Gradient-Based Feature Extraction From Raw Bayer Pattern Images

Wei Zhou, *Graduate Student Member, IEEE*, Ling Zhang, Shengyu Gao, *Graduate Student Member, IEEE*, and Xin Lou, *Member, IEEE*

***Abstract*— In this paper, the impact of demosaicing on gradient extraction is studied and a gradient-based feature extraction pipeline based on raw Bayer pattern images is proposed. It is shown both theoretically and experimentally that the Bayer pattern images are applicable to the central difference gradient-based feature extraction algorithms with negligible performance degradation, as long as the arrangement of color filter array (CFA) patterns matches the gradient operators. The color difference constancy assumption, which is widely used in various demosaicing algorithms, is applied in the proposed Bayer pattern image-based gradient extraction pipeline. Experimental results show that the gradients extracted from Bayer pattern images are robust enough to be used in histogram of oriented gradients (HOG)-based pedestrian detection algorithms and shift-invariant feature transform (SIFT)-based matching algorithms. By skipping most of the steps in the image signal processing (ISP) pipeline, the computational complexity and power consumption of a computer vision system can be reduced significantly.**

***Index Terms*— Gradient, Bayer pattern image, feature extraction, demosaicing.**

I. Introduction

COMPUTER vision studies how to extract useful information from digital images and videos to obtain high-level understanding. As an indispensable component, image sensors convert the outside world scene to digital images that are consumed by computer vision algorithms. To produce color images, the information from three channels, i.e., red (R), green (G) and blue (B), are needed. There are two primary technology families used in today's color cameras: the mono-sensor technique and the three-sensor technique. Although three-sensor cameras are able to produce high-quality color images, their popularity is limited by the high manufacturing cost and large size [2]. As an alternative, the mono-sensor technique is employed in most of the digital color cameras and smartphones nowadays. In a mono-sensor color camera, images are captured with one sensor covered by a color filter array (CFA), e.g., the Bayer pattern [3] shown in Fig. 1(a), such that only one out of three color components is captured by each pixel element. This single channel image is converted to a color image by interpolating the other two missing color components at each pixel. This process is referred to as demosaicing, which is a fundamental step in the traditional image signal processing (ISP) pipeline. Apart from the demosaicing step, other ISP stages are usually determined by the manufacturers according to the application scenarios [4].

Almost all the existing computer vision algorithms take images processed by the ISP pipeline as inputs. However, the existing ISP pipelines are designed for photography with a goal of generating high-quality images for human consumption. Although pleasing scenes can be produced, no additional information is put in by the ISP. In addition, it has been shown that the ISP pipeline may introduce cumulative errors and undermine the original information from image sensors [5]. For example, as the demosaicing process smoothes the image, the information entropy of the image decreases [6]. Moreover, it has been shown that ISP algorithms are computation intensive and consume a significant portion of processing time and power in a computer vision system [7], [8]. Profiling statistics of major steps in an ISP pipeline was presented in [8], which show that the demosaicing step involves a lot of memory access (which may be a bottleneck) and the denoising steps consumes more computation than others (see supplemental material). If certain ISP steps are not necessary, we can skip them to reduce the computational complexity and power consumption of the system. Therefore, for computer vision applications, the configuration or even the necessity of the complete ISP pipeline needs to be reconsidered.

The optimal configuration of the ISP pipeline for different computer vision applications remains an open problem [8]–[10]. In a recent paper, Buckler et. al. use an empirical approach to study the ISP's impact on different vision applications [8]. Extensive experiments based on eight existing vision algorithms are conducted and a minimal ISP pipeline consisting of denoise, demosaicing and gamma compression is proposed. But all the conclusions in [8] are drawn based on experimental results without detailed theoretical analysis.

Manuscript received May 27, 2020; revised September 26, 2020, December 13, 2020, and February 10, 2021; accepted March 9, 2021. Date of publication May 20, 2021; date of current version May 24, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 61801292 and in part by the Shanghai Sailing Program under Grant 18YF1416600. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ioannis Kompatsiaris. *(Corresponding author: Xin Lou.)*

Wei Zhou is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China, also with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200031, China, and also with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China.

Ling Zhang, Shengyu Gao, and Xin Lou are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: louxin@shanghaitech.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3067166, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3067166

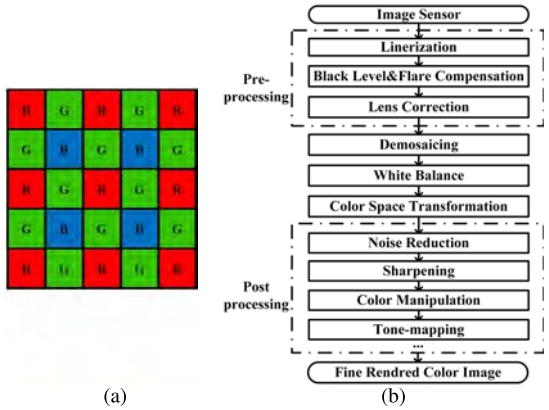


Fig. 1. (a) The RGGB Bayer CFA pattern. (b) The conventional ISP pipeline.

There are also some studies that try to bypass the traditional ISP and extract the high-level global features such as edge and local binary pattern (LBP), from Bayer pattern images [11]–[13]. Moreover, it is experimentally shown in [14] and [15] that the Bayer pattern images can be applied directly in some local feature descriptors such as scale-invariant feature transform (SIFT) and speeded up robust features (SURF) with negligible performance degradation.

It is noted that all the aforementioned works are experiment based, such that the applicability of their results to other vision algorithms is unclear. The basic analysis of extracting gradient-based feature from raw Bayer images is introduced in [16]. In this paper, the impact of demosaicing on gradient-based feature extraction is studied. It is shown both theoretically and experimentally that the raw Bayer pattern images are applicable to the central difference gradient-based feature extraction algorithms with negligible performance degradation. Therefore, instead of demosaicing the Bayer pattern images before gradient computation, we propose to extract gradients directly from the Bayer pattern images by taking advantage of the color difference constancy assumption, which is widely used in demosaicing algorithms.

The reminder of the paper is organized as follows. Section II presents the background information, including the ISP pipeline and several gradient-based high-level vision features. Section III presents the derivation of the gradient-based feature extraction from the Bayer pattern images. Experimental results are presented in Section IV followed by the discussion in Section V and conclusions in Section VI.

II. Background

A. The Conventional ISP Pipeline

Shown in Fig. 1(b) is an ISP pipeline from Adobe DNG converter [17]. Although the specific algorithms and their orders may vary for different manufacturers, the basic steps in Fig. 1(b) are usually covered. The details of the functionality of each step is illustrated in Table I.

TABLE I
THE FUNCTIONALITIES OF THE STEPS IN ISP PIPELINE

ISP Steps	Functionality
Linerization	To transform the raw data into linear space.
Black Level & Flare Compensation	To compensate the noises contributed by black level current and flare.
Lens Correction	To compensate lens distortion and uneven light fall.
Demosaicing	To convert a single-channel Bayer pattern image a three-channel color image.
White Balance	To remove unrealistic color casts such that white objects are rendered white.
Color Space Transformation	To transform the camera color space to a standard color space.
Noise Reduction	To suppress noises introduced in preceding steps.
Sharpening	To enhance the edges for clarity improvement.
Color Manipulation	To generate different styles of photos.
Tone-mapping	To compress the dynamic range of images while preserving the visual effect.

B. Demosaicing

Demosaicing is a crucial step to convert a single-channel Bayer pattern image to a three-channel color image by interpolating the other two missing color components at each pixel. It has a decisive effect on the final image quality. In order to minimize the color artifacts, sophisticated demosaicing algorithms are always computation hungry.

The problem of demosaicing a Bayer pattern image has been intensively studied in the past decades and a lot of algorithms have been proposed [18]–[21]. All these algorithms can be grouped into two categories. The first category considers only the spatial correlation of the pixels and interpolates the missing color components separately using the same color channel. Although these single-channel interpolation algorithms may achieve fairly good results in the low frequency (smooth) regions, they always fail in the high-frequency regions, especially in the areas with rich texture information or along the edges [2].

To improve the demosaicing performance, the other category of algorithms takes the nature of the images' high spectral inter-channel correlation into account. Almost all these algorithms are based on either the color ratio constancy assumption [20] or the color difference constancy assumption [21]. According to the color image model in [20], which is a result of viewing Lambertian non-flat surface patches, the three color channels can be expressed as

$$I^k(x,y) = \rho_k(x,y)\left\langle \overrightarrow{N}(x,y), \overrightarrow{l} \right\rangle, \tag{1}$$

where ρ is the reflection coefficient, $\overrightarrow{N}(x,y)$ is the surface's normal vector at location (x,y), $\overrightarrow{l}$ is the incident light vector, $I(x,y)$ is the intensity at location (x,y) and $k \subseteq \{R,G,B\}$ indicates one of the three channels. Note that a Lambertian surface is equally bright from all viewing directions and does not absorb any incident light [22].

At a given pixel location, the ratio of any two color components, denoted by k and k', is given by

$$\frac{I^k(x,y)}{I^{k'}(x,y)} = \frac{\rho_k(x,y)\left\langle \overrightarrow{N}(x,y), \overrightarrow{l} \right\rangle}{\rho_{k'}(x,y)\left\langle \overrightarrow{N}(x,y), \overrightarrow{l} \right\rangle} = \frac{\rho_k(x,y)}{\rho_{k'}(x,y)}. \tag{2}$$

Suppose that objects are made up of one single material, i.e., the reflection coefficient ρ for each channel is a constant, the ratio of $\rho_k(x,y)/\rho_{k'}(x,y)$ reduces to a constant, such

that (2) can be simplified as

$$\frac{I^k(x,y)}{I^{k'}(x,y)} = constant. \tag{3}$$

Equation (3) is referred to as color ratio constancy. In the same manner, the color difference constancy assumption is given by

$$\begin{aligned} &I^k(x,y) - I^{k'}(x,y) \\ &= \rho_k(x,y)\left\langle \overrightarrow{N}(x,y), \overrightarrow{l} \right\rangle - \rho_{k'}(x,y)\left\langle \overrightarrow{N}(x,y), \overrightarrow{l} \right\rangle \\ &= [\rho_k(x,y) - \rho_{k'}(x,y)]\left\langle \overrightarrow{N}(x,y), \overrightarrow{l} \right\rangle \\ &= C(x,y). \end{aligned} \tag{4}$$

Note that the direction and amplitude of the incident light are assumed to be locally constant, such that the color component difference $C(x,y)$ is also a constant within a neighborhood of (x,y) [2].

The color ratio and difference constancy assumptions are widely used in various demosaicing algorithms [23]. In practical applications, the color difference constancy assumption always is preferred due to its superior peak signal to noise ratio (PSNR) performance. As will be shown, in this work, the color difference constancy can be utilized to directly extract the gradient information from the Bayer pattern images.

C. High-Level Features

In the past decades, many different feature descriptors such as Harr-like features [24], LBP [25], SIFT [26] and histograms of oriented gradients (HOG) [27] have been proposed for object detection. In this work, we mainly focus on the central difference gradient-based feature descriptors, study their applicability on Bayer pattern images and analyze the corresponding performances. Without loss of generality, HOG and SIFT are taken as examples in the analysis and experiments. The results can be extended to other descriptors, such as SURF [28], Color-SIFT [29], Affine-SIFT [30] and F-HOG [31], as long as the central difference is used for gradient computation.

SIFT is a local feature descriptor which detects key points in images. The computation of SIFT can be divided into five steps [32] as

1) Scale space construction. The scale space is approximated by the difference-of-Gaussian (DoG) pyramid, which is computed as

$$\begin{aligned} D(x,y,\sigma) &= \left(G\left(x,y,l^i\sigma\right) - G\left(x,y,l^{i-1}\sigma\right)\right) * I(x,y) \\ &= L\left(x,y,l^i\sigma\right) - L\left(x,y,l^{i-1}\sigma\right). \end{aligned} \tag{5}$$

Here, $G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$ is the Gaussian function, $l = 2^{\frac{1}{s}}$ is a constant multiplicative factor whose value is determined by the number of scales s, $*$ denotes the convolution operator, i indicates the i-th layer in DoG pyramid and $L\left(x,y,l^i\sigma\right)$ is the convolution of the original image with the Gaussion function at scale $l^i\sigma$.

2) Extremum detection. To detect the local maxima and minima by comparing each pixel with its neighbors in a 3×3 neighbourhood among the current scale, scale above and scale below.
3) Key point localization. To perform a refinement of key point candidates identified in the previous step. The unstable key points such as points with low contrast or poorly localized along an edge are rejected.
4) Orientation determination. To assign one or more orientations to each key point. A histogram is created for a region centered on the key point with radius of $3\sigma_0$, where σ_0 is 1.5 times that of the scale of the key point. The direction with the highest bar in the histogram is regarded as the dominant direction and directions with heights of larger than 80% of the highest bar is regarded as the auxiliary directions.
5) Key point description. To construct a descriptor vector for each key point. A gradient histogram with 8 bins is created for each 16×16 pixel region around the key point. The key point descriptor is constructed by concatenating the histograms of a set of 4×4 regions around the key point.

HOG is a feature descriptor initially proposed for pedestrian detection [27]. It counts the number of occurrences of gradient orientation in a detection window. The key steps of HOG feature generation are similar to steps 4 and 5 in the SIFT descriptor. The main difference is that orientation histograms in HOG are usually computed on an 8×8 cell and summarized as a global feature by a sliding window.

III. Gradient and Multiscale Models for Bayer Pattern Images

A. Gradient Extraction From Bayer Pattern Images

Image gradient measures the change of intensity in specific directions. Mathematically, for a two-dimensional function $f(x,y)$, the gradients can be computed by the derivatives with respect to x and y. For a digital image where x and y are discrete values, the derivatives can be approximated by finite differences.

There are different ways to define the difference of a digital image, as long as the following three conditions are satisfied: (i) zero in constant intensity area; (ii) non-zero along the ramps and (iii) nonzero at the onset of an intensity step or ramp [33]. One of the most commonly used image gradient computation is the central difference based approach as

$$G_x(x,y) = I(x+1,y) - I(x-1,y), \tag{6}$$

$$G_y(x,y) = I(x,y+1) - I(x,y-1). \tag{7}$$

Here $I(x,y)$ is the intensity at location (x,y), G_x and G_y represent the gradients in the horizontal and vertical directions, respectively. The computation of (6) and (7) can be implemented by the convolution of the templates in Fig. 2(a) with the images.

The fundamental idea of the proposed Bayer pattern image based gradient extraction is illustrated in Fig. 3(b). Instead of demosaicing the Bayer pattern images before difference

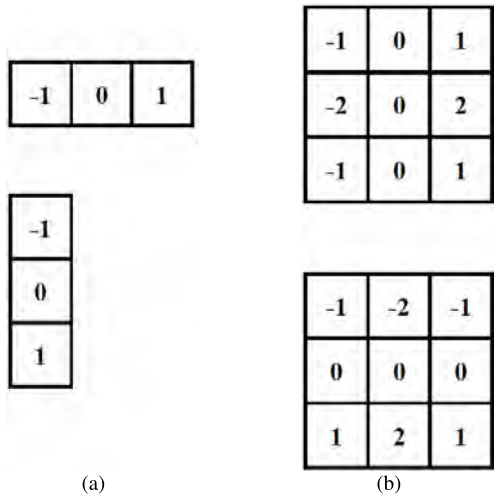


Fig. 2. Gradient operators. (a) The central difference operator and (b) the Sobel operator.

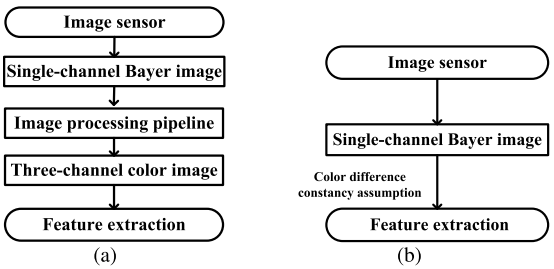


Fig. 3. Feature extraction pipelines. (a) The conventional pipeline. (b) The proposed pipeline.

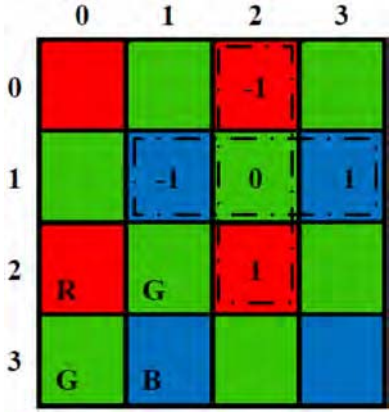


Fig. 4. Gradient computation based on Bayer pattern image.

computation as shown in Fig. 3(a), we propose to take advantage of the color difference constancy assumption directly for gradient extraction based on Bayer pattern images. Note that by convolving the filter templates in Fig. 2(a) directly with a Bayer pattern image, all the three conditions for a valid difference definition mentioned are satisfied. To illustrate this, let us consider the example in Fig. 4.

As we can see, the two input pixels for coefficients 1 and -1 in the convolution templates are from the same channel, i.e., differences are always computed on homogeneous pixels. As shown in Fig. 4, applying the convolution templates at locations $(1, 2)$ generates

$$G^B_{x(1,2)} = I^B_{(1,3)} - I^B_{(1,1)}, \tag{8}$$

$$G^R_{y(1,2)} = I^R_{(2,2)} - I^R_{(0,2)}, \tag{9}$$

where G^B and G^R are the gradients of the blue and red channels, respectively.

In the demosaicing tasks, it is a common practice to interpolate the G channel first followed by the R/B channels. This is because there are twice as many G channel pixels as R/B channel pixels in Bayer pattern images. The color difference constancy assumption in (4) can then be used to estimate the missing pixels of the R and B channels.

$$I^G(x, y) = I^k(x, y) + C^k(x, y). \tag{10}$$

Here, k represents either R or B channel, $C^k(x, y)$ is the difference between the R/B channel and the G channel at pixel location (x, y), which needs to be estimated in demosaicing tasks [34].

Consider two pixels within a small neighborhood at locations (x, y) and (x', y'), according to (10), we have

$$\begin{aligned} &I^G(x, y) - I^G(x', y') \\ &= I^k(x, y) - I^k(x', y') + C^k(x, y) - C^k(x', y') \\ &= I^k(x, y) - I^k(x', y') + \delta^k(x, y, x', y'). \end{aligned} \tag{11}$$

where $\delta^k(x, y, x', y') = C^k(x, y) - C^k(x', y')$. The value of $\delta^k(x, y, x', y')$ is crucial in our analysis and will be discussed in detail.

Generally, there are flat areas (e.g. background) and texture areas (e.g., corners and edges) in a natural image, these two situations will be discussed separately.

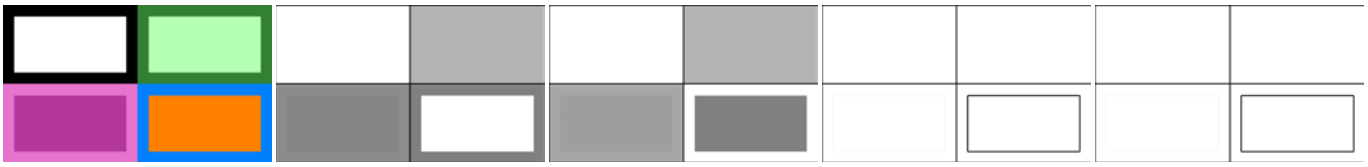
For the flat areas, the difference between two pixels is negligible such that

$$I^G(x, y) - I^G(x', y') = I^k(x, y) - I^k(x', y') \approx 0, \tag{12}$$

i.e., $\delta^k(x, y, x', y') \approx 0$. This means the intensity difference between channels is approximately constant across nearby pixel locations, i.e., the color changes are small in the neighborhood of flat areas.

Importantly, a constant C^k also means that some non-smooth color transitions (i.e., texture) are included as well. For example, for the two synthetic images in the top row of Fig. 5(a), suppose (x, y) is a point in the background and (x', y') is another point in the foreground. These two images correspond to the situation of $C^k(x, y) = C^k(x', y')$. Fig. 5(b)-5(c) are the difference images of $G-R$ and $G-B$, respectively, while Fig. 5(d)-5(e) are the corresponding gradient maps. Note that all the gradient maps and difference images are displayed as inverse images (1 − original gray value), where 1 means difference or gradient is zero and the the corresponding location is displayed as white (likewise for Fig. 6 and 12). Then,

1) Fig. 5(a) top-left: $I^R = I^G = I^B$ for both background and foreground. This results in $C^k(x, y) = 0$, which further leads to $\delta^k(x, y, x', y') = 0$, as shown in Fig. 5(b)-5(e), top-left images.
2) Fig. 5(a) top-right: $I^R = I^B$ for both background and foreground. This results in a constant $C^k(x, y)$ for



(a) (b) (c) (d) (e)

Fig. 5. Examples of color transitions that violate (the bottom-right image) and agree with (top and bottom-left images) the model assumption in (20). (a) Top-left: background with color [0.0, 0.0, 0.0] (black) and foreground color with [1.0, 1.0, 1.0] (white); Top-right: background with color [0.2, 0.5, 0.2] and foreground with color [0.7, 1.0, 0.7]; Bottom-left: background with color [0.9, 0.45, 0.8] and foreground with color [0.7, 0.22, 0.6]. Bottom-right: background with color [0.0, 0.5, 1.0] and foreground with color [1.0, 0.5, 0.0]. (b)-(c) G channel − R channel and G channel − B channel of (a). (d)-(e) The gradient map of (b) and (c).

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig. 6. Comparison of gradients extracted from color images and their Bayer version. (a) Image Kodim17 (top) and Image Kodim04 (bottom). (b)-(c): Gradient magnitude maps generated from (a) using the central difference operator and the Sobel operator in Fig. 2. (d) The resampled Bayer versions of Kodim17 and Kodim04 (displayed as a three-channel image). (e)-(f): Gradient magnitude maps generated from (d) using the central difference operator and the Sobel operator in Fig. 2. (g)-(h): The difference images generate from (a) by (G channel − R channel) and (G channel − B channel). (i)-(j): Gradient magnitude maps generated from (g) and (h) using operators in Fig. 2(a). (k) The gradient magnitude similarity (GMS) maps (29) between (b) and (e) with GMSD = 0.004 (top) and GMSD = 0.007 (bottom). (l) The GMS maps between (c) and (f) with GMSD = 0.011 (top) and GMSD = 0.022 (bottom).

both background and foreground, which further leads to $\delta^k(x, y, x', y') = 0$, as shown in Fig. 5(b)-5(e), top-right images.

For the above two cases, although there are obvious edges in the original images, we still have $\delta^k(x, y, x', y') = 0$.

For the more extreme texture areas, the analysis is more complex. To analyze the areas with complex textures, (11) can be further rewritten as

$$\begin{aligned}&\delta^k(x, y, x', y')\\&\quad = (I^G(x, y) - I^k(x, y)) - (I^G(x', y') - I^k(x', y')).\end{aligned} \tag{13}$$

Note that image's gradients are always computed among a small neighborhood. Considering the central difference-based horizontal gradient computation at pixel location (x, y), we have

$$\begin{aligned}&\delta^k(x+1, y, x-1, y)\\&= (I^G(x+1, y) - I^k(x+1, y)) - (I^G(x-1, y) - I^k(x-1, y))\\&= I^{G-k}(x+1, y) - I^{G-k}(x-1, y)\\&= G_x^{G-k}(x, y).\end{aligned} \tag{14}$$

Here, $G - k$ represents the difference image of the G channel and the R/B channel. It can be observed from (14) that $\delta^k(x+1, y, x-1, y)$ is exactly the gradient of the difference image at location (x, y). It has been shown in [35] that the difference images are slowly-varying over a spatial domain, meaning that the gradient $G_x^{G-k}(x, y)$ in (14) is negligible, i.e., $\delta^k(x+1, y, x-1, y)$ approximates to zero. This can be illustrated by the bottom-left image in Fig. 5(a). Suppose (x, y) is a point on the background border such that $(x+1, y)$ and $(x-1, y)$ are two points in the foreground and background, respectively. In this case, the following relationship holds (5(a), bottom-left).

$$\begin{cases} C^k(x+1, y) \neq C^k(x-1, y), \\ Sgn(C^k(x+1, y)) = Sgn(C^k(x-1, y)). \end{cases} \tag{15}$$

Here, $Sgn(\cdot)$ is the signum function. Let $k = B$, the results of (14) is

$$\begin{aligned}&\delta^B(x+1, y, x-1, y)\\&= (I^G(x+1, y) - I^B(x+1, y)) - (I^G(x-1, y) - I^B(x-1, y))\\&= (0.45 - 0.8) - (0.22 - 0.6)\\&= 0.03\end{aligned} \tag{16}$$

The corresponding result is illustrated in Fig. 5(e) bottom-left, where the edge is negligible.

Note that there are also failure cases. For example, the bottom-right image in Fig. 5(a) satisfies

$$\begin{cases} C^k(x+1, y) \neq C^k(x-1, y), \\ Sgn(C^k(x+1, y)) \neq Sgn(C^k(x-1, y)). \end{cases} \tag{17}$$

In this case, the result of (14) will be $\delta^k(x+1, y, x-1, y) = 1$, which is illustrated in Fig. 5(d) and 5(e), bottom-right. Details of failure cases will be discussion in Section IV-C1.

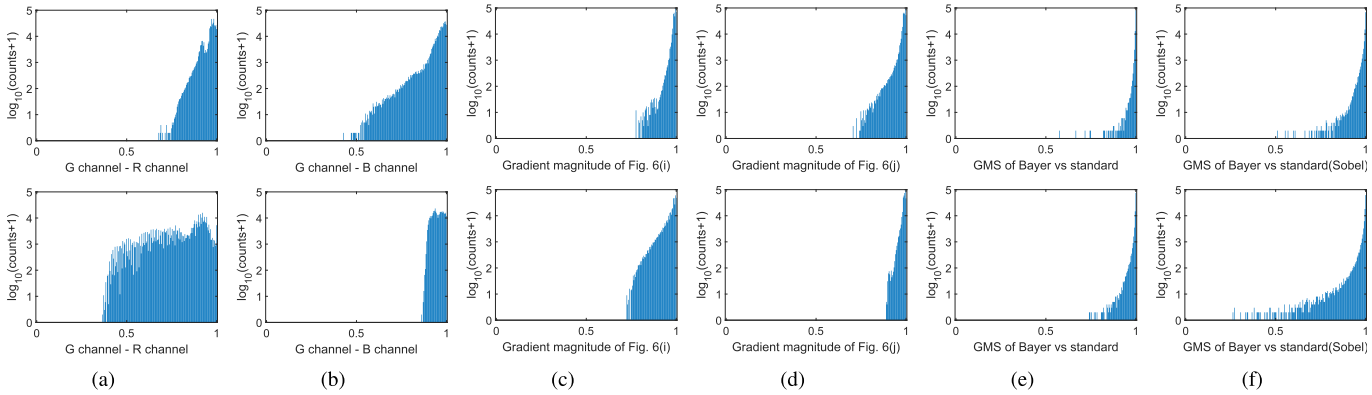


Fig. 7. The gray-level histograms of (g) - (l) in Fig. 6.

Fig. 6 illustrates two situations: image with dull colors (top) and image with high saturation colors (bottom). Fig. 6(g) and Fig. 6(h) are the difference images of G channel − R channel and G channel − B channel ($C^k(x, y)$ in Eq. 2), respectively, while Fig. 6(i) and Fig. 6(j) are the corresponding gradient magnitude maps of the difference images ($\delta^k(x+1, y, x-1, y)$ in (14)) computed using the central difference operator. The corresponding gray-level histograms of these images are shown in Fig. 7(a)-7(d). It can be found that for images with dull colors, the differences between G and R channels are distributed in a much smaller range than that of high saturation color images, while the differences between G and B channels are distributed in a larger range. For all these images, most of $\delta^k(x+1, y, x-1, y)$ are distributed in a small range, as shown in Fig. 7(c) and 7(d). The distribution in Fig. 7(c) bottom is wider than the other three plots, which are caused by texture areas such as hat and hair edges in Fig. 6(i). As we can see, apart from these small exceptions, Fig. 6(i) and Fig. 6(j) are almost all white. Therefore, for most cases, $\delta^k(x, y, x', y')$ is small and can be ignored if pixel locations (x, y) and (x', y') are within a small neighborhood.

As a result of the above discussion, (11) can be rewritten as

$$I^G(x, y) - I^G(x', y') \approx I^R(x, y) - I^R(x', y'), \quad (18)$$

$$I^G(x, y) - I^G(x', y') \approx I^B(x, y) - I^B(x', y'). \quad (19)$$

Combining the gradient definition of (6) and (7) with (18) and (19), we have

$$G \approx G^G \approx G^R \approx G^B, \quad (20)$$

meaning that the gradients of natural images can be computed using any one of the three channels as long as the color difference constancy holds. Combining (20) with (8) and (9), we have

$$G_{x(1,2)} = G^B_{x(1,2)} = I^B_{(1,3)} - I^B_{(1,1)}, \quad (21)$$

$$G_{y(1,2)} = G^R_{x(1,2)} = I^R_{(2,2)} - I^R_{(0,2)}. \quad (22)$$

Therefore, even though two color components are missing at each pixel, the gradients of location $(1, 2)$ can be computed directly from the Bayer pattern image using the blue and red channel. The gradients of any other pixel locations can be computed in the same manner.

Generally, the above conclusion can be extended to other symmetrical first-order differential operators (with alternating zero and nonzero coefficients) on any kind of Bayer pattern. Let us take the Sobel operators in Fig. 2(b) as an example. Applying the Sobel operators in Fig. 2(b) to the pixel location $(1, 2)$ of the Bayer pattern image results

$$\begin{aligned} G'_{x(1,2)} &= I^G_{(0,3)} + 2\times I^B_{(1,3)} + I^G_{(2,3)} - I^G_{(0,1)} - 2\times I^B_{(1,1)} - I^B_{(2,1)} \\ &= (I^G_{(0,3)} - I^G_{(0,1)}) + 2\times (I^B_{(1,3)} - I^B_{(1,1)}) + (I^G_{(2,3)} - I^G_{(2,1)}), \end{aligned} \quad (23)$$

$$\begin{aligned} G'_{y(1,2)} &= I^G_{(2,1)} + 2\times I^R_{(2,2)} + I^G_{(2,3)} - I^G_{(0,1)} - 2\times I^R_{(0,2)} - I^G_{(0,3)} \\ &= (I^G_{(2,1)} - I^G_{(0,1)}) + 2\times (I^R_{(2,2)} - I^R_{(0,2)}) + (I^G_{(2,3)} - I^G_{(0,3)}). \end{aligned} \quad (24)$$

As for gradient computation using (6) and (7), differences are always computed on homogeneous pixels for Sobel-based differential operations in (23) and (24), i.e., pixel values are always subtracted from pixel values of the same channel. Moreover, according to the color difference constancy assumption, (23) and (24) can be rewritten as

$$\begin{aligned} G'_{x(1,2)} &\approx (I^{\widehat{R}}_{(0,3)} - I^{\widehat{R}}_{(0,1)}) + 2\times (I^{\widehat{R}}_{(1,3)} - I^{\widehat{R}}_{(1,1)}) + (I^{\widehat{R}}_{(2,3)} - I^{\widehat{R}}_{(2,1)}) \\ &\approx (I^G_{(0,3)} - I^G_{(0,1)}) + 2\times (I^{\widehat{G}}_{(1,3)} - I^{\widehat{G}}_{(1,1)}) + (I^G_{(2,3)} - I^G_{(2,1)}) \\ &\approx (I^{\widehat{B}}_{(0,3)} - I^{\widehat{B}}_{(0,1)}) + 2\times (I^B_{(1,3)} - I^B_{(1,1)}) + (I^{\widehat{B}}_{(2,3)} - I^{\widehat{B}}_{(2,1)}), \end{aligned} \quad (25)$$

$$\begin{aligned} G'_{y(1,2)} &\approx (I^{\widehat{R}}_{(2,1)} - I^{\widehat{R}}_{(0,1)}) + 2\times (I^R_{(2,2)} - I^R_{(0,2)}) + (I^{\widehat{R}}_{(2,3)} - I^{\widehat{R}}_{(0,3)}) \\ &\approx (I^G_{(2,1)} - I^G_{(0,1)}) + 2\times (I^{\widehat{G}}_{(2,2)} - I^{\widehat{G}}_{(0,2)}) + (I^G_{(2,3)} - I^G_{(0,3)}) \\ &\approx (I^{\widehat{B}}_{(2,1)} - I^{\widehat{B}}_{(0,1)}) + 2\times (I^{\widehat{B}}_{(2,2)} - I^{\widehat{B}}_{(0,2)}) + (I^{\widehat{B}}_{(2,3)} - I^{\widehat{B}}_{(0,3)}), \end{aligned} \quad (26)$$

where $\widehat{R}$, $\widehat{G}$ and $\widehat{B}$ represent the missing color components at the corresponding locations. Therefore, the Sobel-based gradients can also be extracted directly from the Bayer pattern images as long as the color difference constancy holds.

In terms of different Bayer patterns, they are merely different arrangements of the RGB pixels, while the alternating pattern of R, G and B at each row and column are preserved. For example, discarding the first column of the Bayer pattern in Fig. 4 generates the GRBG Bayer pattern. Therefore, different Bayer patterns do not have any impact on the applicability

of the discussed differential operators to Bayer pattern images. Moreover, the discussed gradient extraction method can be directly extended to other special CFA patterns with alternating color filter arrangements, e.g., RYYB, RGB-IR, as long as the arrangement of CFA patterns matches the gradient operators such that gradient operations are performed on the same color channel, i.e., subtract or add operations are performed on the same color channel, and the coefficients of the subtract or add terms in the gradient operator are equal such that the gradients compute from R/B channel can be approximated to G channel.

To validate the proposed Bayer pattern image-based gradient extraction, the differential operators in Fig. 2 are applied to true color images Kodim17 and Kodim04 from the Kodak image dataset [36] and the corresponding resampled Bayer version. The generated gradient maps are shown in Fig. 6. For display purpose, images in Fig. 6(a) are shown as color images while the gradient magnitude maps in Fig. 6(b) and 6(c) are computed from the corresponding gray-scale images generated using Fig. 6(a). The Bayer pattern images in Fig. 6(d) are presented as three-channel images to illustrate its Bayer "mosaic" structure. For the clearness of presentation, all the gradient maps and difference images in Fig. 6 are displayed as inverse images. As illustrated in Fig. 6, the gradient maps generated from the Bayer pattern images look almost the same as that generated from the true color version. To compare these gradient maps, two GMS maps are presented in Fig. 6(k) and 6(l), and the corresponding distributions of GMS values are presented in Fig.7(e) and 7(f). As can be seen, the GMS maps are almost pure white, and the histograms are distributed in a small range, meaning that the compared gradient maps are very close to each other. Overall, gradients of Bayer images yield a good approximation of the image gradients, except for a few pixels around certain color edges.

B. The Multiscale Model for Bayer Pattern Images

In SIFT, the scale-space is approximated by a DoG pyramid. The construction of the DoG pyramid can be divided into two parts: Gaussian blurring at different scales and resizing of the blurred images. Due to the special alternating pixel arrange of Bayer pattern images, directly Gaussian blur the images will destroy the "mosaic structure". This phenomenon is illustrated in Fig. 8. If the Bayer pattern image is directly Gaussian filtered, the resulting image (after demosaicing) looks like a "three-channel grayscale image" as shown in Fig. 8(d), meaning that the Bayer image is treated as a single channel image, ignoring the "mosaic structure" from the beginning and, effectively, loosing/destroying the color information. Thus, smoothing on Bayer pattern image directly will blend the channels, which is akin to a RGB-to-gray conversion. Moreover, loss of the color information makes some of the algorithms in the SIFT family such as "C-SIFT" and "RGB-SIFT" no longer applicable.

To address the above mentioned problem, the super-pixel approach as illustrated in Fig. 8(e) is used in this work. A super-pixel is a compound pixel consisting of a complete Bayer pattern. The Bayer pattern image can therefore be regarded as a "continuous" image filling with super-pixels. Operating on the super-pixel structure preserves the Bayer pattern of the original images. Fig. 8(f) shows the Gaussian blurred image (after demosaicing) based on the super-pixel structure. As can be seen, it is close to that generated by the full color approach. Moreover, the super-pixel structure can also be used for resizing when constructing the scale space. The detailed comparison results will be presented in Section IV-C.

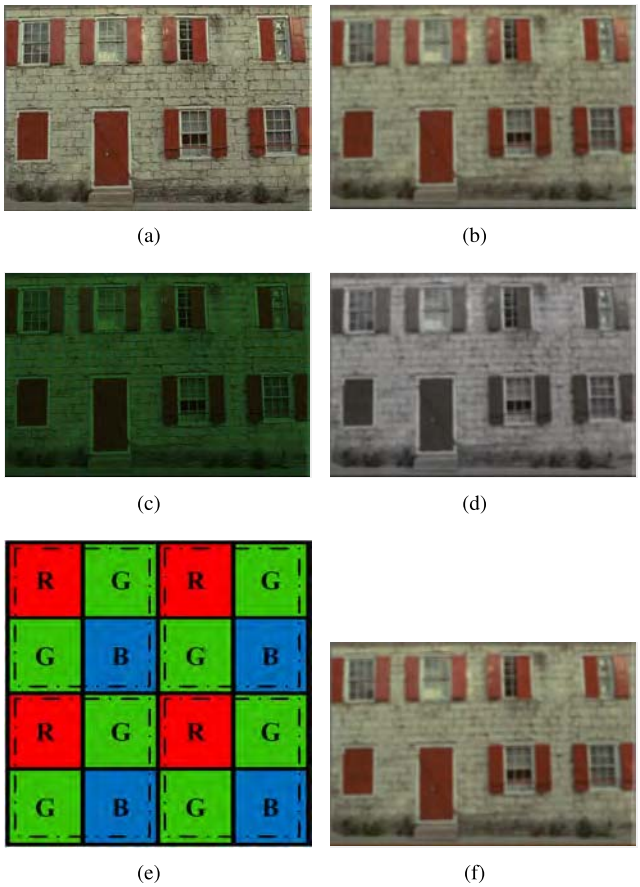


Fig. 8. (a) The true color image Image Kodim01. (b) Gaussian blurred version of (a). (c) Bayer version of (a). (d) Direct Gaussian blurred version of (c) then demosaicing. (e) The 2×2 super-pixel structure. (f) Super-pixel structure-based Gaussian blurred version of (c) then demosaicing.

IV. Experiments

In this section, experimental results are presented to demonstrate the effectiveness of the proposed Bayer pattern image-based gradient extraction. The datasets used in the experiments are introduced first, followed by the details of the experiments setup and evaluation results.

A. Datasets

There are five datasets used for differents experiments in this work. Among these five datasets, four are commonly used benchmarks in different image processing and computer vision tasks such as demosaicing, pedestrian detection. A brief description of these datasets is presented in Table II.

TABLE II
NOTATION OF DIFFERENT DATASETS USED IN EXPERIMENTS

Datasets	Brief Introduction	Generation of the Corresponding Color/Bayer Versions	
		Color	Bayer
The Kodak lossless true color image suite [36]	A popular standard test suite for demosaicing algorithms.	-	Resampling according to the corresponding Bayer pattern.
The SHTech pedestrian dataset	Our own pedestrian dataset shoot by a Huawei Honor 8 mobile phone with the FreeDcam APP [37] to by pass the entire ISP.	ISP pipeline in [17].	-
The PASCALRAW dataset [38]	A recently published raw image dataset for object detection.	ISP pipeline in [17].	-
The INRIA pedestrian dataset [39]	A popular dataset for pedestrian detection algorithms.	-	Reverse ISP pipeline introduced in [8].
The See-in-the-Dark (SID) dataset [40]	A recently published raw image dataset shoot under low light conditions.	-	-

B. Experiments Setup and Evaluation Criteria

1) Gradient Map and Multiscale Model: In our experiment, the operators in Fig. 2(a) are used to extract the gradients from color images and their corresponding Bayer versions. For color images, gray scale images are generated for gradient extraction. To blur and resize the Bayer pattern images, the super-pixel structure discussed in Section III-B is utilized.

To estimate the differences among gradient maps, blurred images and resized images, some image quality assessment methods are used in these experiments.

The gradient magnitude similarity deviation (GMSD) is proposed in [41] to evaluate the similarity of gradient magnitudes. Given two gradient maps, the GMSD is defined by

$$GMSD = \sqrt{\frac{1}{H \times W}\sum_{x=1}^{H}\sum_{y=1}^{W}(GMS(x,y) - GMSM)^2}, \tag{27}$$

where,

$$GMSM = \frac{1}{H \times W}\sum_{x=1}^{H}\sum_{y=1}^{W} GMS(x,y), \tag{28}$$

$$GMS(x,y) = \frac{2m_1(x,y)m_2(x,y) + c}{m_1^2(x,y) + m_2^2(x,y) + c}. \tag{29}$$

Here, W and H are the width and height of the images, $m_j(x,y)$ is the gradient magnitude of the j-th image at pixel location (x,y), defined by $m(x,y) = \sqrt{G_x(x,y) + G_y(x,y)}$, and c is a small value set to 0.0026 to avoid divisions by 0. According to [41], the smaller the GMSD is, the closer the gradient maps are.

Mean squared error (MSE) is the simplest and most commonly used full-reference quality metric. It is an evaluation that is computed by averaging the squared intensity differences of distorted and reference image pixels. For two given images, the MSE is given by

$$MSE = \frac{1}{H \times W}\sum_{x=1}^{H}\sum_{y=1}^{W}\left(I_1(x,y) - I_2(x,y)\right), \tag{30}$$

where W and H is the width and height of the image. The MSE can be converted to PSNR by

$$PSNR = 10\log_{10}\left(\frac{(2^n-1)^2}{MSE}\right), \tag{31}$$

where n represent the pixel bit depth of images. For images with 8-bit pixel depth, the typical values of PSNR for lossy images are between 30 and 50 dB [42].

Structural similarity (SSIM) is also a full-reference quality metric which compares luminance, contrast, structure among two images [43]. The SSIM ranges from 0 to 1, where 1 means that the two compared images are identical. Due to the fact that SSIM is a metric for local region comparison, the mean SSIM (MSSIM) is usually used in practice.

2) Influence of Noise: Noise reduction, which has a deterministic impact on the quality of imaging, is a critical step in image processing pipelines. Basically, there are two kinds of noise in an image, i.e., signal-independent noise (e.g., bad-pixels, dark currents) and signal-dependent noise (e.g., photon noise). For modern cameras, the signal-dependent noise, which is affected by lighting conditions and exposure time [44], [45], is the dominant noise source. In [44], image noise is modeled as additive noise, which is a mixture of Gaussian and Poissonian process that obeys the distribution of

$$\eta_h \sim N(0, ay(x) + b). \tag{32}$$

Here, η_h is the signal noise, $y(x)$ is the noise-free signal and a, b are two parameters. Note that the dataset used in pedestrian detection experiments is all shoot under sufficient illumination and proper exposure. To study the influence of noise on the proposed Bayer pattern image-based gradient feature extraction pipeline, we use the See-in-the-Dark (SID) dataset introduced in [40] and the model in (32) to obtain a set of different noise parameters under low light conditions (2650 parameter pairs in total) and randomly choose parameter pairs for each image in pedestrian detection datasets to generate the corresponding noisy images.

3) HOG Descriptor: To compare the performance of HOG descriptors extracted from color images and Bayer pattern images, the traditional HOG + support vector machine (SVM) framework proposed in [27] is used to detect pedestrians from color images and their Bayer versions. The INRIA, SHTech and PASCALRAW dataset are used in the pedestrian detection

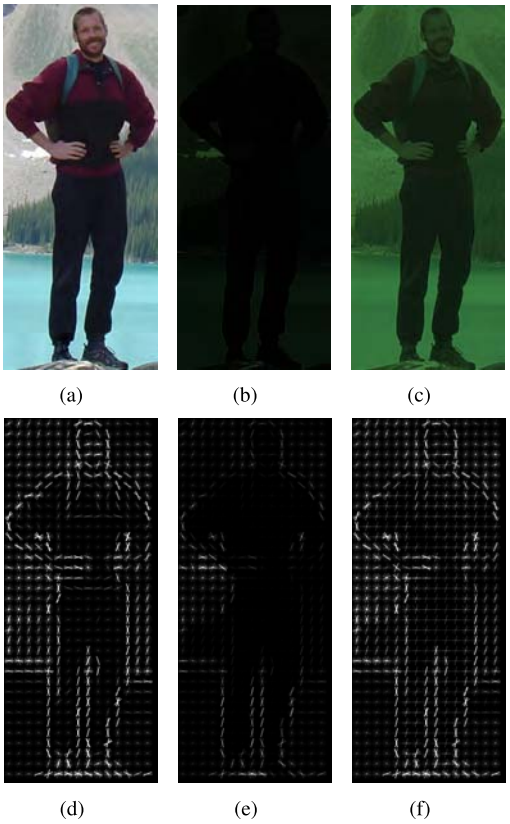


Fig. 9. Comparison of HOG features. (a) An image from the INRIA pedestrian dataset. (b) The converted Bayer version of (a) using the reverse pipeline in [8]. (c) The Bayer version image after gamma compression. (d)-(f): Visualization of the generated HOG descriptors.

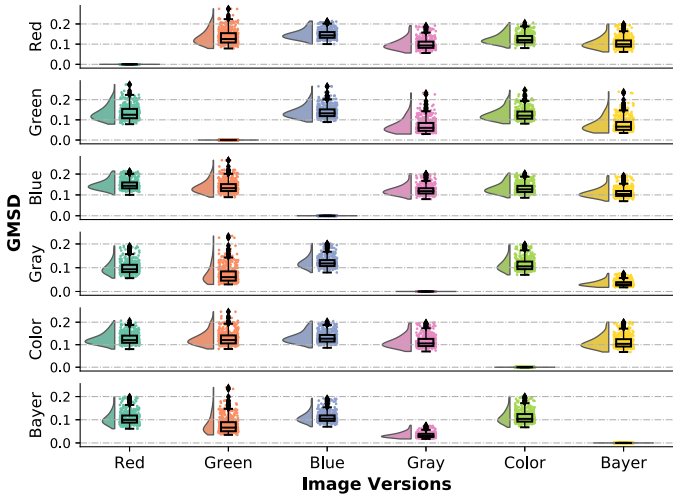


Fig. 10. GMSD (between different channels and versions of images) distribution of the SHtech Dataset.

task, where models are trained and tested on each dataset separately. Precision-recall curve along with average precision are used to present the detection results [46].

4) SIFT Descriptor: For the proposed Bayer pattern image-based SIFT feature extraction, extremums are searched among a 5×5 neighborhood instead of 3×3. To validate the scale and rotation invariant property of the generated SIFT features, key points are detected from the transformed images, i.e., the resized, rotated and blurred images. These key points

TABLE III

COMPARISON RESULTS OF BAYER IMAGE BASED AND COLOR IMAGE BASED GRADIENTS

Datasets	Average		
	MSSIM	PSNR	GMSD
Kodak	0.975	38.276	0.069
SHTech	0.850	34.683	0.119
PASCALRAW	0.9367	37.36	0.127
INRIA	0.817	30.288	0.148

For the INRIA dataset, gamma compression with scale factor of 2 and exponent of 0.5 is used.

are matched with the ones detected from the untransformed images. The repeatability criteria introduced in [47] are used to evaluate the performance of SIFT descriptors in finding matching points. Given a pair of images, repeatability is defined by

$$P = \frac{M}{min\,(n_1, n_2)}, \tag{33}$$

where n_1 and n_2 are the number of descriptors detected on the images, T is the transform between the original image I and its transformed version I_{tran} [48], M is the number of correct matches. Pixel coordinates (x_1, y_1) and $T^{-1}\{(x_2, y_2)\}$ is considered matched within a t-neighborhood if

$$d((x_1, y_1), T^{-1}\{(x_2, y_2)\}) < t. \tag{34}$$

Here, $d(\cdot)$ is the Euclidean distance between (x_1, y_1) and $T^{-1}\{(x_2, y_2)\}$. For a given pixel coordinates (x_1, y_1), the θ-rotated pixel coordinate (x_2, y_2) can be computed as

$$(x_2, y_2, 1) = (x_1, y_1, 1)H \tag{35}$$

Here,

$$H = \begin{pmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is the homography matrix, corresponding to transform T in (34). Moreover, for two $N \times 3$ matrices A and B, which consist of $N(N \geqslant 3)$ pairs of matched points $(x_1^j, y_1^j, 1)$ and $(x_2^j, y_2^j, 1)$, $j = 1, 2, 3 \ldots N$, the homography matrix $\widehat{H}$ can be estimated as

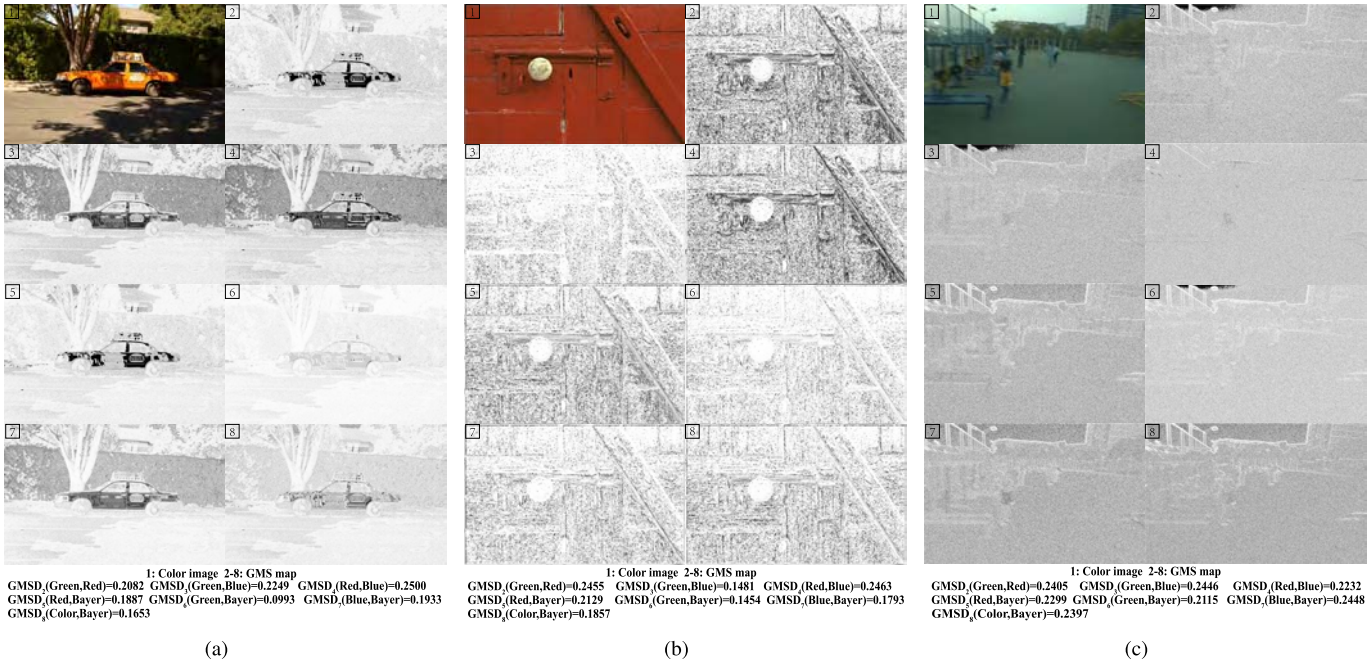
$$\widehat{H} = A^{\dagger}B, \tag{36}$$

where $A^{\dagger}$ is the pseudo-inverse of A.

C. Experimental Results

1) Comparison of Gradient Maps: In this experiment, the gradient maps generated from the original color images and the corresponding Bayer versions are compared. Note that for the INRIA dataset, gamma compression is applied to the converted Bayer pattern images to adjust the contrast, while this is not needed for the other two datasets.

The comparison results of different versions of gradient maps are presented in Table III. Generally speaking, all the three evaluation criteria reveal similar trends that different versions of gradient maps are close to each other. As shown in Table III, for the Kodak dataset, the gradients generated



(a) (b) (c)

Fig. 11. Three situations which cause large gradient difference for different channels. (a) The light source shines directly on a smooth surface. (b) Irregular texture and (c) heavy noise.

TABLE IV
COMPARISON RESULTS OF THE TRUE COLOR IMAGES AND IMAGES GENERATED USING DIFFERENT DEMOSAICING ALGORITHMS

Methods	Average		
	MSSIM	PSNR	GMSD
Nearest Neighbor	0.8865	25.744	0.082
Linear Interpolation	0.945	29.255	0.089
Cubic Interpolation	0.952	29.354	0.084
Adaptive Color Plane Interpolation	0.976	34.452	0.070
Hybrid Interpolation	0.990	39.010	0.065

from color as well as Bayer pattern images are almost identical, while for the other two datasets, the similarities are slightly lower. This is because for the Kodak dataset, both the color and Bayer pattern images can be regarded as "true" (a true color image dataset with Bayer version generated by resampling), while for the SHTech dataset and the PASCALRAW dataset, images are interpolated from Bayer pattern images using demosaicing algorithm. It is well known that extra errors will be introduced no matter how sophisticated the demosaicing algorithms is. This can be observed from the comparison results of the true color images and images generated using different demosaicing algorithms shown in Table IV.

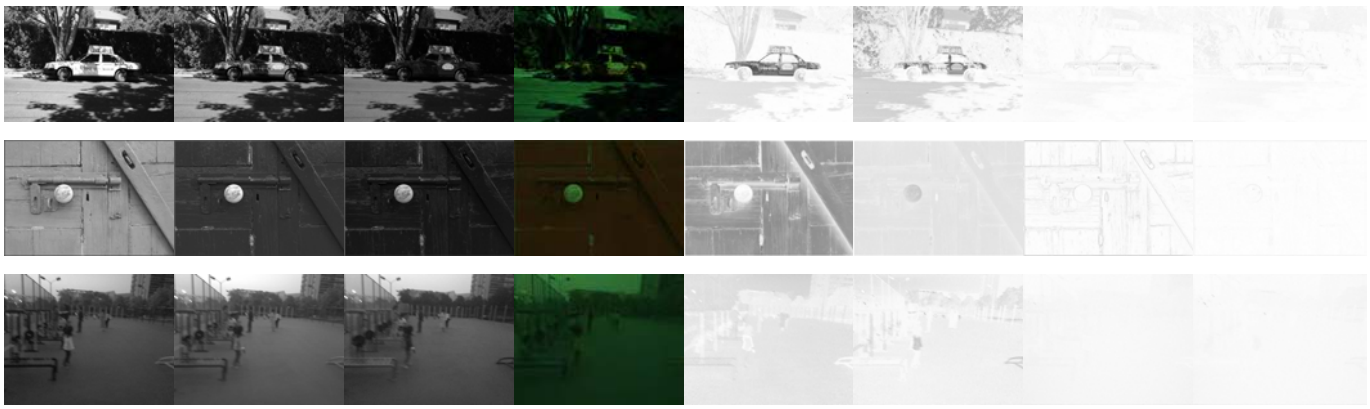
Moreover, for the INRIA dataset, both the color and Bayer pattern images are "estimated" since the color version is interpolated and the Bayer version is reversely converted from the color version. Errors are injected in both forward and reverse ISP pipeline. Therefore, for the evaluation of the proposed Bayer pattern image-based gradient extraction pipeline, the Kodak dataset is more reliable than the other two.

This can be illustrated using Fig. 9, where three versions of HOG descriptors, i.e., HOG from the original image, HOG from the Bayer pattern image without gamma compression and HOG from the Bayer pattern image with gamma compression, are presented. Note that as we mentioned, for the reversed INRIA dataset, proper gamma compression is necessary because the reversed images are at a low bit width, which is a side effect of forward + reverse ISP for Bayer image generation. As shown in Fig. 9(f), the descriptors cannot find enough features in low contrast Bayer pattern image without gamma compression (in Fig. 9(b)). But after adjusting the contrast by gamma compression, the HOG feature extracted from Fig. 9(c) becomes more stable, and close to the one extracted from the original color image in Fig. 9(d).

Fig. 10 presents the distribution of per-channel GMSD comparison results of the SHTech dataset, where color means the maximum gradient among the three channels is selected when computing the gradient maps. It can be found that the distributions of GMSD between any pairing of the three channels are close. The gradient maps computed from gray images and Bayer images are closer to that computed from the green channel. This is because the green channel contributes a larger proportion in both gray images (60%) and Bayer images (50%).

By analyzing the outliers in Fig. 10, it is found that there are three situations that lead to notable gradient difference, which may harm the gradients generated from Bayer images. These situations are illustrated in Fig. 11. Note that GMS is designed to range from 0 to 1, where 1 means no error. Thus, the brighter in GMS map, the higher the similarity.

The first situation is when the light source shines directly on a smooth surface (e.g. smooth wall, metal, etc.), especially for bright colored smooth objects. This situation violates the assumptions of Lambertian non-flat surface patches model because the reflection, in this case, is closer to specular reflection than diffuse reflection. A flat surface cannot be treat



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 12. The Corresponding (a) R channel, (b) G channel, (c) B channel, (d) Bayer pattern images of Fig. 11. (e) G channel − R channel. (f) G channel − B channel. (g) Gradient map of (e). (h) gradient map of (f).

as a Lambertian surface, because the brightness of an object is different when seen from different view point. Highlight areas caused by specular reflection make the illuminance no longer slow-varying. Fig. 11(a) illustrates this phenomenon. When sunlight hits the car directly, the GMS map shows a big difference among the bodywork (the dark areas in GMS map), leading to a big $C^k(x, y)$ in the car body but a small $C^k(x, y)$ in the background. This is illustrated in Fig. 12(e) and Fig. 12(f) (top). These areas result in edges as shown in Fig.12(g) and Fig.12(h) (top), corresponding to the non-zero $\delta^k(x+1, y, x-1, y)$ term in (14). The second situation is when there are irregular textures as shown in Fig. 11(b). In this situation, the $\delta^k(x+1, y, x-1, y)$ term in (14) can no longer be ignored, as shown in Fig. 12(g) and Fig. 12(h) (middle). However, these kind of violations appear mostly inside objects such that the influence on the edges is relatively small. For example, the edge of the door handle. The last situation is when there exists heavy noise as shown in Fig. 11(c). This situation is usually caused by low light condition and motion blur. It can be found from the GMS map that in this case, the gradient difference is evenly distributed throughout the image.

It can be found from the examples in Fig. 11 that the GMSD values of Bayer pattern images with other images, especially with green channel images, are smaller than other combinations. The failures that caused by a bright spot or saturated colors may not occur between all channels because they may lead to a small $\delta^R(x+1, y, x-1, y)$ but a big $\delta^B(x+1, y, x-1, y)$ (Fig. 12(g) and 12(h) top) or vice versa (Fig. 12(g) and 12(h) middle).

2) Blur and Resize: The purpose of this experiment is to show that multiscale model construction (mainly resize and scale operation) can also be performed on Bayer images by super-pixel based resize and scale operations. The operations can either be performed in RGB domain (three-channel) or Bayer domain (single-channel). Since demosaicing affects performance (Table IV), we performed these comparisons in Bayer domain. The resize operation here refers to the change of width and height of a digital image into a specified size, e.g, scale = 0.5 means to reduce the height and width of a image to half. For Bayer pattern images, blur and resize are directly applied on the super-pixel structure, while for color images, the original images are blurred and resized followed by the generation of Bayer pattern images through resampling, i.e., a blur + resize + resampling pipeline is used to generate Bayer pattern images from color images. The Kodak dataset is used in this experiment.

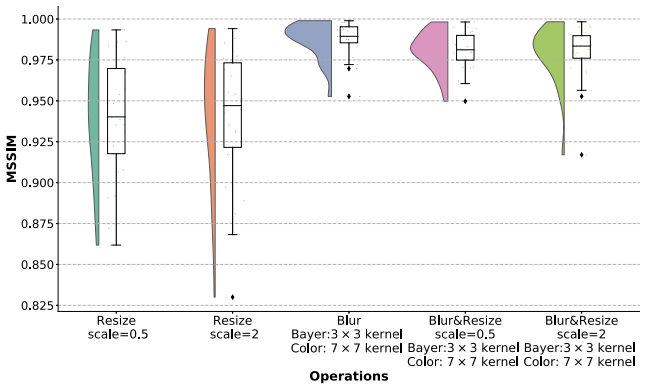
TABLE V
COMPARISON RESULTS OF BAYER IMAGE BASED AND COLOR IMAGE BASED BLUR AND RESIZE

Operation	Parameter		Average		
	Bayer	Color	MSSIM	MSE	PSNR
Gaussian blur	3×3 kernel	3×3 kernel	0.952	46.010	32.334
		5×5 kernel	0.979	18.040	36.293
		7×7 kernel	0.988	9.807	38.962
		9×9 kernel	0.985	11.762	38.094
Resize	Scale=0.5		0.938	70.453	30.232
	Scale=2		0.912	93.584	30.031
Blur & Resize	3x3 kernel, scale=0.5	7x7 kernel, scale=0.5	0.977	21.444	35.499
	3x3 kernel, scale=2	7x7 kernel, scale=2	0.976	15.667	36.691

Presented in Table V are the comparison results of blur and resize. For a certain $a \times a$ kernel, σ in Eq. (5) can be determined by the specific application or using the following equation [49]

$$\sigma = 0.3 \times ((a-1) \times 0.5 - 1) + 0.8. \tag{37}$$

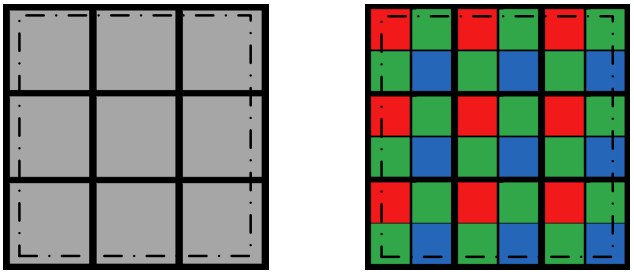
According to the experiment, blur and resize on Bayer pattern images using super-pixel structure generates similar results with that on color images. It can be observed from Table V that the 7×7 kernel for color images approach to the 3×3 kernel for super-pixel Bayer pattern images. This is because a super-pixel is a collection of pixels in a Bayer pattern which may expand the smooth area. As illustrated in Fig. 14, a 3×3 kernel on super-pixel Bayer pattern images covers a 6×6 pixel location in the original Bayer pattern image. Therefore, we expand the kernel used in gray images accordingly. Since the length of kernels needs to be odd, we have tried different kernel sizes in our experiments and



(a)

(b)

Fig. 13. (a) MSSIM distribution of the corresponding operations in Table V and (b) original images and SSIM maps after scaling operation. The top-left in (b) is an image with rich details and the bottom-left one is an image with less textures. The SSIM maps are generated from the Bayer image scaled by super-pixel structure directly and resampled (as Bayer) scaled color image. The blur parameters are 3×3 kernel and $\sigma = 0.8$ for Bayer images and 7×7 kernel and $\sigma = 1.4$ for color images.



(a) (b)

Fig. 14. The coverage of a 3×3 kernel on a (a) gray image and (b) the corresponding super-pixel Bayer pattern image.

presented the results in Table V. According to the results in Table V, the Bayer pattern image-based blur and resize generates similar results to the color image-based operations. Fig. 13(a) presents the MSSIM distribution of the operations in Table V. It can be found that the resize operation makes the distribution more dispersed, while performing blur (low-pass filtering) before resize may alleviate the quality loss caused by the scaling operation. Outliers often appear in images with rich details, e.g., the top left image in Fig. 13(b). The bottom-left image gives an example with less texture. As the SSIM maps show, images with rich details have larger difference among edges and performing blur before resize can improve it.

Moreover, to evaluate the memory access and computation time of the proposed method, the following two different pipeline configurations are compared.

TABLE VI
EVALUATION RESULTS OF PIPELINE 1 AND PIPELINE 2

		Original		Normalized	
		Time(ms)	Memory(kb)	Time	Mermory
Pipeline 1		0.87	1148936	1	1
Pipeline2	Nearest Neighbor	0.91	1698144	1.05	1.48
	Linear Interpolation	89.45	1673512	102.88	1.46
	Cubic Interpolation	95.62	2212948	109.98	1.93
	Adaptive Color Plane Interpolation	65.33	2699868	75.14	2.35
	Hybrid Interpolation	57.66	3079720	66.32	2.68

- Pipeline 1. Starting from Bayer images, perform blur + resize using the super-pixel method without demosaicing, then compute gradient magnitude images.
- Pipeline 2. Starting from Bayer images, demosaic it to color image, then perform blur + resize to each channel, generate gray images and compute gradient magnitude image.

The comparison results of pipeline 1 and 2 are presented in Table VI. Five different demosaicing algorithms are used in the evaluation of pipeline 2. All these pipelines are profiled using MATLAB R2020a, on a Windows 10 PC with i7-7700 CPU and 16G memory. This experiment is performed on resampled Kodak dataset. It can be found from Table IV and Table VI that complex interpolation algorithms lead high image quality, but also increase time and memory consumption. By skipping the complex operations, both time and memory can be saved.

3) Key Points Matching: Fig. 15 illustrates the key point matching performance based on SIFT feature using the original color version of Kodim09 image and its resampled Bayer version. The arrows in Fig. 15(a)-15(d) illustrate the scale and orientation of 20 SIFT descriptors, the cyan lines in Fig 15(e)-15(h) indicate the matched point pairs. As we can see, matched points can be identified in both color and Bayer pattern image pairs, meaning that the SIFT features extracted from the Bayer pattern images are robust regardless of the rotate operation. Note that the SIFT descriptors extracted from Bayer pattern image look different from that extracted from gray image. This is because we generate DoG pyramids of Bayer image based on super-pixel structure and extrema are searched among a 5×5 neighborhood instead of 3×3. It is the difference in DoG pyramids and search area that mainly lead to different descriptors between gray images and Bayer images.

To evaluate the scale and rotation invariance of the SIFT descriptor, the original images and the Bayer pattern images are transformed into different versions by blurring, scaling and rotating. The repeatability among each image is evaluated using the criteria mentioned in Section IV-B. To maintain the 'mosaic' structure, rotation on Bayer pattern images are performed by extracting the pixels with the same color from the Bayer images to form four sub-images, i.e., R, G_1, G_2 and B, rotating them separately and reorganizing them back into Bayer pattern images. Note that this process is just for generating experimental samples. Three scales are used in our experiments ($s = 3$ in Equation (5)). Euclidean distance is used as the distance measurement between a pair of matching

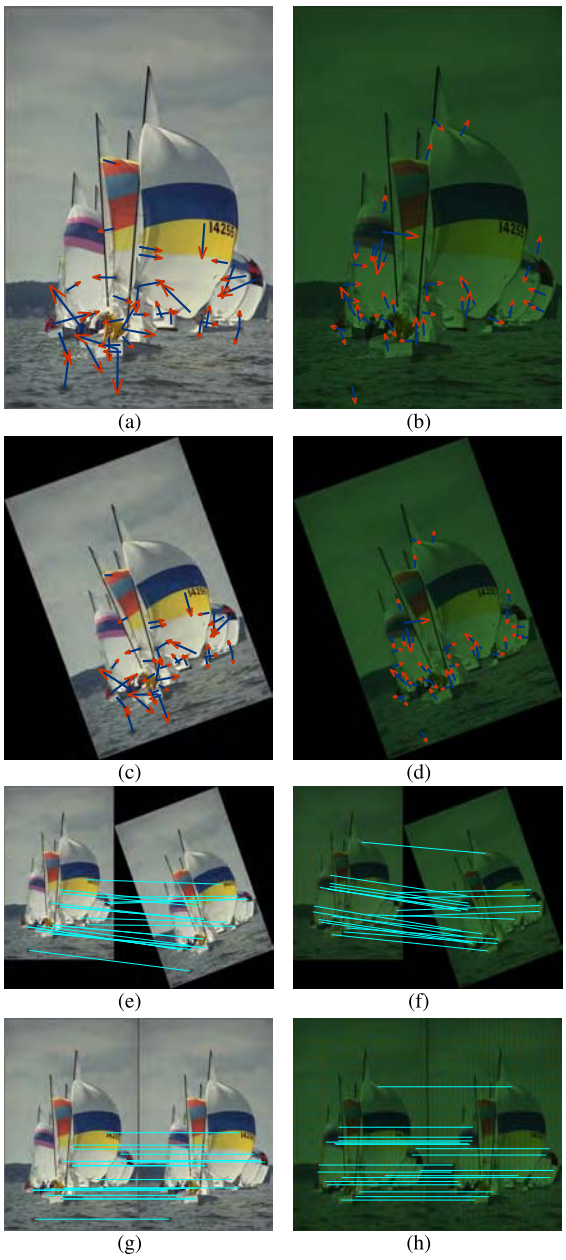


Fig. 15. (a)-(d): Part of the SIFT descriptors in the original Kodim09 image, its Bayer version and corresponding 20-degree-rotate version. (e)-(f): Twenty matches in (c) and (d). (g)-(h): Projecting the matches in (e) and (f) back to the location in (a) and (b) by homography matrix H.

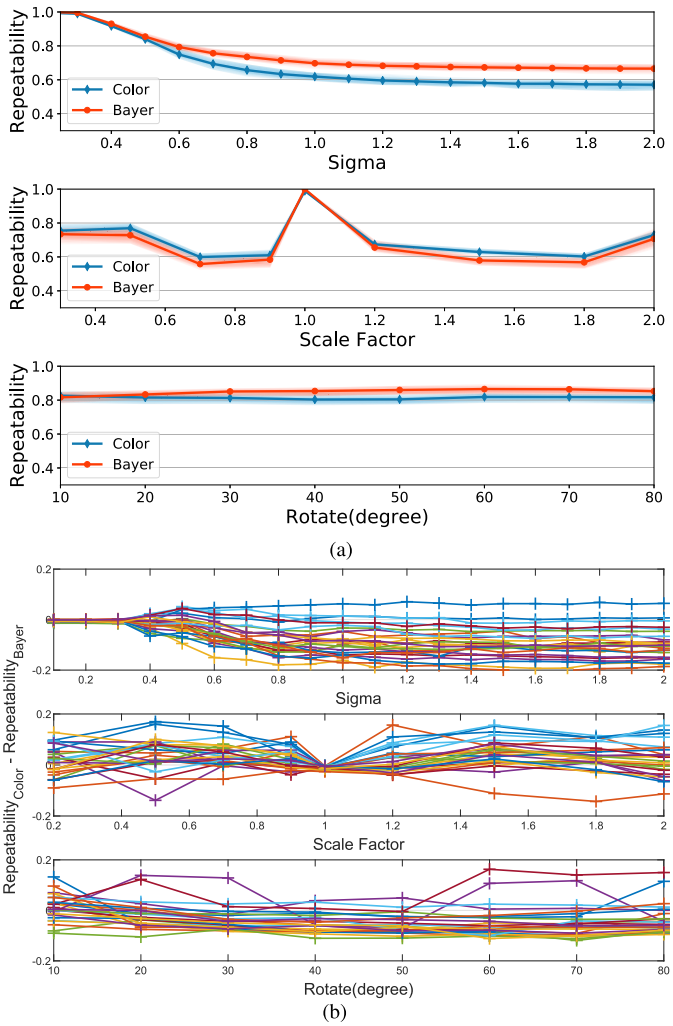


Fig. 16. (a) Average repeatability of SIFT descriptor after blur, scale change and rotate on Kodak dataset. The shaded area indicates the 25-75% quantile band. (b) $\text{Repeatability}_{\text{Color}} - \text{Repeatability}_{\text{Bayer}}$ for each image.

pixels and threshold t in (34) is set to 3. Estimation of H from Bayer and Gray images is highly accurate and the difference between both approaches is negligible (see supplemental material).

Fig. 16(a) depicts the average repeatability scores for both color and Bayer version. As it can be observed, the curves in Fig. 16(a) are very close to each other, with the Bayer version performs slightly better in the blur and rotation experiment while slightly worse in the scale experiment. Fig. 16(b) illustrates the difference of repeatability for each image. The repeatability on Bayer pattern images is generally better than that on color images for the Blur operation. This may because the stages in the ISP pipeline will introduce some extra 'blur' effects. For scale and rotate operation, outliers often appear in images with rich textures, e.g., the top left image in Fig. 13(b), where failure cases are more likely to appear.

4) Pedestrian Detection: The HOG + SVM model is used as benchmark framework to evaluate the performance of the proposed Bayer pattern image-based gradients in object detection algorithms. Fig. 17 shows the pedestrian detection results on INRIA, SHTech and PASCALRAW datasets. As we can see, the performances of detection rate versus false positive per image are very close for different versions of images.

As shown in Fig. 17(a), HOG + SVM achieves 63.56% average precision (AP) on Bayer version of the INRIA dataset, compared to 63.39% on the original INRIA dataset. The results are similar on the SHTech dataset, while the average precision in SHTech dataset is worse than that in INRIA dataset for both Bayer and color version. This is due to the difference in the number and posture of the dataset samples. In PASCALRAW datasets, the detection rate for Bayer version is also close to its color version counterpart. Therefore, the gradients extracted directly from the Bayer pattern images are robust enough to be

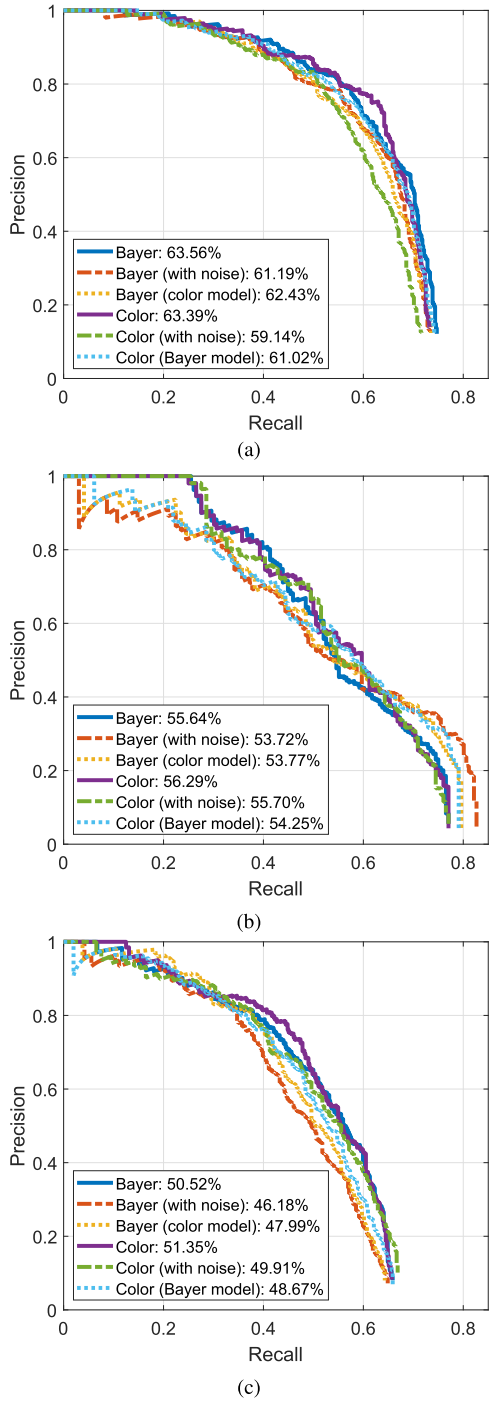


Fig. 17. Evaluation of pedestrian detection performance based on different versions of gradients using (a) the INRIA dataset, (b) the SHTech dataset and (c) the PASCALRAW dataset. Here Bayer (color model) means the model is trained on gradient from color images and tested on gradients from Bayer images while Color (Bayer model) means the opposite.

used in pedestrian detection algorithm, while the performance can be maintained.

We also conduct transfer experiments, i.e., train a classifier on gradients from color images and evaluated on gradients from Bayer images, and vice versa. From the results presented in Fig. 17, it is found that there are small decreases in performance when a detector is not trained on the same version. But the decreases are very small, which means the gradients generated from color images are very close to that generated from Bayer images.

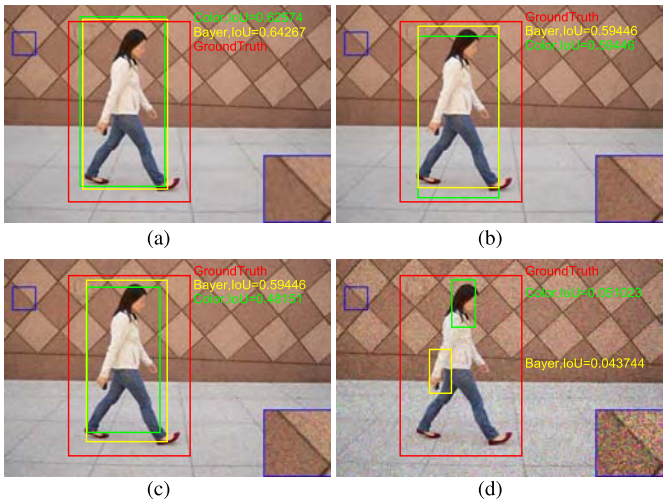


Fig. 18. The pedestrian detection results on (a) the original image, (b) noisy image with parameters of $a = 9.63 \times 10^{-4}, b = 3.43 \times 10^{-5}$, (c) $a = 4.80 \times 10^{-3}, b = 2.00 \times 10^{-4}$, (d) $a = 3.59 \times 10^{-2}, b = 3.40 \times 10^{-3}$.

The pedestrian detection performance under the influence of noise is also presented in Fig. 17. Note that in this experiment, the models are not retrained, i.e., the models trained using the noise-free images are used for pedestrian detection in noisy images. It can be found that the detection performance decreases slightly on all three datasets for both Bayer and color versions. The detection results on one of the images with different noise level are shown in Fig. 18(a)-(d). It is found that with the increase of noise level, the bounding boxes tend to be smaller. As shown in Fig. 18(d), where the severest noise parameters are applied, the model seems not working for both Bayer and color versions.

V. Discussion

The objective of computer vision is to obtain high-level understanding from images and videos. Traditional vision algorithms take fully rendered color images as inputs. However, in scenarios where color is not required, such as the gradient-based algorithms discussed in this paper, demosaicing is redundant. It not only costs computing time, but also wastes three times the storage space to get almost the same results.

It has been shown in [8] that in a conventional computer vision system consisting of an image sensor, an image signal processor and a vision processor (to run the computer vision algorithms), the image signal processor consumes a significant amount computation resources, processing time and power. For example, a well-designed HOG processor consumes only 45.3 mW to process 1080P videos at 60 frames per second (FPS) [50], while a typical image signal processor dissipates around 250 mW to process videos with the same resolution and frame rate [51]. Therefore, from the system perspective, if we can skip the ISP pipeline (or most of the ISP steps), the computational complexity and power consumption of the computer vision system can be reduced significantly. Even in some features where color information is necessary, such as integral channel features (ICF) [52] or color descriptors

in SIFT family [29], the location of demosaicing in the ISP pipeline need to be reconsidered. This is because as long as the mosaic structure is maintained, color information can be recovered whenever it is needed, through demosaicing for example. Moreover, though this paper shows that gradients extracted from Bayer pattern images are close to that from color images, the optimality of color image-based gradients extraction deserves a careful reconsideration. According to our understanding, the ISP pipeline and computer vision algorithms need to be co-designed for better performance.

This paper presents a method and corresponding analysis to extract gradient-based features from raw Bayer pattern images. But there are some limitations. The applicability of the proposed method is influenced by the relationship between gradient operators and CFA patterns. To make the proposed approach applicable, it is crucial to ensure that the gradient calculation is performed on pixels from the same color channel, i.e., subtract or add operations are performed on the same color channel, and the coefficients of the subtract or add terms in the gradient operator are equal such that the gradients compute from R/B channel can be approximated to G channel. Moreover, although the method hold in flat areas and some non-smooth texture areas when computing gradients, there are failure cases which not satisfy the model's assumption.

VI. Conclusion

In this paper, the impact of demosaicing on gradient extraction is studied and a gradient-based feature extraction pipeline based on raw Bayer pattern images is proposed. It is shown both theoretically and experimentally that the Bayer pattern images are applicable to the central difference gradient-based algorithms with negligible performance degradation. The color difference constancy assumption, which is widely used in various demosaicing algorithms, is applied in the proposed Bayer pattern image-based gradient extraction pipeline. Experimental results show that the gradients extracted from Bayer pattern images are robust enough to be used in HOG-based pedestrian detection algorithms and SIFT-based matching algorithms. Therefore, if gradient is the only information needed in a vision algorithm, the ISP pipeline (or most of the ISP steps) can be eliminated to reduced the computational complexity as well as power consumption of the systems.

References

[1] *Open Image Signal Processor (openISP)*. Accessed: Apr. 4, 2021. [Online]. Available: https://github.com/cruxopen/openISP

[2] O. Losson, L. Macaire, and Y. Yang, "Comparison of color demosaicing methods," *Adv. Imag. Electron Phys.*, vol. 162, pp. 173–265, Oct. 2010.

[3] B. E. Bayer. *Color Image Filter*. Accessed: Sep. 10, 2019. [Online]. Available: https://patents.google.com/patent/US3971065A/en

[4] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 34–43, Jan. 2005.

[5] F. Heide *et al.*, "FlexISP: A flexible camera image processing framework," *ACM Trans. Graph.*, vol. 33, no. 6, p. 231, 2014.

[6] J. Sporring and J. Weickert, "Information measures in scale-spaces," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 1051–1058, Apr. 1999.

[7] A. Omid-Zohoor, C. Young, D. Ta, and B. Murmann, "Toward always-on mobile object detection: Energy versus performance tradeoffs for embedded HOG feature extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 5, pp. 1102–1115, May 2018.

[8] M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the imaging pipeline for computer vision," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 975–984.

[9] H. Blasinski, J. Farrell, T. Lian, Z. Liu, and B. Wandell, "Optimizing image acquisition systems for autonomous driving," *Electron. Imag.*, vol. 2018, no. 5, pp. 1–7, Jan. 2018.

[10] Z. Liu, T. Lian, J. Farrell, and B. Wandell, "Neural network generalization: The impact of camera parameters," 2019, *arXiv:1912.03604*. [Online]. Available: http://arxiv.org/abs/1912.03604

[11] O. Losson and L. Macaire, "CFA local binary patterns for fast illuminant-invariant color texture classification," *J. Real-Time Image Process.*, vol. 10, no. 2, pp. 387–401, Jun. 2015.

[12] A. Aberkane, O. Losson, and L. Macaire, "Edge detection from Bayer color filter array image," *J. Electron. Imag.*, vol. 27, no. 1, pp. 53–66, 2018.

[13] O. Losson, A. Porebski, N. Vandenbroucke, and L. Macaire, "Color texture analysis using CFA chromatic co-occurrence matrices," *Comput. Vis. Image Understand.*, vol. 117, no. 7, pp. 747–763, Jul. 2013.

[14] A. Trifan and A. J. R. Neves, "On the use of feature descriptors on raw image data," in *Proc. 5th Int. Conf. Pattern Recognit. Appl. Methods*, 2016, pp. 655–662.

[15] A. J. Neves, A. Trifan, B. Cunha, and J. L. Azevedo, "Real-time color coded object detection using a modular computer vision library," *Adv. Comput. Sci., Int. J.*, vol. 5, no. 1, pp. 110–123, 2016.

[16] W. Zhou, S. Gao, L. Zhang, and X. Lou, "Histogram of oriented gradients feature extraction from raw bayer pattern images," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 5, pp. 946–950, May 2020.

[17] H. Can and M. Brown, "A software platform for manipulating the camera imaging pipeline," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9905, Oct. 2016, pp. 429–444.

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

[19] L. Wenmiao and T. Yap-Peng, "Color filter array demosaicking: New method and performance measures," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1194–1210, Oct. 2003.

[20] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.

[21] 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.

[22] R. Jain, R. Kasturi, and B. Schunck, *Machine Vision*, vol. 9. New York, NY, USA: McGraw-Hill, 1995.

[23] P. Eliason, L. Soderblom, and P. Chavez, Jr., "Extraction of topographic and spectral albedo information from multispectral images," *Photogramm. Eng. Remote Sens.*, vol. 47, pp. 1571–1579, Oct. 1981.

[24] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2003, pp. I–I.

[25] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *Proc. Eur. Conf. Comput. Vis.*, vol. 1842, Jun. 2000, pp. 404–420.

[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[28] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Proc. Comput. Vis. Image Understand.*, vol. 110, pp. 404–417, Jun. 2006.

[29] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

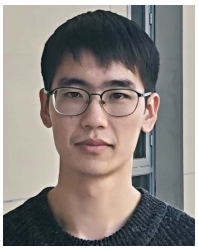
[30] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, Jan. 2009.

[31] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[32] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 99, Sep. 1999, pp. 1150–1157.

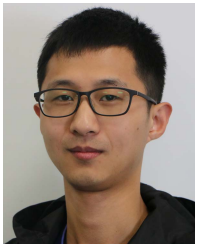
[33] R. C. Gonzalez and P. Wintz, *Digital Image Processing* (Applied Mathematics and Computation). Reading, MA, USA: Addison-Wesley, 1977.

[34] J. E. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," *The Int. Soc. for Opt. Eng.*, vol. 117, no. 7, pp. 117–125, 1997.
[35] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*, 1st ed. Boca Raton, FL, USA: CRC Press, 2008.
[36] R. W. Franzen. *True Color Kodak Images*. Accessed: Sep. 10, 2019. [Online]. Available: http://r0k.us/graphics/kodak/
[37] Jerpelea. *FreeDcam*. Accessed: Sep. 10, 2019. [Online]. Available: https://github.com/freexperia/FreeDcam
[38] A. Omid-Zohoor, D. Ta, and B. Murmann. *PASCALRAW: Raw Image Database for Object Detection*. Accessed: Sep. 11, 2019. [Online]. Available: http://purl.stanford.edu/hq050zr7488
[39] *Inria Person Dataset*. Accessed: Sep. 10, 2019. [Online]. Available: http://pascal.inrialpes.fr/data/human/
[40] C. Chen, C. Qifeng, X. Jia, and K. Vladlen, "Learning to see in the dark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2018, pp. 3291–3300.
[41] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 684–695, Feb. 2014.
[42] S. Welstead, *Fractal and Wavelet Image Compression Techniques*. (Society of Photo-Optical Instrumentation Engineers). Bellingham, WA, USA: SPIE, Jan. 1999.
[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[44] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.
[45] M. L. Uss, B. Vozel, V. V. Lukin, and K. Chehdi, "Image informative maps for component-wise estimating parameters of signal-dependent noise," *J. Electron. Imag.*, vol. 22, no. 1, Feb. 2013, Art. no. 013019.
[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[47] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, 2000.
[48] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[49] *OpenCV 2.4.13.7 Documentation*. Accessed: Sep. 11, 2019. [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html#getgaussiankernel
[50] A. Suleiman and V. Sze, "Energy-efficient HOG-based object detection at 1080HD 60 fps with multi-scale support," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2014, pp. 1–6.
[51] J. Hegarty *et al.*, "Darkroom: Compiling high-level image processing code into hardware pipelines," *Acm Trans. Graph.*, vol. 33, no. 4, pp. 1–11, 2014.
[52] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf.*, 2009, p. 91.

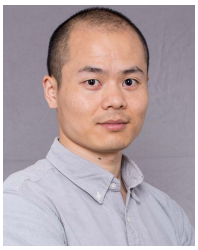


Wei Zhou (Graduate Student Member, IEEE) received the B.Eng. degree in instrument science and engineering from Southeast University, Nanjing, China, in 2018. He is currently pursuing the master's degree in electronic science and technology with ShanghaiTech University, Shanghai, China. His research interests include digital image processing and computer vision.

Ling Zhang received the B.Eng. degree in electrical engineering from Xidian University, Xi'an, China, in 2018. She is currently pursuing the master's degree with ShanghaiTech University, Shanghai, China. Since September 2018, she has been with the VLSI Signal Processing Laboratory, School of Information Science and Technology, ShanghaiTech University. Her research interests include computer vision accelerator design, especially pedestrian detection circuits and systems.



Shengyu Gao (Graduate Student Member, IEEE) received the B.Eng. degree in electrical engineering from the Wuhan University of Technology, China, in 2018. He is currently pursuing the master's degree with ShanghaiTech University, Shanghai, China. His research interest includes stereo vision-related topics.



Xin Lou (Member, IEEE) received the B.Eng. degree in electronic information technology and instruments from Zhejiang University, Hangzhou, China, in 2010, the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Sweden, in 2012, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2016. From 2016 to 2017, he was a Research Scientist with Nanyang Technological University, Singapore. Since 2017, he has been with the School of Information Science and Technology, ShanghaiTech University, where he is currently an Assistant Professor. His research interests include VLSI digital signal processing and smart vision circuits and systems.