Multimed Tools Appl (2020) 79:8599–8626
https://doi.org/10.1007/s11042-018-6396-4

I3D: a new dataset for testing denoising and demosaicing algorithms

Cristian Bonanomi[1] · Simone Balletti[1] · Michela Lecca[2] · Marco Anisetti[1] · Alessandro Rizzi[1] · Ernesto Damiani[3]

Received: 24 January 2018 / Revised: 30 June 2018 / Accepted: 10 July 2018 /
Published online: 21 July 2018

Abstract In this paper we present a dataset of images to test the performance of image processing algorithms, in particular demosaicing and denoising methods. Despite the plethora of demosaicing and denoising algorithms present in the literature, only few benchmarks are available to test their performance, and most of them are quite old, thus inadequate to represent the images captured by modern devices. The proposed dataset is composed by twenty 16 bit-depth images that can be used to test full-reference image quality metrics.

This work was partially supported by the "Ministero degli Affari Esteri e della Cooperazione Internazionale" of Italy under Grant PGR00217.

✉ Cristian Bonanomi
cristian.bonanomi@unimi.it

Simone Balletti
simone.balletti@studenti.unimi.it

Michela Lecca
lecca@fbk.eu

Marco Anisetti
marco.anisetti@unimi.it

Alessandro Rizzi
alessandro.rizzi@unimi.it

Ernesto Damiani
ernesto.damiani@kustar.ac.ae

[1] Department of Computer Science, University of Milan, via Comelico 39, 20135 Milan, Italy

[2] Fondazione Bruno Kessler, Center for Information and Communication Technology, via Sommarive 18, 38123 Trento, Italy

[3] ETISALAT BT Innovation Center Khalifa University, Abu Dhabi, United Arab Emirates

More specifically, twelve pictures have been synthetically created by means of 2D or 3D softwares, while eight images have been captured by a high-end digital camera.

Keywords Image dataset · Demosaicing · Denoising · Image quality

1 Introduction

Current cameras are highly technological devices that when the shutter button is pressed to capture a scene, execute a number of operations to produce the final image like correction for sensor nonlinearities, demosaicing, denoising, white balancing, compression, to name but a few. The final quality of the produced image strongly relies on these operations. In this paper we focus on two of the most impacting ones, demosaicing and denoising. Demosaicing is the procedure that reconstructs a color image starting from the monochromatic RAW image. Denoising is the procedure that reduces or eliminates the noise in an image.

Many denoising and demosaicing algorithms exist, working individually or jointly. In order to test their quality and to compare different methods, a common dataset is strongly required. Most of the currently used datasets showed limitations preventing to have effective quality and comparison evaluation. Some of them are old and are not representative of modern cameras results, others contain quite small images, and in most of the cases it is not clearly stated how the images have been digitized.

In this work we propose a new dataset of images, called I3D (Image Dataset for Denoising and Demosaicing), created with the purpose of overcome these limitations. Our dataset is composed by two sets of images: i) natural images, that means images of the world taken with a real high-end camera, and ii) synthetic images, namely created with a rendering software.

Our dataset is particularly suitable to exercise demosaicing and denoising algorithms as challenging benchmark. First, differently from other image datasets, the images of I3D are 16 bit depth, to reflect real situations, where raw images are typically 12-14-16 bit depth. Second, some of the images are synthetically created by means of 2D and 3D softwares. For this reason they are *denoised* and *demosaiced* by definition, as all the sub-pixels values are known, and noise due to lens or sensor inaccuracy is not present. Specifically, we created some images that look like test targets, and others which are intended to appear more natural. Third, these images have a large size, in order to capture fine details. In addition, this allows to measure the computational time of the algorithms in real scenario.

The I3D dataset can be freely downloaded at the following address: http://sesar.di.unimi.it/jdemdb/.

The remainder of this paper is organized as follows. Section 2 explores prominent denoising and demosaicing techniques showing their peculiarities. Section 3 describes how to measure the quality of the images reconstructed or enhanced by an algorithm. Section 4 reports some of the most used image datasets and Section 5 presents our I3D dataset. Section 6 provides some examples of use of I3D and comparisons and finally Section 7 gives some conclusions.

2 Demosaicing and denoising

Nowadays digital images play an important role in everyday life. Everyone has a device to take pictures, whether it is a smartphone or a professional reflex. For the user, pressing the shutter button is a simple gesture, but there are a number of operations that takes place before the final image can be displayed. In this paper we are mainly interested in two of these operations: demosaicing and denoising.

2.1 Demosaicing algorithms

When the shutter button is pressed, the light of the scene reaches the sensor and it is converted first in electrical pulses by the photodiodes present on it, then to digital values to be saved as a RAW file. This file stores the original monochrome image data captured by the camera, referred only to the luminance image. To capture a color image, a mosaic of colored filter called color filter array (CFA), is placed in front of the sensor. Each photodiode is able to capture only the information related to one of these components and for this reason can not provide all the data needed for the reconstruction of pixels. In order to obtain a three channels color image, the information of the missing pixels has to be derived from the existing ones. This procedure is performed by a demosaicing algorithm.

To be effective, a demosaicing algorithms must: (i) avoid the introduction of color artifacts, (ii) preserve the image resolution and (iii) have a low computational complexity, especially for in-camera implementation. Points (i) and (ii) are important to preserve the quality of an image, and they are features commonly measured to evaluate a demosaicing algorithm. Blurring, false color and zipper effects are artifacts usually introduced by demosaicing. The blurring causes a loss of the details and in general it is a consequence of not taking in account the direction of the edges. False colors occur when demosaicing is not accurate enough, due to a loss of correlation among the color component. Finally, the zipper effect emerges when there are colored high frequency patterns. Point (iii) has to be carefully considered when the demosaicing step has to be performed on camera, while it is less important if the calculation is done off-line, or in applications not requiring real-time or fast execution times.

Many algorithms exist to reconstruct the missing color information of the pixels. They usually estimate the missing color value by an interpolation procedure that can be adaptive, i.e.it exploits spatial features from the pixel neighborhood or non-adaptive, i.e. independent of the image visual features. Three issues are particular important in reconstruction: i) the spatial correlation for images with uniform portions, ii) the spectral correlation between the different RGB color planes and finally iii) the edge information, relevant to avoid evident artifacts.

In [1], the image channels are processed independently and the value of any missing pixel is obtained by interpolationg bilinearly the colors of adjacent pixels through an efficient 3 ×3 convolution kernel. This method works well on regions with the same color, but it generates artifacts near the edges. An enhancement comes from bicubic interpolation [54], very similar to the previous, but with a larger kernel. Differently from the bilinear and bicubic interpolations, that can provide good results in uniform region of an image, but fail in presence of edges, edge-directed interpolation is an adaptive approach that locally analyzes the edges around each target. The algorithms based on this approach (i.e. [32, 38, 45]) interpolate colors *along* directional edges, rather than *across* them. In [57] the authors explore a demosaicing technique for color filter array composed by W-RGB sensor, where W indicates that one green filter is substituted by one white (transparent) filter, to improve the sensitivity of the sensor. They propose an edge direction adaptive method which preserves edges by analyzing the correlation among white, red, green, and blue samples. The method in [89] proposes an universal demosaicing technique that reconstructs images from different kinds of Color Filter Arrays (CFAs) for enabling the comparison of the image quality. This method computes the chrominance components by two mechanisms (distance-related and edge-sensing weighting to reflect the confidence levels of the inter-pixel chrominance components, and pseudo-inverse-based estimation from the components in a window) and reconstructs the colors by a linear transformation, that combines the actual CFA color component and the two estimates of the chrominance components. A group of demosaicing algorithms, e.g. [1, 22, 40, 58, 73], assumes that the color ratio of an object uniformly colored is constant with respect to the lighting conditions under which the object is captured. These algorithms first interpolate the G channel (e.g. by means of a bilinear interpolation or an edge-directed interpolation), and then they estimate R and B channels considering the R-to-G ratio and the B-to-G ratio. Laplacian filter is used in [13] and [14] to interpolate the missing channels horizontally or vertically, choosing a direction to preserve edges. The authors in [6] proposed an improved method based on high-order interpolation, using the Sobel operator to detect edges and to calculate the gradients. Some demosaicing approaches exploit artificial neural networks, e.g. [28, 68, 76], where training images learnt offline are used to reconstruct missed colors of new pictures. The paper [63] presents the so- called sparse based radial basis function network for color image demosaicing: it introduces a sparse model of the input image and compute the weights used to minimize the reconstruction error. Sparse demosaicing encoders are used to pre-train the layers of a deep neural network, in order to reduce the system complexity in training a network from scratch. To improve the performance, adaptive demosaicing is used and optimized weights are calculated according to some features of the image in order to predict the missing color component. The method in [80] aims at achieving high quality images but with low computational time. It is based on a directional difference regression that learns the priors offline from training data. The method is a post-processing step that can be combined with other demosaicing methods. In particular, it learns offline linear regressors for the residuals between demosaiced training images and the ground truth. Then, at runtime, it applies them to the output of the demosaicing method. The multi-scale gradient approach presented in [48] extends and improves the algorithm [59]: after a first demosaicing phase, the multi-scale gradient approach corrects inaccurate interpolations, considering chrominance correlation between the channels. A polynomial interpolation-based demosaicing algorithm is proposed in [79]. This method consists of three steps: generation of the error predictors on the basis of on the polynomial interpolation, edge classification based on color differences and a final refinement step to enhance the quality of the image reducing the artifacts. In [77] a method based on an advanced non-local mean (NML) filter is introduced. The NML filter considers patches similarity, rather than pixel similarity, to reconstruct missing color pixels. A spatial locality distance is defined to favor the selection of pixels with a structure similar to that of the missing pixel in the neighborhood. Some methods work in the frequency domain. For instance, the work in [49] proposes an adaptive filtering method to preserve high-frequency details, and therefore keeps a high image quality. Wavelets are used in [51] to estimate edge directions, and in [72] to improve the performance of CFA. In particular, in [72] the authors implements a three-step method: (i) downsampling the colour channels, (ii) wavelet transformation to extract vertical and horizontal edge information and (iii) edge comparison in sub-bands. The work in [26] presents a minimization technique that combines spectral and spatial sparse representations of natural images to solve the inverse problem of recovering missing color samples from the CFA data. The spectral sparse representation describes spectral correlations and results from a physical image formation model, while the spatial sparse representation is built on an adaptive principal component analysis and describes local spatial structures of the image. The work in [82] copes with image fusion, i.e. with the process of combining information from two or more images of a scene into a single more informative image. The authors combine image fusion and demosaicing techniques, using a random panchromatic CFA, which is composed by filters of different color, not only RGB, randomly arranged. This procedure aims at avoiding the demosaicing preprocessing step before the fusion operation, in order to prevent the propagation of possible demosaicking artifacts to the fusion step. Fusion and demosaicking can be integrated because the a priori models such as multi-resolution, sparsity, inter-channel correlation that are employed in demosaicing techniques, often are used also by several image fusion algorithms.

2.2 Denoising algorithms

During the acquisition phase an image recorded by a digital camera is always affected by a certain amount of noise (i.e. thermal noise, variance in the amplifier gains, etc.). Noise can corrupt the signal during the acquisition or transmission process and in general it is described as the deviation of the recorded signal from the actual signal. The goal of denoising algorithms is to remove noise preserving the sharp structures of the image and avoiding the introduction of artifacts.

Survey papers have been recently published [35, 53, 64]. Image denoising techniques can be divided in two macro-categories: spatial domain filtering and transform domain filtering. Spatial domain filtering includes the so-called classical approaches, i.e. non-linear and linear filters. To the first belong the methods that remove the noise without any attempts to identify it. They use a low pass filtering on groups of pixels to remove the high region of the spectrum, where the noise is supposed to be. This tends to remove also the edges. Some approaches attempt to overtake this limits using weighted median filters [81] or relaxed median filters [31]. Linear filtering is optimal to reduce Gaussian noise in terms of mean square error. Nevertheless, the algorithms that fall in this category tend to remove edges and details too. An example is the Wiener filter [29, 78]. The method proposed in [47] describes a blind image denoising algorithm working in two steps. The first step estimates a signal and frequency dependent noise model, while the second performs image denoising through a multiscale adaptation of the non-local Bayes denoising method, that is a method to deal with Gaussian signal-indipendent noise. The work in [9] describes the Non-Local Means (NML) algorithm. This latter takes advantage from the fact that every small window in a natural image has many similar windows in the rest of the image. In practice, the algorithm aims at finding many samples of every detail in the image. A fast, parameter-free implementation of the algorithm is presented in [25]

The transform domain filtering category is richer, and its many methods can be subdivided according to the used basis functions. In [29] the Fast Fourier Transform (FFT) is used to low pass the image. A frequency domain filter is designed to cut-off frequencies of the noise components when decorrelated from the useful signal. Various algorithms employ wavelets to removes noise while preserving the characteristics of the signal, thanks to their properties like sparsity, multiresolution and multiscaling. The drawback of these techniques is the complexity and the high computational cost. Four different sub-groups are identified: linear filtering, non-linear threshold filtering, wavelet coefficient models, non-orthogonal wavelet transform. In [65], the author proposes a local implementation of the BLS-GSM algorithm [62], where a Gaussian Scale Mixture (GSM) is used to model a neighborhood of each wavelet coefficient, and a Bayesian least squares method (BLS) is used to estimate the noise free coefficients. A Non-Local Bayes implementation is presented also in [46]. The Wiener filter implemented in the wavelet domain [12, 69] and its improvement [86] belong to the linear filters, while [19, 20, 24] are based on non-linear coefficient thresholding. Wavelet coefficient models exploit the multi-resolution properties of the wavelets, observing the signal to different resolutions, both in a deterministic [7, 50] or statistical way [10, 67]. The work in [60] proposes the Data Adaptive Dual Domain Denoising (DA3D), which is a *last step* technique used after the application of some other denoising algorithm, as refinement step. DA3D implements a frequency domain shrinkage on shape- and data-adaptive patches, in order to adapt the processing to the underlying data. The works [11, 16, 44] belong to the last category: non-orthogonal wavelet transform. A final group includes methods derived from ICA (Independent Component Analysis) and assumes a non-gaussian noise [34, 37]. These methods employ sliding windows, and need to sample noise-free data, generally difficult to be obtained.

2.3 Joint approaches of demosaicing and denoising

Denoising and demosaicing are usually seen as two separated aspects inside the digital acquisition pipeline, and, for example, many demosaicing algorithms assume that CFA are noise-free, but it is not always the case. For this reason both denoising and demosaicing are necessary, and three possible strategies can be followed: denoising after demosaicing, denoising before demosaicing, and joint denoising and demosaicing. The first strategy is the easiest one, however, in this way, the noise is propagated during the demosaicing procedure as the algorithm mixes up the noise contribution of the three channels, and this makes difficult to characterize and remove the noise. The second approach is more difficult to be implemented because the exiting denoising algorithms cannot be applied on an incomplete image. The third strategy is more recent and proposes a joint approach to denoising and demosaicing. In [87] a spatially-adaptive denoising algorithm based on the principal component analysis (PCA) is presented. PCA is a statistical technique to reduce the dimensions of a dataset. The algorithm in [87] works on the color filter array data, using a supporting window to calculate the local statistics of the image. The aim is to eliminate the noise preserving the fine structure, as edges and details, of the image. In [4] the CFA raw data are converted to a pseudo four-channel image by rearranging pixels. The color space of this four-channel data is transformed according to the PCA. Then, a denoising algorithm is applied, and the result is transformed back in a denoised CFA raw data. Finally a demosaicing algorithm is used to generate the full color image. Another method based on PCA is presented in [36] and implements three steps: first, the PCA is performed to denoise the image captured by the Bayer pattern; second, a demosaicing based on multiscale gradients is done; third a method to suppress false color and other artifacts is implemented.

The work in [33] presents a unified method to combine demosaicing and denoising procedures. It assumes that an optimal filter for estimating a pixel value from a noisy single-color image is also an optimal filter for demosaicing given noisy sensor data. The method proposed in [56] is tailored to denoise Poisson or heteroscedastic noise, by exploiting the correlation between the different color channels. This task is accomplished by using local polynomial approximation (LPA) and intersection of confidence intervals (ICI). An approach based on space-varying filters is reported in [52] to jointly perform demosaicing and denoising. The work in [17] adopts a variational formulation in which the reconstructed image has minimal total variation under a consistency constraint. Precisely, the reconstructed color image has smooth chrominance and sharp edges while the noise is transferred to the luminance channel, which is then denoised. The method in [30] integrates denoising into demosaicing, using wavelets. It is based on Gaussian Scale Mixture (GSM) prior model with two hidden variables: the GSM scale factor and the local edge direction. A machine learning approach based on Regression Tree Fields, to perform simultaneously demosaicing in presence of noise is presented in [39]. The neural network is trained with hundreds of natural images to learn a statistical model of images and noise. Edges and corners that are computed and taken into account to avoid artifacts during the demosaicing phase. Deep learning through convolutional neural network (CNN) is used in [27] to jointly perform the demosaicing and denoising. A new metric is introduced to detect Moiré effects, in order to create a better training set. The CNN is trained on a large set of ground truth data. In [41], demosaicing is treated as an image restoration problem, that is, missing color information has to be deduced from existing data. The estimation of the model parameters is optimized by an offline training procedure, specifically trained for different sensor layouts. Without making model assumptions but learning the specific image statistics, this method aims at produce natural results.

3 Measuring image quality

The many processes involved in elaborating an image could be source of distortions during the acquisition and transmission phases. For this reason, an image quality metric (IQM) is an important tool to measure the degradation of an image. Two types of techniques are possible to assess the image quality: one involves the human beings (in general the end users) and exploiting subjective considerations; the other implements objective method to automatically compute a score.

The work in [5] recommends how to conduct appropriate subjective assessments, discussing about the viewing conditions, environment, monitor, signals, observers, methodology, etc. Subjective evaluation can be based on double stimulus if the source image (undistorted) is available or single stimulus if a reference is not present.

According to [5], at least fifteen non-expert observers are necessary, and a final mean opinion score (MOS) is computed. This is an accurate technique as closely linked to human perception.

The objective IQM are divided in i) full-reference (FR) methods, if the perfect (ideal, undistorted) image is available or ii) no-reference (NR) methods. Furthermore, there is a in-between class, iii) called reduced-reference (RR), to predict the perceptual quality of an image having only partial information available on the ideal image [75].

Considering the aim of this paper to introduce a new dataset of images to test denoising and demosaicing algorithms, FR methods provide a measure of fidelity to the original, undistorted image. Examples of FR metrics that can be used to assess the performance accuracy of denoising and demosaicing are e.g. the Mean Square Error (MSE), the peak signal-to-noise ratio (PSNR), average and maximum difference, Laplacian MSE [23].

These methods do not consider the characteristics of the human visual system (HVS), therefore they do not correlate well with subjective evaluations. A set of other metrics incorporating HVS features in their calculation, as the contrast sensitivity function (CSF), luminance contrast sensitivity or contrast masking [74], may be used.

4 Literature image datasets

Demosaicing and denoising algorithms have the goal to reconstruct missing information and reduce errors. To evaluate the performance of these algorithms by FR methods, a set of reference images are used in literature. On the web some image datasets are proposed. The three most used in particular, to test demosaicing algorithms are the Kodak [42, 43], the McMaster DB (originally IMAX) [83, 88] and the Laurent Condat's Image base [18].

The Kodak Image dataset was distributed by Kodak as a PhotoCD sampler, in the early 1990's: the Sampler contained 24 images with copyright permissions that allowed use for any image processing research purpose [42]. Some of the images were created by Kodak's professional photographers; others were selected from the winners of the Kodak International Newspaper Snapshot Awards (KINSA), a photo competition sponsored by Kodak. The work in [42] explains that the images on the Photo CD are encoded in five different resolutions, from *thumbnail* (128 x192 pixels) to *full details* (2048 x 3072 pixels). The document [42] provides with the index of the pictures, reporting the name of photographer, the location, the used film and other information. This dataset contains a lighthouse picture, that is one of the most used image for testing demosaicing algorithms: in fact, the high frequency pattern generated by the railing, is a challenging task for the algorithms.

The images in the Kodak dataset are smooth and with low saturation, while modern cameras output sharp and saturated images. For this reason the authors of [83, 88] propose a new color image dataset, the McMaster. Originally, it contained 8 high resolution color images, captured by Kodak film and digitalized. From this original set, they crop 18 sub-images (500x500 px) used to evaluate demosaicing algorithms. However, there are no specifications on the methodology used to digitalized the images.

The Laurent Condat's Image base (LC), is composed by 150 color and grayscale images [18]. The images are taken with different consumer cameras (7 or 8 Megapixels size) in jpg format, scaled to 720x540 pixels, and saved in tiff. No other processing to enhance the images was applied. The 150 images are available both in color and in grayscale. The grayscale conversion is obtained through the formula $0.299 * R + 0.587 * V + 0.114 * B$.

5 The I3D: image dataset for denoising and demosaicing

The three datasets described in Section 4 are widely used in the research community, especially the Kodak one, as they of high quality and legal to use, however they have some limits: as already anticipated, the Kodak images are not representative of modern cameras results, and the other two datasets contain quite small images without a precise specification about the digitalization phase.

To overcome these limits, we propose the I3D dataset of images (Fig. 1) that contains synthesized and natural images. Synthesized images have been produced via software, therefore all the sub-pixels values are known. Images have been created using 2D software, or a 3D scene and then rendered as a 2D image.

Natural images have been taken with an high-end reflex camera and processed as explained in detail in Section 5.2. In theory, it would be possible to avoid the use of a color filter array, using specific device, i.e. a three-CCD camera, or a monochromatic camera with a wheel provided with RGB filters. These kinds of cameras are usually employed in astrophotography, where resolution is extremely important.

5.1 Synthesized images

The synthesized images are created with four different softwares (Fig. 1a–n).

- Figure 1a has been created with Adobe Photoshop [3]. It is the *lorem ipsum* text, a classical filler text used as placeholder in graphics or presentation. The image depicts

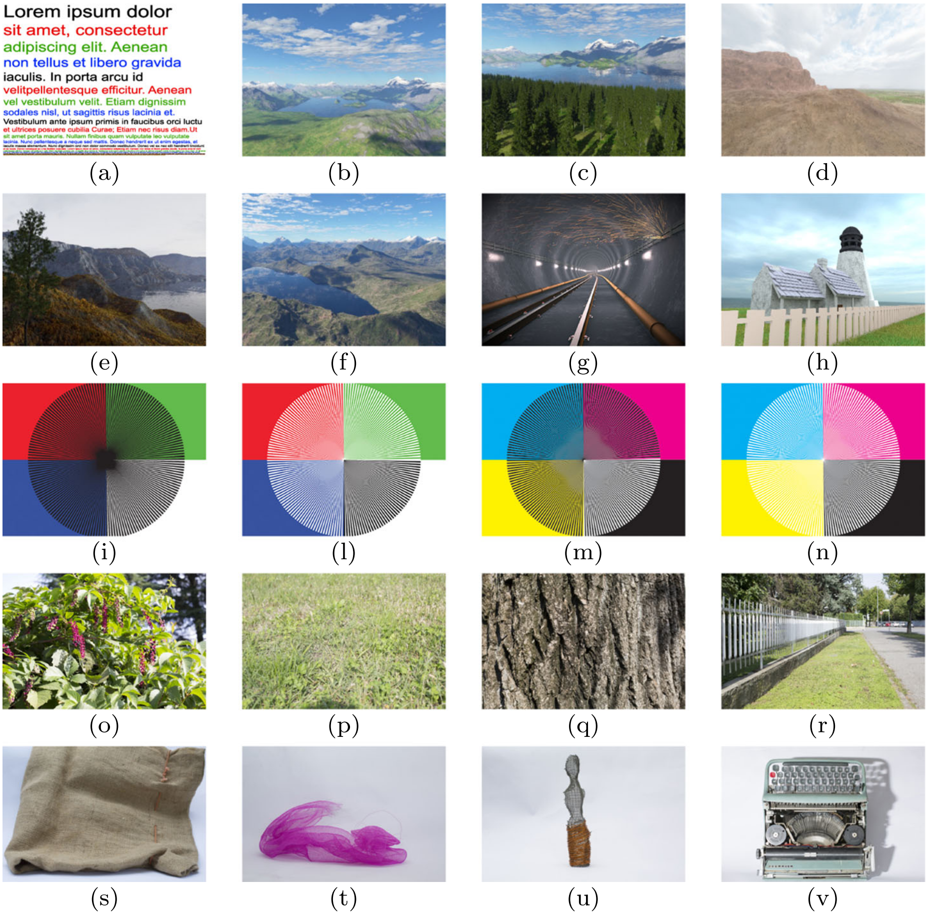

Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Aenean
non tellus et libero gravida
iaculis. In porta arcu id
velitpellentesque efficitur. Aenean
vel vestibulum velit. Etiam dignissim
sodales nisl, ut sagittis risus lacinia et.
Vestibulum ante ipsum primis in faucibus orci luctu
et ultrices posuere cubilia Curae; Etiam nec risus diam.Ut
sit amet porta mauris. Nullam finibus quam vulputate leo vulputate
(a) (b) (c) (d) (e) (f) (g) (h) (i) (l) (m) (n) (o) (p) (q) (r) (s) (t) (u) (v)

Fig. 1 The twenty images composing the proposed dataset

19 lines of the text, each line of different color, repeated as black (0, 0, 0) - a red (233, 0, 0) - a green (54, 165, 0) - a blue (0, 41, 246), on a white background, with a size-varying font, from 80 points for the first line to 2 points of the last, to include different spatial frequency.

The advantage of using text are the many shapes involved: curve lines as well as straight: horizontal, vertical and diagonal, with different inclinations. Furthermore, on a monitor big enough, all the lines are readable, and this can be a possible way to test the algorithms with human participants: if errors are introduced, words could be not (or poorly) readable.

– Figure 1b–f have been created with Terragen 4, by Planetside software [61]. Terragen is a 3D software to build and render realistic natural environments. It is possible to create procedural landscape (algorithmically created, without manual intervention, but only by setting landscape features parameters), and add natural features as clouds or seas. Some of the images contain trees or other geometrical structures with high spatial frequencies.

- Figure 1g–h have been created using Blender, an Open Source 3D software [8]. One of the picture shows a simplified reconstruction of the light-house of the Kodak Photo CD, as a tribute to one of the most used images to test demosaicing algorithms, thanks to high spatial frequencies patterns. We use the Cycles ray-trace based render engine. This render engine is called unbiased, as it uses algorithms based on the light physics, but it has high computation cost, otherwise artifacts are introduced. In our case, some noise due to the sampling mechanism is present: as a consequence, these images are suitable to test demosaicing algorithms, while they are not appropriate to test denoising techniques.
- Figure 1i–n have been generated with Adobe Illustrator [2]. Each figure represents a set of white and/or black stripes radially arranged with a rotation angle of one degree and placed on a background divided into four quadrants: RGBW or CMYK. The images were subsequently exported in eps format and imported into Adobe Photoshop to be converted in the format used in the dataset.

All the images are saved as Tiff file without compression, 16 bit depth for channel (RGB), with resolutions: 4000 x 3000 pixels.

These images can be used as ground truth to test several algorithms.

For instance, they can be artificially distorted by mosaic patterns or noise and used to evaluate demosaicing and denoising algorithms. An example is shown in Section 6.

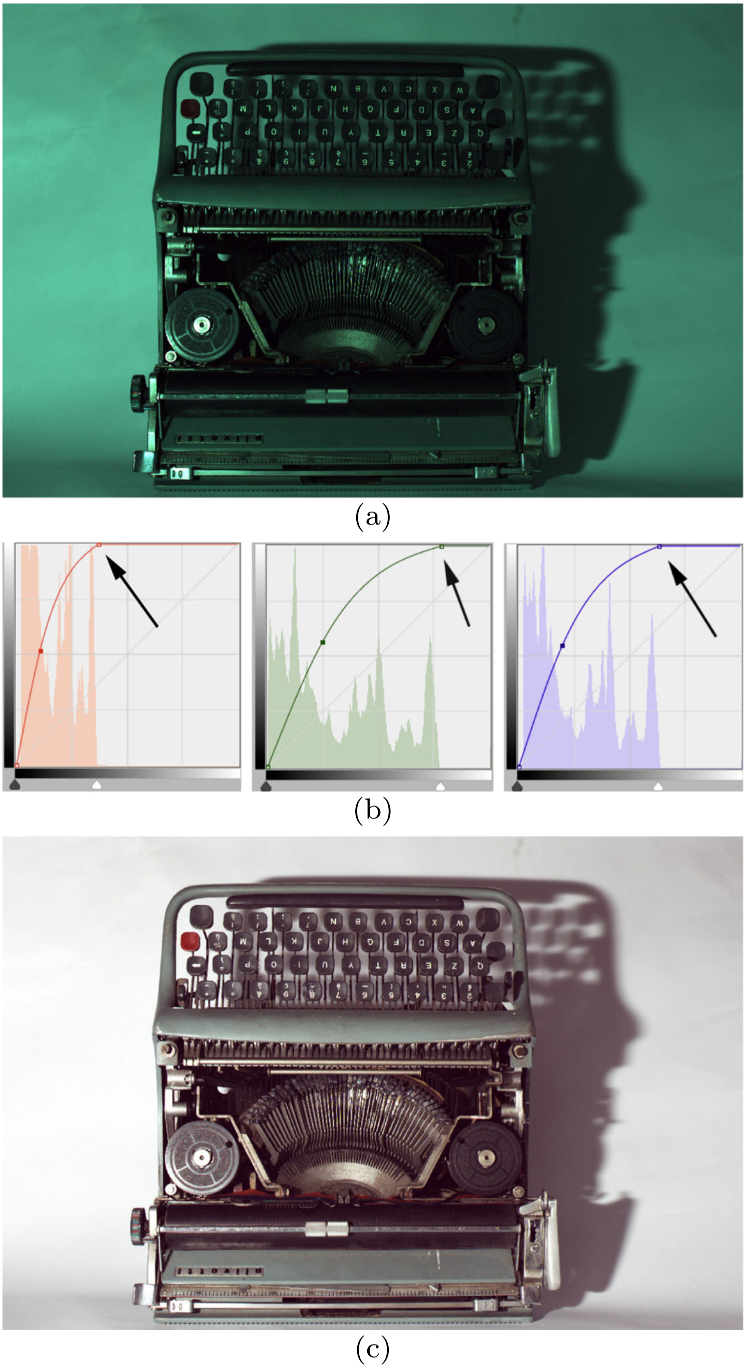
5.2 Natural images

The natural images (Fig. 1o–v) have been taken using a Canon EOS 5D Mark III (from now on simply Canon 5D), with a Canon EF 50mm f/1.4 USM lens at f/6.3 aperture, that, according to Dpreview [21] allows the better sharpness for that lens. This prevents diffraction problems, that may occur decreasing the aperture to have wider depth of field. The set is composed by natural scenes as well as objects, all with textures, high frequency patterns or small details. However, the images present also some smooth areas, result of the out-of-focus blurring.

Four different versions of the same file are available in the dataset:

- File.CR2 - original raw image, as downloaded from the camera (size: 5796 x 3870 pixels).
- File.jpg - a jpg version of the original raw, for previewing purpose (size: 5760 x 3840 pixels: note that the edges of the image are cropped).
- File.tiff - tiff extracted from the cr2 file using dcraw [15] with the following parameters -T (Tiff file format) -4 (Linear 16 bit-per-channel) -D (Document mode without scaling, totally raw). This is a gray-scale image (size: 5796 x 3870 pixels).
- File_ref.tiff - Starting from the File.tiff image, the actual pixel values were extracted, as recorded by the camera, and assigned to the correct channel (R or G or B) according to their position in the raw image. Precisely, pixels with both odd coordinates are assigned to the red channel, pixels with one even and one odd to green channel (we use only one of the two), and pixels with both even to the blue channel. The size is one quarter of the original (size: 2898 x 1935 pixels, for three channels). We consider the resulting image as reference since no interpolation has occurred.

Employing a full frame camera with a good lens, reduces the problem introduced by using a CFA, that is generating one pixels on three channels, starting from a 2x2 matrix The used camera produces 14 bit raw images. However, as in other Canon cameras tested by


(a)

(b)

(c)

Fig. 2 How to adjust the histogram and curves to obtain a good result in terms of exposure. The following operations are necessary if a white balancing algorithm is not applied. **a** Photoshop automatic color function applied on the original image. Some bright pixels do not allow to correctly stretch the histogram **b** Manual adjustment of the histograms, **c** Final result

Table 1 Minimum and maximum values of the images of the proposed dataset broken down by channels, as specified by the subscript

Index	Image name	$[m_{red}, M_{red}]$	$[m_{green}, M_{green}]$	$[m_{blue}, M_{blue}]$
(a)	lorem_ipsum.tif	[0, 65535]	[0, 65535]	[0, 65535]
(b)	terragen_1.tif	[6535, 65324]	[11672, 65163]	[15746, 64517]
(c)	terragen_2.tif	[1151, 65265]	[1632, 65078]	[1430, 64472]
(d)	terragen_3.tif	[11782, 65467]	[11290, 65475]	[10327, 65464]]
(e)	terragen_4.tif	[306, 60106]	[392, 59987]	[527, 60105]
(f)	terragen_5.tif	[0, 64983]	[0, 64680]	[14507, 64105]
(g)	underground.tif	[0, 65535]	[0, 65535]	[0, 65535]
(h)	lighthouse.tif	[451, 65535]	[517, 65535]	[345, 65535]
(i)	RGB_black.tif	[8738, 65535]	[7968, 65535]	[8224, 65535]
(l)	RGB_white.tif	[1028, 65535]	[1800, 65535]	[1800, 65535]
(m)	CMYK_black.tif	[0, 65535]	[0, 65535]	[0, 65535]
(n)	CMYK_white.tif	[0, 65535]	[0, 65535]	[0, 65535]
(o)	berry_ref.tiff	[2042, 15283]	[2058, 15282]	[2040, 15283]
(p)	grass_ref.tiff	[2079, 14186]	[2115, 15282]	[2064, 14099]
(q)	cortex_ref.tiff	[2032, 15283]	[2035, 15282]	[2026, 15282]
(r)	railing_ref.tiff	[2036, 15283]	[2046, 15282]	[2035, 15283]]
(s)	canvas_ref.tiff	[2063, 4905]	[2093, 8060]	[2064, 6498]
(t)	magenta_ref.tiff	[2475, 4099]	[2267, 6460]	[2361, 5308]
(u)	statue_ref.tiff	[2101, 5900]	[2135, 8819]	[2074, 7156]
(v)	typewriter_ref.tiff	[1822, 15328]	[2057, 15283]	[2050, 15282]

us, a *pedestal* is present, both for the black and white points, to improve image quality (i.e. to reduce noise in the dark areas). For this reason, the actual range does not cover all the $2^{14} = 16384$ values available in theory, but it is smaller. Performing some experiments,[1] we found that more than 99% of the pixels values spans in the range [2048 - 15284], making us consider this as the actual available interval and the other pixels as outsiders, due to some electrical issue of the sensor. We also note that the tiff files are 16 bit-per-channel containers of a 14 bit-per-channel images (Canon CR2 file format), for this reason they appear dark if no processing is applied. Therefore these images can also be used to test white balancing algorithms. To preview the image, we suggest to stretch the histograms individually for the three channels and apply a gamma companding. However, if there are some bright pixels, an automatic procedure (as the Photoshop automatic color function) will not work, and a manual procedure is recommended (Fig. 2).

These operations are also needed to display the final output: they and can be done before or after the application of the demosaicing/denoising algorithm under test or alternatively replaced with an ad hoc white balancing algorithm.

[1] To check the dark bias, a picture with a 1/8000 sec of exposure time, and the cap in front of the lens was taken. For the white point, we took a picture of a light source, in particular a lamp big enough to cover almost all the viewing area, for 15 sec, to be sure to saturate the sensor.

5.3 A numerical characterization of brightness, contrast and color distribution of the proposed dataset

Here, we provide a numerical characterization of the visual features of proposed dataset by reporting channel by channel the following values, that are usually altered by noise and mosaicing effects:

1. Minimum and Maximum values m and M of the values in each color channel I, i.e.:

$$m = \min_{x \in S(I)} I(x), \qquad M = \max_{x \in S(I)} I(x) \tag{1}$$

where $S(I)$ is the support of I, i.e. the set of the spatial coordinates (row, column) of I;
2. Mean Brightness B: this is the average value of the intensities of I, i.e.

$$B = \frac{1}{|I|} \sum_{x \in S(I)} I(x), \tag{2}$$

where $|I|$ indicates the number of pixels of I, and $S(I)$ is the image support, as before;
3. Multi-Resolution Contrast C: this is the mean value of the mean local contrast of a set of K re-scaled versions of I, as proposed in [66]. The value of C is computed as follows. Let $I_0, \ldots, I_{K-1}$ be the K re-scaled version of I such that $I_0 := I$ and I_k is the image I re-scaled by the factor 2^{-k}, with $k = 1, \ldots, K-1$. The mean local contrast c_k of I_k

Table 2 Values of the mean brightness B of the images of the proposed dataset broken down by channels, as specified by the subscript

Index	Image name	B_{red}	B_{green}	B_{blue}	$\overline{B}$
(a)	lorem_ipsum.tif	58628	57520	57605	57918
(b)	terragen_1.tif	32915	40830	44893	39546
(c)	terragen_2.tif	24613	29771	28987	27790
(d)	terragen_3.tif	50526	47826	45040	47797
(e)	terragen_4.tif	29382	29055	29123	29187
(f)	terragen_5.tif	28117	34056	39036	33736
(g)	underground.tif	19057	18723	19603	19128
(h)	lighthouse.tif	45561	50284	48751	48199
(i)	RGB_black.tif	31768	27641	26771	28726
(l)	RGB_white.tif	44439	40519	39644	41534
(m)	CMYK_black.tif	30766	27137	25579	27827
(n)	CMYK_white.tif	43441	40022	38387	40616
(o)	berry_ref.tiff	3110	4178	3017	3435
(p)	grass_ref.tiff	3204	4286	2910	3467
(q)	cortex_ref.tiff	2798	3275	2673	2916
(r)	railing_ref.tiff	3134	4102	3085	3440
(s)	canvas_ref.tiff	3028	3835	3131	3331
(t)	magenta_ref.tiff	3505	4945	4258	4236
(u)	statue_ref.tiff	4344	7002	5751	5699
(v)	typewriter_ref.tiff	3967	6319	5374	5220

The last column reports the mean value $\overline{B}$ of the channel brightnesses. All the values shown here have been cast to integer numbers, according to the discrete nature of the image data is the mean value of the local contrast computed in a 3×3 window and averaged over $|I_k|$, i.e.:

$$c_k = \frac{1}{|I_k|} \sum_{x \in S(I_k)} \frac{1}{8} \sum_{y \in W(x) \setminus \{x\}} |I(x) - I(y)| \tag{3}$$

where, as before, $|I_k|$ is the number of pixels of I_k, $S(I_k)$ is the support of I_k, and $W(x)$ is the 3 ×3 window centered at the pixel x of I_k. The value of C is given by the following Equation:

$$C = \frac{1}{K} \sum_{i=1}^{K} c_k. \tag{4}$$

4. Histogram Flatness F: the value of F measures the L^1 distance of the histogram H of the image channel I from the histogram representing the uniform probability density function U over the range of the image intensity. Precisely:

$$F = \frac{1}{N} \sum_{b=1}^{N} |H(b) - U(b)| \tag{5}$$

Table 3 Values of the multi-resolution contrast C of the images of the proposed dataset broken down by channels, as specified by the subscript

Index	Image name	C_{red}	C_{green}	C_{blue}	$\overline{C}$
(a)	lorem_ipsum.tif	438	487	486	470
(b)	terragen_1.tif	264	200	143	202
(c)	terragen_2.tif	519	452	275	415
(d)	terragen_3.tif	205	199	209	204
(e)	terragen_4.tif	569	549	551	556
(f)	terragen_5.tif	1267	810	367	815
(g)	underground.tif	1354	1114	1148	1206
(h)	lighthouse.tif	142	134	140	139
(i)	RGB_black.tif	475	635	534	548
(l)	RGB_white.tif	198	772	744	571
(m)	CMYK_black.tif	386	222	167	258
(n)	CMYK_white.tif	246	312	372	310
(o)	berry_ref.tiff	61	100	74	78
(p)	grass_ref.tiff	42	71	35	49
(q)	cortex_ref.tiff	134	204	108	149
(r)	railing_ref.tiff	79	138	76	98
(s)	canvas_ref.tiff	45	72	50	56
(t)	magenta_ref.tiff	23	37	30	30
(u)	statue_ref.tiff	31	48	40	40
(v)	typewriter_ref.tiff	49	83	69	67

The last column reports the mean value $\overline{C}$ of the channel multi-resolution contrasts. All the values shown here have been cast to integer numbers, according to the discrete nature of the image data

Table 4 Values of the histogram flatness F of the images of the proposed dataset broken down by channels, as specified by the subscript

Index	Image name	F_{red} $[\times 10^{-3}]$	F_{green} $[\times 10^{-3}]$	F_{blue} $[\times 10^{-3}]$	$\overline{F}$
(a)	lorem_ipsum	7.472	7.469	7.537	7.493
(b)	terragen_1.tif	3.367	3.854	3.355	3.526
(c)	terragen_2.tif	2.288	1.954	3.636	2.626
(d)	terragen_3.tif	4.524	4.753	4.710	4.662
(e)	terragen_4.tif	3.678	3.629	4.019	3.775
(f)	terragen_5.tif	2.721	2.226	3.284	2.743
(g)	underground.tif	4.034	4.187	4.084	4.102
(h)	lighthouse.tif	2.916	3.268	3.425	3.203
(i)	RGB_black.tif	7.212	7.204	7.201	7.206
(l)	RGB_white.tif	7.312	7.315	7.320	7.316
(m)	CMYK_black.tif	7.393	7.333	7.339	7.355
(n)	CMYK_white.tif	7.471	7.338	7.355	7.388
(o)	berry_ref.tiff	7.086	6.500	6.963	6.850
(p)	grass_ref.tiff	7.392	7.163	7.467	7.341
(q)	cortex_ref.tiff	7.340	7.095	7.380	7.272
(r)	railing_ref.tiff	7.180	6.858	7.096	7.046
(s)	canvas_ref.tiff	7.477	7.108	7.295	7.293
(t)	magenta_ref.tiff	7.623	7.324	7.453	7.467
(u)	statue_ref.tiff	7.442	7.181	7.338	7.320
(v)	typewriter_ref.tiff	7.217	6.553	6.799	6.856

The last column reports the mean value $\overline{F}$ of the channel histogram flatnesses

where b is an histogram bin and N is the number of bins representing the image intensity values. In this case, since the pictures of the proposed dataset are 16 bit-images, b ranges over $\{0, \ldots, 65355\}$ and thus $N = 65536$. Here, histograms are supposed to be normalized in order to range over $[0, 1]$. Peaked histograms have a high value of F.

The variability range $[m, M]$ and the values of B, C, F, broken down by images and channels are reported respectively in Tables 1, 2, 3 and 4.

Figures 3, 4 and 5 report the distributions of the mean brightness, multi-resolution contrast and histogram flatness of the images in the datasets I3D, Kodak, McMaster and Laurent-Condant respectively. In order to allow the comparison among these distributions, the intensity values of the I3D 16 bit pictures have been manipulated to range over [0, 255]. For the images from (a) to (n), this manipulation only re-scaled the channel values by 256, while for the images taken by the camera, from (o) to (v), the following transforms have been implemented for any channel I and for any pixel x:[2]

- $I(x)$ is mapped to $v(x) := \frac{I(x)-2048}{15284-2018}$;

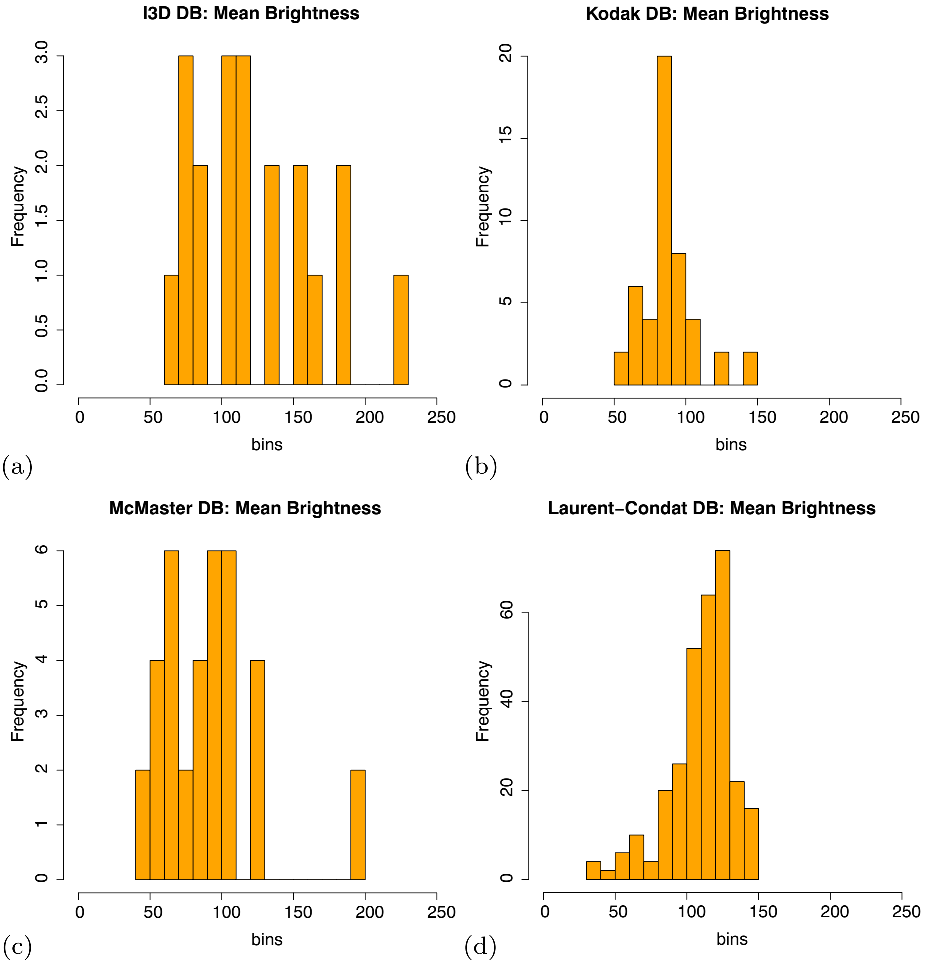
[2] These operations are needed to take account of the two pedestals, and to transform the linear raw values in sRGB.

- $v(x)$ is mapped to $u(x)$ such that:

$$u(x) = \begin{cases} 0 & \text{if } v(x) < 0 \\ 12.92\ v(x) & \text{if } 0 < v(x) \leq 0.003131 \\ 1.055\ v(x)^{1.0/2.4} - 0.055 & \text{if } 0.003131 < v(x) \leq 1 \\ 1 & \text{otherwise} \end{cases}$$

- $u(x)$ is re-scaled to range over [0, 255].

These histograms show how much varied the datasets are with respect to the considered features. Note that the Laurent-Condat dataset consists of 150 images, while the others have



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

Fig. 3 Histograms of the mean brightness for **a** I3D, **b** Kodak, **c** McMaster, **d** Laurent-Condant datasets. Note that the intensities of the brightnesses of the images of I3D have been normalized over [0, 255] to allow the comparison with the other datasets

18 - 24 images. From the comparison of the distributions of Figs. 3, 4 and 5 it comes out that:

- the images of I3D have more sparse mean brightnesses than those of the other datasets; the images with the smallest and highest mean brightness belong respectively to Laurent-Condant and I3D;
- the distribution of the multi-resolution contrast of I3D is similar to those of Kodak and McMaster, while Laurent-Condat exhibits the more stretched distribution;
- the distribution of the histogram flatness of I3D is larger than the others, meaning that some of its images have good tonal distribution, while others are characterize by brightness peaks; the distributions of I3D and Kodak are more centered than those of McMaster and Laurent-Condat.

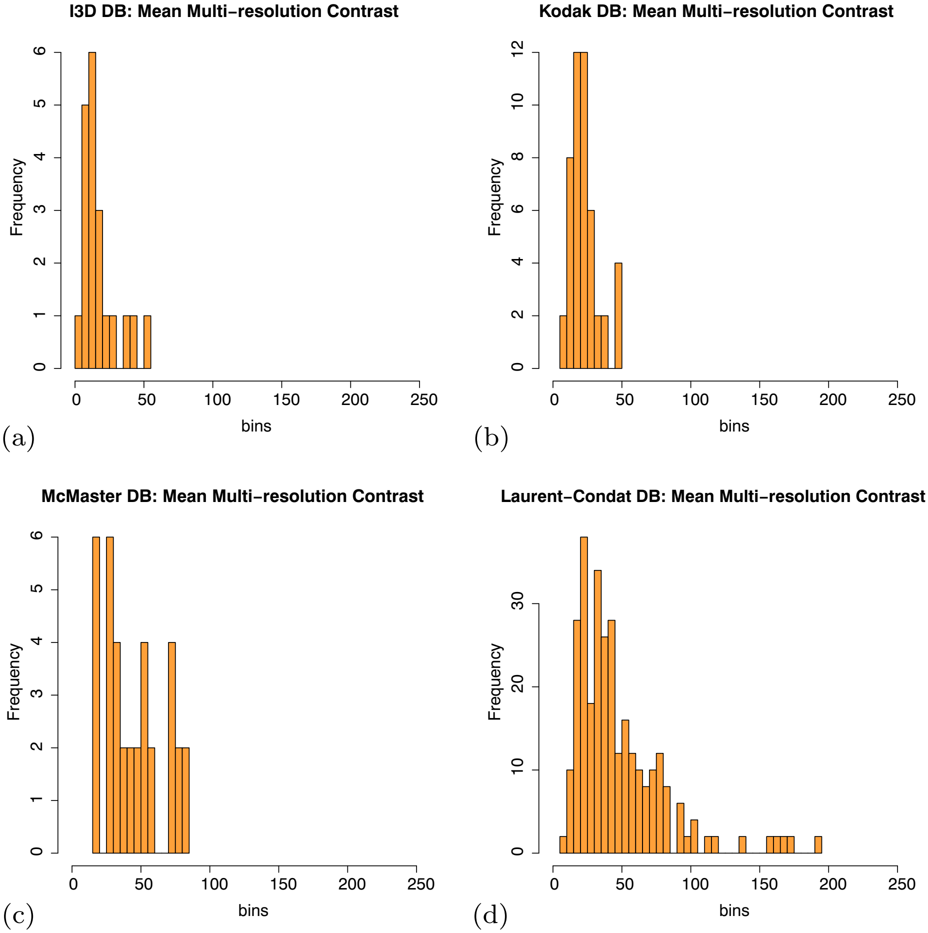


Fig. 4 Histograms of the multi-resolution contrast for **a** I3D, **b** Kodak, **c** McMaster, **d** Laurent-Condant datasets. Note that the intensities of the brightnesses of the images of I3D have been normalized over [0, 255] to allow the comparison with the other datasets

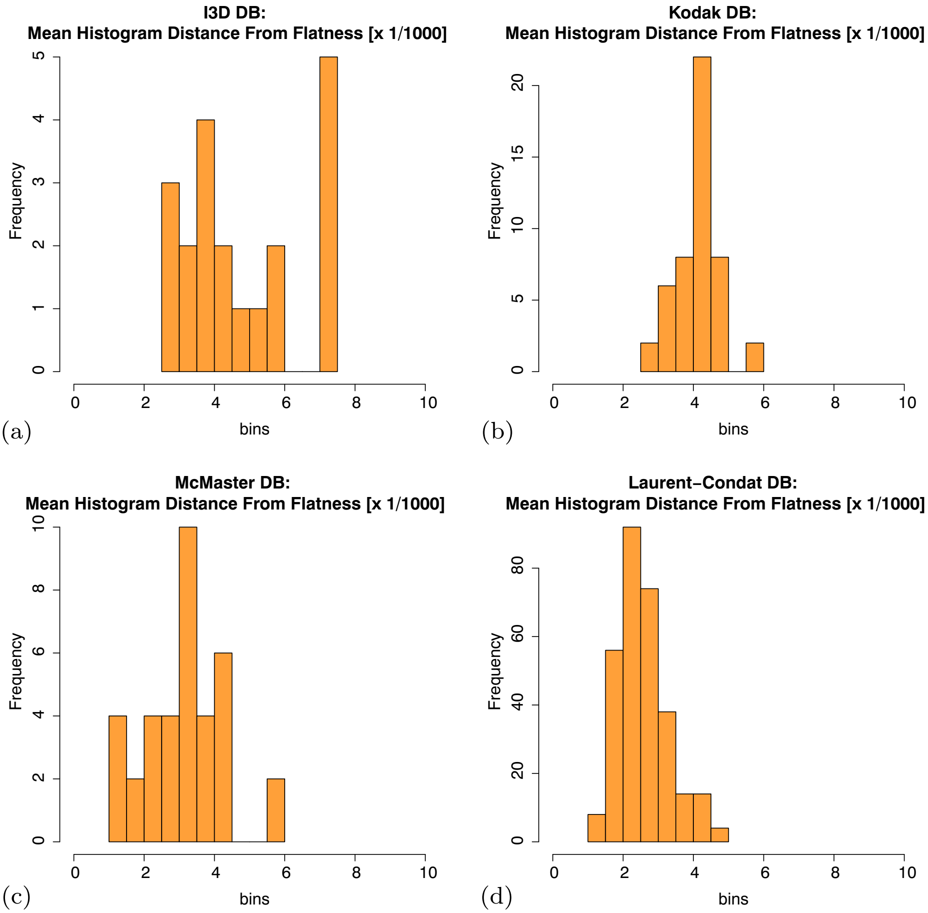


Fig. 5 Histograms of the histogram flatness for **a** I3D, **b** Kodak, **c** McMaster, **d** Laurent-Condant datasets. Note that the intensities of the brightnesses of the images of I3D have been normalized over [0, 255] to allow the comparison with the other datasets

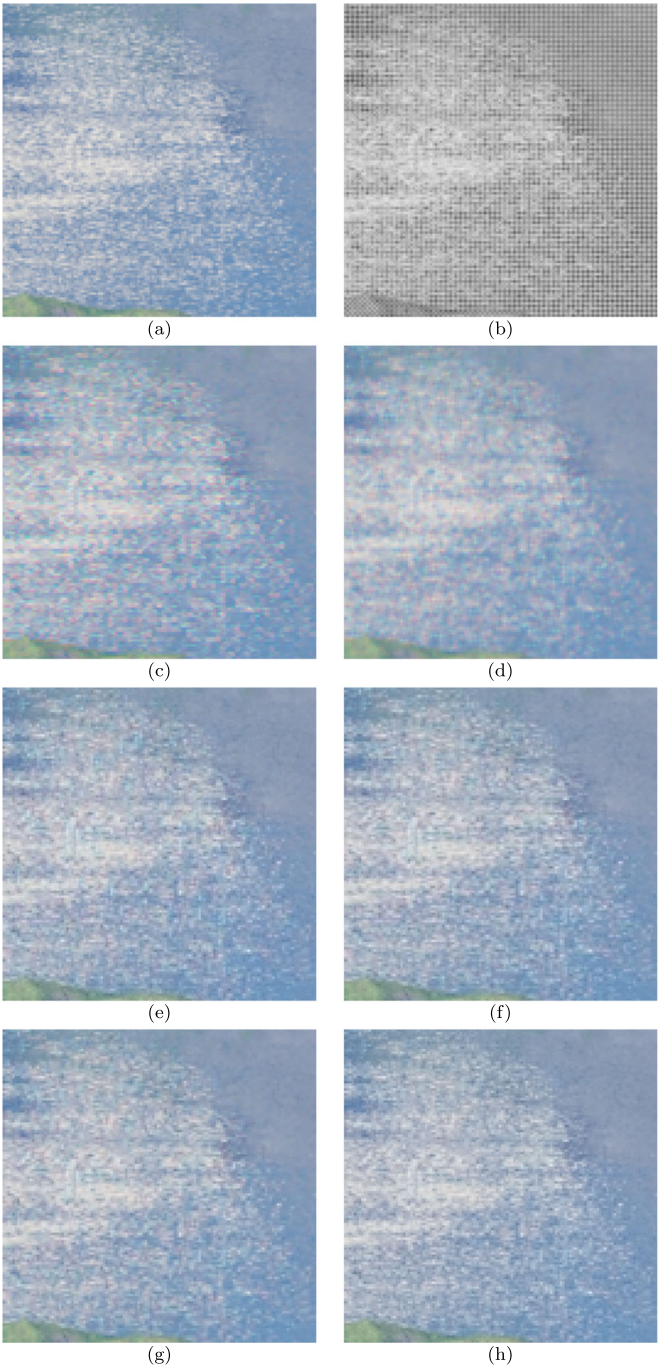
6 Examples of use

In this section we present some examples on how to use of the dataset, to test both demosaicing and denoising algorithms. Therefore, this section is intended just as an example and not an actual comparison between state of the art algorithms.

6.1 Testing demosaicing algorithms

To test demosaicing algorithms, we selected image Fig. 1b of the dataset and we downsampled it according to a Bayer Pattern (RG/GB), to simulate a raw file (Fig. 6b) and we applied six different demosaicing algorithms. Precisely:

- Nearest neighbor interpolation (Fig. 6c, detail).
- Bilinear interpolation (Fig. 6d, detail).



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 6 Example of use of the dataset on a detail of image 1b (crop and 400% magnification). **a** Original image, **b** *Mosaiced* image, **c** Nearest neighbor interpolation, **d** Bilinear interpolation, **e** Iterative method weighted-edge and color difference, **f** Directional linear minimum MSE estimation, **g** Neural Network backpropagation, **h** Adaptive inter-channel correlation

- An iterative demosaicing using weighted-edge and color-difference interpolations [70] (Fig. 6e, detail). The Matlab code was downloaded from [71].
- A demosaicing via directional linear minimum mean square-error estimation [84] (Fig. 6g, detail). The Matlab code was downloaded from [85]
- A fast implementation of the stochastic gradient descent to train a multilayer neural network [76] (Fig. 6f, detail). The demo is available at [76]
- A demosaicing algorithm with adaptive inter-channel correlation [22] (Fig. 6h, detail). The demo is available at [22]

We evaluated these six demosaicing techniques by objective measures, i.e. the Peak Signal-to-Noise Ratio (PSNR) and the mean square error (MSE), reported in Table 5. The last two algorithms are computed by using the demos available on IPOL (Image Processing On Line) journal [55]. The online processing limits the size of the custom images, therefore they have been cropped to 700x600 pixels. For this reason we report two measurements: on the whole images when available, and on the cropped images.

The objective measures are calculated on the whole image, however in Fig. 6 we selected a specific region that shows significant artifacts after the nearest neighbor interpolation, in order show how different algorithms can lead to different results.

The objective evaluations score better the last four algorithms (Fig. 6e–h). Due to their adaptive method to reconstruct missing information, they can avoid the artifacts introduced in the nearest neighbor and bilinear interpolation. A visual inspection of the selected area can confirm the result of the objective measurements.

6.2 Testing denoising algorithms

As for the previous section, our aim here is to show the possible use of our dataset on denoising algorithms. To this purpose, we selected four algorithms from IPOL [55], where online demos were available. The original image has been corrupted with gaussian noise (the same amount for all the algorithms, the parameter to introduce noise was set to ten a detail can be seen in Fig. 7b). Also in this case, the images have been cropped to 700x600 pixels. The four algorithms used are (Table 6):

- A *blind image denoising algorithm* that, in a first step, estimates a signal and frequency dependent noise model, and then denoises the image by a multiscale adaptation of the non-local Bayes denoising method [47] (Fig. 7c, detail).

Table 5 Objective measurements of the quality between the original image and the results of six different demosaicing algorithms

	$PSNR_t$	MSE_t	$PSNR_c$	MSE_c
Nearest neighbor interpolation	35.4	18.6	32.2	39.5
Bilinear interpolation	39.0	8.2	35.1	20.2
Iterative method weighted-edge and color diff. [70]	40.8	5.4	37.9	10.6
Directional linear min. MSE estimation [84]	41.1	5.1	38.2	9.7
Neural Network backpropagation [76]	n.a.	n.a.	39.1	7.9
Adaptive inter-channel correlation [22]	n.a.	n.a.	38.7	8.9

The subscript *t* means that the measurements are performed on the entire image, while *c* means that the images where cropped at 700x600 pixel in order to use the online demo to run the demosaicing algorithms

- The local version of BLS-GSM (Bayesian Least Squares - Gaussian Scale Mixture [65] (Fig. 7d, detail).
- The DA3D (Data Adaptive Dual Domain Denoising) algorithm [60] (Fig. 7e, detail). This is a *last step* algorithm, to improve the result of another method. The chosen leading method was a Non-Local Bayes Implementation (NLB) [46]

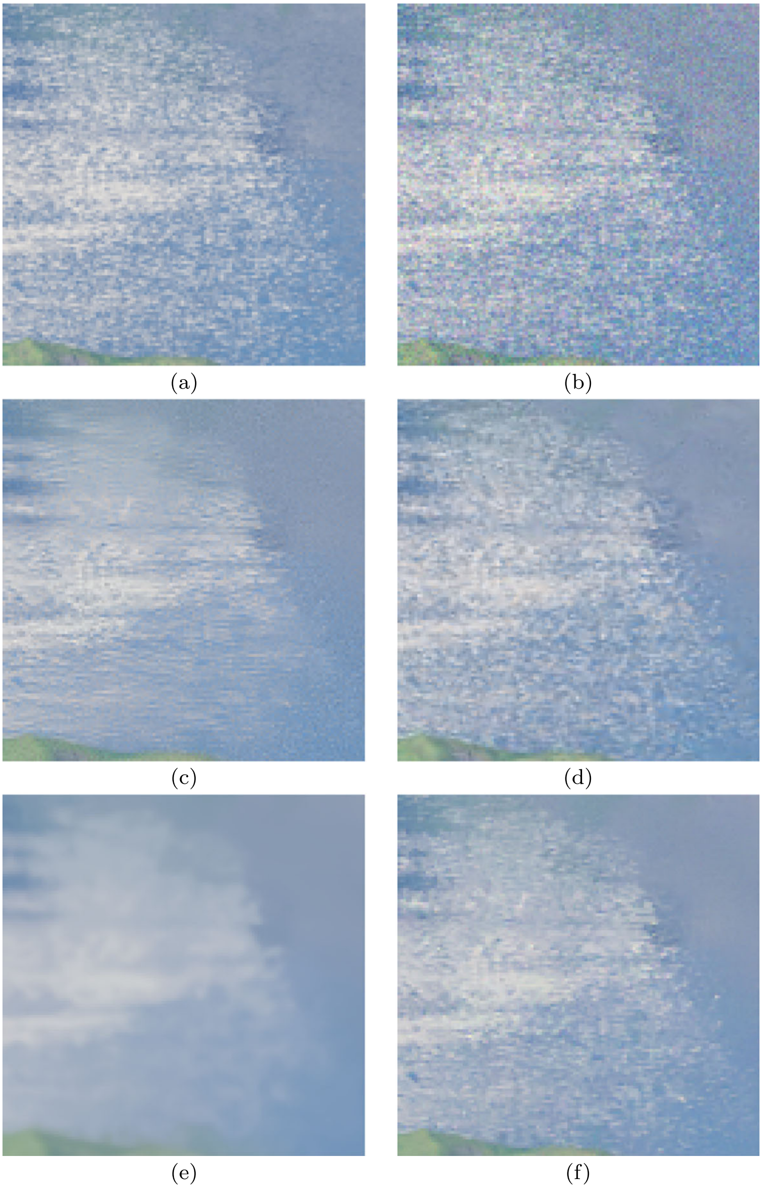

(a) (b) (c) (d) (e) (f)

Fig. 7 Example of use of the dataset on a detail of image Fig. 1b (crop and 400% magnification). **a** Original image, **b** Noisy image, **c** Nearest neighbor interpolation, **d** Bilinear interpolation, **e** Iterative method weighted-edge and color difference, **f** Directional linear minimum MSE estimation, **g** Neural Network backpropagation, **h** Adaptive inter-channel correlation

Table 6 Objective measurements of the quality between the original image and the results of four denoising algorithms

	$PSNR_c$	MSE_c
Blind Algorithm [47]	32.4	37.8
Local BLS-GSM [65]	35.5	18.4
DA3D [60] after NLB [46]	31.6	44.8
PFF-NLM [25]	34.7	21.8

The subscript *c* means that the images where cropped at 700x600 pixel in order to use the online demo to run the denoising algorithms

– The parameter-free fast (PFF) implementation [25] of the Non-Local Means (NLM) denoising [9] (Fig. 7f, detail).

7 Conclusions

In this work we have proposed a new dataset called I3D, consisting in 20 images with 16 bit color depth per channel, suitable to test image enhancement algorithms, in particular demosaicing and denoising methods. Many algorithms exist, as reported in Section 2, that can be evaluated using a image quality metric. A set of specifically designed images is necessary to use full-reference methods. The images presented here are both synthetic, created with 2D and 3D softwares, and natural, captured with a high-end digital camera. In general, they own thin details, or high frequency areas, that allow to test the performance of algorithms in the most extreme cases.

Acknowledgments We would like to thank John McCann, Prof. Bradley Lucier and Prof. Michael Kriss, that answered to our questions about the Kodak Photo CD.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Adams JE Jr (1995) Interactions between color plane interpolation and other image processing functions in electronic photography. In: IS&T/SPIE's Symposium on electronic imaging: science & technology. International Society for Optics and Photonics, pp 144–151
2. Adobe (2017) http://www.adobe.com/products/illustrator.html
3. Adobe (2017) http://www.photoshop.com/products
4. Akiyama H, Tanaka M, Okutomi M (2015) Pseudo four-channel image denoising for noisy cfa raw data. In: 2015 IEEE International Conference on image processing (ICIP). IEEE, pp 4778–4782
5. Assembly IR (2003) Methodology for the subjective assessment of the quality of television pictures. International Telecommunication Union
6. Baek M, Jeong J (2014) Demosaicing algorithm using high-order interpolation with sobel operators. In: Proc. of the World Congress on engineering and computer science
7. Baraniuk R (1999) Optimal tree approximation with wavelets. In: SPIE's International symposium on optical science, engineering, and instrumentation. International Society for Optics and Photonics, pp 196–207
8. Blender (2017) https://www.blender.org/

9. Buades A, Coll B, Morel JM (2005) A review of image denoising algorithms, with a new one. Multiscale Model Simul 4(2):490–530
10. Buccigrossi RW, Simoncelli EP (1999) Image compression via joint statistical characterization in the wavelet domain. IEEE Trans Image Process 8(12):1688–1701
11. Bui TD, Chen G (1998) Translation-invariant denoising using multiwavelets. IEEE Trans Signal Process 46(12):3414–3420
12. Choi H, Baraniuk R (1998) Analysis of wavelet-domain wiener filters. In: Proceedings of the IEEE-SP International symposium on time-frequency and time-scale analysis, 1998. IEEE, pp 613–616
13. Chung KH, Chan YH (2006) Color demosaicing using variance of color differences. IEEE Trans Image Process 15(10):2944–2955
14. Chung KH, Chan YH (2010) Low-complexity color demosaicing algorithm based on integrated gradients. J Electr Imaging 19(2):021104–021104
15. Coffin D (2017) https://www.cybercom.net/~dcoffin/dcraw/
16. Cohen I, Raz S, Malah D (1999) Translation-invariant denoising using the minimum description length criterion. Signal Process 75(3):201–223
17. Condat L, Mosaddegh S (2012) Joint demosaicking and denoising by total variation minimization. In: 2012 19th IEEE International conference on image processing. IEEE, pp 2781–2784
18. Condat L. (2016) http://www.gipsa-lab.grenoble-inp.fr/%7elaurent.condat/imagebase.html
19. Donoho DL (1995) De-noising by soft-thresholding. IEEE Trans Inf Theory 41(3):613–627
20. Donoho DL, Johnstone JM (1994) Ideal spatial adaptation by wavelet shrinkage. Biometrika 81(3):425–455
21. DPreview (2017) https://www.dpreview.com/reviews/canon-50-1p4-c16/5
22. Duran J, Buades A (2015) A demosaicking algorithm with adaptive inter-channel correlation. Image Process Line 5:311–327. https://doi.org/10.5201/ipol.2015.145. http://demo.ipol.im/demo/145/
23. Eskicioglu AM, Fisher PS (1995) Image quality measures and their performance. IEEE Trans Commun 43(12):2959–2965
24. Fodor IK, Kamath C (2003) Denoising through wavelet shrinkage: an empirical study. J Electr Imag 12(1):151–160
25. Froment J (2014) Parameter-free fast pixelwise non-local means denoising. Image Process Line 4:300–326. https://doi.org/10.5201/ipol.2014.120. http://www.ipol.im/pub/art/2014/120/
26. Gao D, Wu X, Shi G, Zhang L (2012) Color demosaicking with an image formation model and adaptive pca. J Vis Commun Image Represent 23(7):1019–1030
27. Gharbi M, Chaurasia G, Paris S, Durand F (2016) Deep joint demosaicking and denoising. ACM Trans Graph (TOG) 35(6):191
28. Go J, Sohn K, Lee C (2000) Interpolation using neural networks for digital still cameras. IEEE Trans Consum Electron 46(3):610–616
29. Gonzalez RC, Richard E (2008) Woods, digital image processing. Prentice Hall Press, Pearson
30. Goossens B, Aelterman J, Luong H, Pižurica A, Philips W (2013) Complex wavelet joint denoising and demosaicing using gaussian scale mixtures. In: 2013 IEEE International conference on image processing. IEEE, pp 445–448
31. Hamza AB, Luque-Escamilla PL, Martínez-Aroza J, Román-Roldán R. (1999) Removing noise and preserving details with relaxed median filters. J. Math. Imaging Vis. 11(2):161–177
32. Hibbard RH (1995) Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients. US Patent 5,382,976
33. Hirakawa K, Parks TW (2006) Joint demosaicing and denoising. IEEE Trans Image Process 15(8):2146–2157
34. Hyvarinen A, Oja E, Hoyer P, Hurri J (1998) Image feature extraction by sparse coding and independent component analysis. In: Fourteenth International conference on pattern recognition, 1998. Proceedings, vol 2. IEEE, pp 1268–1273
35. Jain P, Tyagi V (2016) A survey of edge-preserving image denoising methods. Inf Syst Front 18(1):159–170
36. John DM, Thomas A (2015) Combined denoising and demosaicing of cfa images. In: 2015 IEEE International Conference on signal processing, informatics, communication and energy systems (SPICES). IEEE, pp 1–6
37. Jung A (2001) An introduction to a new data analysis tool: independent component analysis. In: Proceedings of Workshop GK “Nonlinearity”. Regensburg
38. Kakarala R, Baharav Z (2002) Adaptive demosaicing with the principal vector method. IEEE Trans Consum Electron 48(4):932–937
39. Khashabi D, Nowozin S, Jancsary J, Fitzgibbon AW (2014) Joint demosaicing and denoising via learned nonparametric random fields. IEEE Trans Image Process 23(12):4968–4981

40. Kimmel R (1999) Demosaicing: image reconstruction from color ccd samples. IEEE Trans Image Process 8(9):1221–1228
41. Klatzer T, Hammernik K, Knobelreiter P, Pock T (2016) Learning joint demosaicing and denoising based on sequential energy minimization. In: 2016 IEEE International Conference on computational photography (ICCP). IEEE, pp 1–11
42. Kodac - Bradley J. (2016) Lucier webpage: https://www.math.purdue.edu/~lucier/PHOTO_CD/RIGHTS/RIGHTS.USE
43. Kodac - Rich Franzen webpage: (2016) http://r0k.us/graphics/kodak/
44. Lang M, Guo H, Odegard JE, Burrus CS, Wells RO Jr (1995) Nonlinear processing of a shift-invariant discrete wavelet transform (dwt) for noise reduction. In: SPIE's 1995 Symposium on OE/aerospace sensing and dual use photonics. International Society for Optics and Photonics, pp 640–651
45. Laroche CA, Prescott MA (1994) Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. US Patent 5,373,322
46. Lebrun M, Buades A, Morel JM (2013) Implementation of the "Non-Local Bayes" (NL-Bayes) image denoising algorithm. Image Process Line 3:1–42. https://doi.org/10.5201/ipol.2013.16
47. Lebrun M, Colom M, Morel JM (2015) The noise clinic: a blind image denoising algorithm. Image Process Line 5:1–54. https://doi.org/10.5201/ipol.2015.125. http://demo.ipol.im/demo/125/
48. Lee K, Jeong S, Choi Js, Lee S (2014) Multiscale edge-guided demosaicking algorithm. In: The 18th IEEE International symposium on consumer electronics (ISCE 2014). IEEE, pp 1–3
49. Lian NX, Chang L, Tan YP, Zagorodnov V (2007) Adaptive filtering for color filter array demosaicking. IEEE Trans Image Process 16(10):2515–2525
50. Lu J, Weaver JB, Healy DM, Xu Y (1992) Noise reduction with a multiscale edge representation and perceptual criteria. In: Proceedings of the IEEE-SP international symposium on time-frequency and time-scale analysis, 1992. IEEE, pp 555–558
51. Menon D, Calvagno G (2007) Demosaicing based onwavelet analysis of the luminance component. In: 2007 IEEE International conference on image processing, vol 2. IEEE, pp II–181
52. Menon D, Calvagno G (2009) Joint demosaicking and denoisingwith space-varying filters. In: 2009 16th IEEE International conference on image processing (ICIP). IEEE, pp 477–480
53. Motwani MC, Gadiya MC, Motwani RC, Harris FC (2004) Survey of image denoising techniques. In: Proceedings of GSPX, pp 27–30
54. Nuno-Maganda MA, Arias-Estrada MO (2005) Real-time fpga-based architecture for bicubic interpolation: an application for digital image scaling. In: 2005 International Conference on reconfigurable computing and FPGAs (ReConFig'05). IEEE, pp 8–pp
55. Online IP (2018) http://www.ipol.im/
56. Paliy D, Foi A, Bilcu R, Katkovnik V (2008) Denoising and interpolation of noisy bayer data with adaptive cross-color filters. In: Electronic Imaging 2008. International Society for Optics and Photonics, pp 68221K–68221K
57. Park J, Jang ES, Chong JW (2016) Demosaicing method for digital cameras with white-rgb color filter array. ETRI J 38(1):164–173
58. Pei SC, Tam IK (2003) Effective color interpolation in ccd color filter arrays using signal correlation. IEEE Trans Circ Syst Vid Technol 13(6):503–513
59. Pekkucuksen I, Altunbasak Y (2013) Multiscale gradients-based color filter array interpolation. IEEE Trans Image Process 22(1):157–165
60. Pierazzo N, Facciolo G (2017) Data adaptive dual domain denoising: a method to boost state of the art denoising algorithms. Image Process Line 7:93–114. https://doi.org/10.5201/ipol.2017.203. http://www.ipol.im/pub/art/2017/203/
61. Planetside (2017) http://planetside.co.uk/
62. Portilla J, Strela V, Wainwright MJ, Simoncelli EP (2003) Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Trans Image Process 12(11):1338–1351
63. Prakash VS, Prasad KS, Prasad TJC (2016) Color image demosaicing using sparse based radial basis function network. Alexandria Eng J
64. Preethi S, Narmadha D (2012) A survey on image denoising techniques. Int J Comput Appl, 58(6)
65. Rajaei B (2014) An analysis and improvement of the BLS-GSM denoising method. Image Process Line 4:44–70. https://doi.org/10.5201/ipol.2014.86. http://www.ipol.im/pub/art/2014/86/
66. Rizzi A, Algeri T, Medeghini G, Marini D (2004) A proposal for contrast measure in digital images. In: CGIV 2004 - Second European conference on color in graphics, imaging, and vision and sixth international symposium on multispectral color science. Aachen, pp 187–192
67. Romberg JK, Choi H, Baraniuk R (2001) Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. IEEE Trans Image Process 10(7):1056–1068

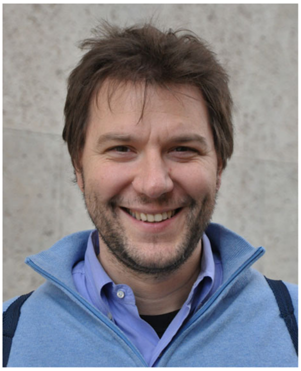
68. Shao L, Rehman AU (2014) Image demosaicing using content and colour-correlation analysis. Signal Process 103:84–91
69. Strela V (2001) Denoising via block wiener filtering in wavelet domain. In: European Congress of mathematics. Springer, pp 619–625
70. Su CY (2006) Highly effective iterative demosaicing using weighted-edge and color-difference interpolations. IEEE Trans Consum Electron 52(2):639–645
71. Su CY (2017) http://web.ntnu.edu.tw/~scy/heid_demo.html
72. Sung DC, Tsao HW (2015) Demosaicing using subband-based classifiers. Electron Lett 51(3):228–230
73. Tao B, Tastl I, Cooper T, Blasgen M, Edwards E (1999) Demosaicing using human visual properties and wavelet interpolation filtering. In: Color and imaging conference, vol 1999. Society for Imaging Science and Technology, pp 252–256
74. Thung KH, Raveendran P (2009) A survey of image quality measures. In: 2009 International Conference for technical postgraduates (TECHPOS). IEEE, pp 1–4
75. Wang Z, Simoncelli EP (2005) Reduced-reference image quality assessment using a wavelet-domain natural image statistic model. In: Human Vision and electronic imaging, vol 5666, pp 149–159
76. Wang YQ, Limare N (2015) A fast C++ implementation of neural network backpropagation training algorithm: application to Bayesian optimal image demosaicing. Image Process Line 5:257–266. https://doi.org/10.5201/ipol.2015.137. http://demo.ipol.im/demo/137/
77. Wang J, Wu J, Wu Z, Jeon G (2017) Filter-based bayer pattern cfa demosaicking. Circ, Syst, Signal Process, 1–24
78. Wiener N (1949) Extrapolation, interpolation, and smoothing of stationary time series, vol 2. MIT Press, Cambridge
79. Wu J, Anisetti M, Wu W, Damiani E, Jeon G (2016) Bayer demosaicking with polynomial interpolation. IEEE Trans Image Process 25(11):5369–5382
80. Wu J, Timofte R, Van Gool L (2016) Demosaicing based on directional difference regression and efficient regression priors IEEE transactions on image processing: a publication of the IEEE signal processing society
81. Yang R, Yin L, Gabbouj M, Astola J, Neuvo Y (1995) Optimal weighted median filtering under structural constraints. IEEE Trans Signal Process 43(3):591–604
82. Yang B, Luo J, Guo L, Cheng F (2016) Simultaneous image fusion and demosaicing via compressive sensing. Inf Process Lett 116(7):447–454
83. Zhang (2016) http://www4.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm
84. Zhang L, Wu X (2005) Color demosaicking via directional linear minimum mean square-error estimation. IEEE Trans Image Process 14(12):2167–2178
85. Zhang L, Wu X (2017) http://www4.comp.polyu.edu.hk/~cslzhang/PCA-CFA-Denoising.htm
86. Zhang H, Nosratinia A, Wells R (2000) Image denoising via wavelet-domain spatially adaptive fir wiener filtering. In: 2000 IEEE International Conference on acoustics, speech, and signal processing, 2000. ICASSP’00. Proceedings, vol 6. IEEE, pp 2179–2182
87. Zhang L, Lukac R, Wu X, Zhang D (2009) Pca-based spatially adaptive denoising of cfa images for single-sensor digital cameras. IEEE Trans Image Process 18(4):797–812
88. Zhang L, Wu X, Buades A, Li X (2011) Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. J Electron Imaging 20(2):023016–023016
89. Zhang C, Li Y, Wang J, Hao P (2016) Universal demosaicking of color filter arrays. IEEE Trans Image Process 25(11):5173–5186

Cristian Bonanomi is a research fellow at the Department of Computer Science of the University of Milan. He received a PhD in computer science in 2011 from the University of Milan. His research interests include computer graphics, human vision, color appearance models, and digital photography.

Simone Balletti received his degree in computer science in 2016, at the Dipartimento di Tecnologie dell'Informazione, University of Milano, Milano, Italy, and now is pursuing the master degree. His research interests are in computer graphics, digital photography, image processing and color.

Michela Lecca received her master's degree in Mathematics from the University of Trento. Since 2002, she works at the Research Unit Technologies of Vision of Fondazione Bruno Kessler of Trento, Italy, where she is a permanent researcher. Her research interests include machine vision, color image processing, object recognition, image retrieval, and low-level image processing for embedded systems. She is a member of the International Association for Pattern Recognition IAPR-GIRPR and of the Gruppo Italiano del Colore-Associazione Italiana Colore.

Marco Anisetti is an Assistant Professor at the Università degli Studi di Milano, Italy. His research interests are in the area of Computational Intelligence, and its application to the design of complex systems and services. Recently, he has been investigating the adoption of Computational Intelligence techniques in the area of security mechanisms for distributed systems, and software/service testing/monitoring for certification.



Alessandro Rizzi is a full professor at the Department of Computer Science at the University of Milan. Since 1990, his research has been in the field of color, digital imaging, and vision. He is particularly focused on visualization and on the visual issues related to digital imaging. He has been one of the founders of the Italian Color Group, secretary of CIE Division 8, and IS&T fellow and vice president. In 2015, he received the Davies medal from the Royal Photographic Society.

Ernesto Damiani is a Full Professor at the Università degli Studi di Milano, where he leads the SESAR research lab, and the leader of the Big Data Initiative at the EBTIC/Khalifa University in Abu Dhabi, UAE. He is the Principal Investigator of the H2020 TOREADOR project. He was a recipient of the Chester-Sall Award from the IEEE IES Society (2007). He was named ACM Distinguished Scientist (2008) and received the Stephen S. Yau Services Computing Award (2016).