lossless and lossy compressed images.

Our technique works in the absence of any digital watermark or signature, offering a complementary approach to authenticating digital images. While statistical techniques such as those presented here pose many challenges, we believe that their development will be important to contend with the cases when watermarking technologies are not applicable.

As with any authentication scheme, our approach is vulnerable to counterattack. A tampered image could, for example, be resampled onto a CFA, and then reinterpolated. This attack, however, requires knowledge of the camera's CFA pattern and interpolation algorithm and may be beyond the reach of a novice forger.

We are currently exploring several other techniques for detecting other forms of digital tampering. We believe that many complementary techniques such as that presented here, and those that we [5], [8]–[10] and others [4], [6], [7] develop, will be needed to reliably expose digital forgeries. There is little doubt that even with the development of a suite of detection techniques, more sophisticated tampering techniques will emerge, which, in turn, will lead to the development of more detection tools, and so on, thus making the creation of forgeries increasingly more difficult.

APPENDIX
EM ALGORITHM

/* Initialize */

choose $\{\alpha_{u,v}^{(0)}\}$ randomly

choose N and σ_0

set p_0 as 1 over the size of the range of values of $f(x,y)$

$n = 0$

repeat

/* expectation step */

for each sample location (x,y)

$$r(x,y) = \left| f(x,y) - \sum_{u,v=-N}^{N} \alpha_{u,v}^{(n)} f(x+u, y+v) \right|$$

end

for each sample location (x,y)

$$P(x,y) = \frac{1}{\sigma_n \sqrt{2\pi}} \exp\left[-r^2(x,y)/2\sigma_n^2\right]$$
$$w(x,y) = \frac{P(x,y)}{P(x,y) + p_0}$$

end

/* maximization step */

compute $\{\alpha_{u,v}^{(n+1)}\}$ from the linear system in (7)

$$\sigma_{n+1} = \left(\frac{\sum_{x,y} w(x,y) r^2(x,y)}{\sum_{x,y} w(x,y)} \right)^{1/2}$$
$$n = n + 1$$

until $(\sum_{u,v} |\alpha_{u,v}^{(n)} - \alpha_{u,v}^{(n-1)}| < \epsilon)$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers from an earlier paper [8] for suggesting that we consider the effects of resampling on color filter array-interpolated images.

REFERENCES

[1] S. Katzenbeisser and F. Petitcolas, *Information Techniques for Steganography and Digital Watermarking*. Norwell, MA: Artech House, 2000.

[2] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*. San Francisco, CA: Morgan Kaufmann, 2002.

[3] S. Craver, M. Wu, B. Liu, A. Stubblefield, B. Swartzlander, and D. Wallach, "Reading between the lines: Lessons from the SDMI challenge," in *Proc. 10th USENIX Security Symp.*, Washington, DC, 2001.

[4] J. Fridrich, D. Soukal, and J. Lukáš, "Detection of copy-move forgery in digital images," in *Proc. Digital Forensic Res. Workshop*, Aug. 2003.

[5] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," in *Proc. IEEE Workshop Statist. Anal. Comput. Vision*, Madison, WI, 2003.

[6] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2004.

[7] I. Avcibas, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A classifier design for detecting image manipulations," in *Proc. Int. Conf. Image Process.*, Oct. 2004.

[8] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[9] ——, "Statistical tools for digital forensics," in *Proc. Sixth Inf. Hiding Workshop*, May 2004.

[10] ——, "Exposing Digital Forgeries by Detecting Duplicated Image Regions," Dartmouth College, Hanover, NH, Tech. Rep. TR2004-515, Aug. 2004.

[11] A. C. Popescu, "Statistical Tools for Digital Image Forensics," Ph.D. Dissertation, Dartmouth College, Hanover, NH, Dec. 2004.

[12] B. E. Bayer, "Color Imaging Array," US Patent, 3 971 065, 1976.

[13] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, "Demosaicking methods for Bayer color arrays," *J. Electron. Imag.*, vol. 11, no. 3, pp. 306–315, Jul. 2002.

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

[15] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," in *Proc. Int. Conf. Image Process.*, vol. 3, Sept. 2003, pp. 669–672.

[16] D. D. Muresan and T. W. Parks, "Adaptively quadratic (AQua) image interpolation," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 690–698, May 2004.

[17] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[18] D. R. Cok, "Signal Processing Method and Apparatus for Producing Interpolated Chrominance Values in a Sampled Color Image Signal," US Patent, 4 642 678, 1987.

[19] W. T. Freeman, "Median Filter for Reconstructing Missing Color Samples," US Patent, 4 724 395, 1988.

[20] C. A. Laroche and M. A. Prescott, "Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Chrominance Gradients," US Patent, 5 373 322, 1994.

[21] J. F. Hamilton and J. E. Adams, "Adaptive Color Plan Interpolation in Single Sensor Color Electronic Camera," US Patent, 5 629 734, 1997.

[22] E. Chang, S. Cheung, and D. Y. Pan, "Color filter array recovery using a threshold-based variable number of gradients," in *Proc. SPIE Sensors, Cameras, Applicat. Digit. Photography*, vol. 3650, N. Sampat and T. Yeh, Eds., Mar. 1999, pp. 36–43.

[23] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 99, no. 1, pp. 1–38, 1977.

[24] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[25] R. Fisher, "The use of multiple measures in taxonomic problems," *Ann. Eugen.*, vol. 7, pp. 179–188, 1936.

Alin C. Popescu received the B.E. degree in electrical engineering in 1999 from the University Politehnica of Bucharest, Bucharest, Romania, the M.S. degree in computer science in 1999 from Université de Marne-la-Vallée, Marne-la-Vallée, France, and the Ph.D. in computer science in 2005 from Dartmouth College, Hanover, NH.

Hany Farid received the B.S. degree in computer science and applied mathematics in 1988 from the University of Rochester, Rochester, NY, and the Ph.D. degree in 1997 in computer science from the University of Pennsylvania, Philadelphia.

He joined the faculty at Dartmouth College, Hanover, NH, in 1999, following a two-year post-doctoral position with the Brain and Cognitive Sciences Department, Massachusetts Institute of Technology, Cambridge.