JEI Journal of Electronic Imaging

RESEARCH PAPER

Color–polarization filter array image demosaicing: linear minimum mean squared error augmented by anisotropic diffusion

Ronan Dumoulin,[a] Alexandra Spote,[a] Pierre-Jean Lapray,[a,*] Jean-Baptiste Thomas,[b,c] and Ivar Farup[c]

[a]Université de Haute-Alsace, IRIMAS UR, Mulhouse, France
[b]Université de Bourgogne, Department IEM (Informatique, Électronique, Mécanique), Imagerie et Vision Artificielle (ImVIA) Laboratory, Dijon, France
[c]Norwegian University of Science and Technology (NTNU), Department of Computer Science, Gjøvik, Norway

ABSTRACT. Linear minimum mean square error can be used to demosaic images from a color–polarization filter array (CPFA) sensor. Despite its good performance, the reconstruction produces high-frequency artifacts. An additional refinement step could enable enhancement of both the quantitative and visual quality of the demosaiced image. We propose a complete demosaicing framework by first studying the model selection for linear minimum mean square error using cross-validation techniques and then optimizing the anisotropic diffusion parameters. The results show that the training model converges quickly and that the refinement step enables the reduction of the edge artifacts. We also demonstrate that the proposed demosaicing method performs better compared with a dedicated CPFA demosaicing algorithm in terms of peak signal-to-noise ratio.

 [DOI: 10.1117/1.JEI.34.3.033044]

Keywords: imaging; filter array; demosaicing

Paper 250050G received Jan. 13, 2025; revised May 14, 2025; accepted May 16, 2025; published Jun. 10, 2025.

1 Introduction

For multimodal imaging, some sensor uses the strategy of division of focal plane. For these sensors, the filter is a compound of a mosaic of filters called filter arrays. From the development of color filter arrays (CFAs), spectral filter array imaging (SFA), and polarization filter array (PFA) systems,[1,2] we observe a tendency toward their generalization to several joint optical modalities. One particular example is the color–polarization filter array (CPFA). A spatial modulation on the focal plane array permits sampling the intensities of the light field through a combination of color and polarization filters. A pixel is covered by one color filter and one polarization filter, so that it detects a specific spectropolarimetric channel among C. This provides a compact and cost-effective way to capture multimodal information in a single shot.

The SONY IMX250 MYR[3] is the most common CPFA sensor commercially available, and its spatial sampling pattern is shown in Fig. 1. It is a 12-channel sensor, which combines three color filters arranged in a quad Bayer spatial arrangement[4] and four polarization angles of analysis equally distributed between 0 and 180 deg ($p = 0$, 45, 90, and 135 deg).[5] For each pixel position, only 1 intensity measurement out of the 12 is made, so the other 11 channel values are missing.

*Address all correspondence to Pierre-Jean Lapray, pierre-jean.lapray@uha.fr

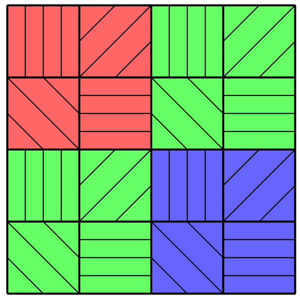

Fig. 1 CPFA filter architecture from the SONY IMX250 MYR sensor. The optical path is composed of two layers of microfilters, one above the other, where one is made of spectral filters ($c = r, g, b$) and the other of polarization filters ($p = 0$, 45, 90, and 135 deg). The 0-deg filter is at the top left of the mosaic (vertical lines). This 4×4 pixel arrangement forms what we call a superpixel and is repeated over the total photosensitive area.

To get a full resolution image, a demosaicing algorithm is applied to the images. This has been widely used for CFA, SFA, and PFA in the past, but now, there are also a few CPFA-dedicated demosaicing algorithms, which can be classified into two different categories: the filtering-based methods[6,7] and the learning-based methods.[8–15]

The filtering-based algorithms do not need a training stage and are standalone. One of these recent techniques dedicated to CPFA is the edge-aware residual interpolation[6,16] (EARI) algorithm. The term residual refers to the difference between an observed and a tentative pixel estimation. It starts from the computation of the total intensity image from polarization channels, i.e., the S_0 Stokes vector component (Stokes theory[17] is a method for describing polarization behavior of light). It exploits the redundancy of information that is inherent to PFA sensors, which have four polarization angles of analysis, whereas only three are sufficient to estimate the first three Stokes components. Thus, the total intensity S_0 is estimated in two ways: either by $S_{0,1} = I_0 + I_{90}$ or $S_{0,2} = I_{45} + I_{135}$. The two estimations are averaged for each of the four spatial directions around the pixel to demosaic (north, east, south, and west). Then, weights are computed in the four directions from the intensity difference $S_{0,1} - S_{0,2}$, giving a larger weight if the difference is small. The weighted average of the intensity estimation forms the guide image. Finally, the interpolation of missing values is done by residual interpolation,[18] using subsampled images and the computed guide. The same process is applied to each spectral channel.

The learning-based algorithms employ a training stage to define a model that is used for demosaicing. The chromatic polarization demosaicing network[11] (CPDNet) algorithm uses a convolutional neural network (CNN), trained over a set of 105 full resolution spectropolarimetric images. The network is made up of four simulation modules composed of three convolutional layers and one residual block. One simulation module is used to obtain a full red, green and blue (RGB) image from the mosaiced image. After splitting the RGB image as a function of channel color, the three other simulation modules are used in parallel to obtain a full polarization image from the full RGB image. Finally, they concatenate the result and go through three convolutional layers to obtain the full RGB polarization image. Two-step color–polarization demosaicing network[12] is another learning method formed by two subnetworks, one for color demosaicing and one for polarization demosaicing. The algorithm is a combination of the EARI and the CPDNet algorithms. The two subnetworks have the same strategy; the first step is a bilinear interpolation, and the second step is a refinement with a CNN. They use 30 images for the training. Qiu et al.[8] use the alternating direction method of multipliers (ADMM) framework to solve the inverse problem. Unlike the other algorithms, it reconstructs the Stokes vector images from the mosaiced image without estimating the individual channel intensities. They use a training procedure to obtain a linear model, with an additional noise parameter. The efficiency of the demosaicing is limited and needs several iterations. Nevertheless, this kind of algorithm has the advantage of using a limited number of data for the training. For CPFA images, the performance of the ADMM algorithm is very close to the bilinear algorithm. This paper presents a demosaicing algorithm, based on machine learning [linear minimum mean square error (LMMSE)], and a refinement algorithm based on anisotropic diffusion. We apply the algorithms to CPFA images and study the impact of the number of images used in the training for the demosaicing step. A dedicated refinement step based on anisotropic diffusion is adapted to the case of CPFA and permits to enhance jointly the signal-to-noise ratio and the visual quality of the reconstructed image. We evaluate the performance of the algorithm in terms of peak signal-to-noise ratio (PSNR) and compare it to a filtering-based technique dedicated to CPFA images.

2 LMMSE Demosaicing

In the case of the joint acquisition of color and polarization, we observe correlations among the channels.[19] This means that demosaicing algorithms can potentially benefit from these correlations, by assuming the separability of the signal autocorrelation, i.e., the inter-channel (spectral/polarization) and the intra-channel (spatial) relationships. LMMSE demosaicing is based on the principle of an inverse problem, where the missing values in a mosaic can be reconstructed by a linear combination of the neighboring pixels.

Originally, the LMMSE was applied to one-dimensional (1D) signal reconstruction, then for degradation correction in intensity images (such as noise or optical blur),[20,21] then for demosaicing of Bayer and other CFA images,[20–26] and finally for SFA[27] and CPFA.[9,10] The algorithm has the advantage of being retrained with a limited number of images (unlike deep learning methods).

LMMSE training needs both full resolution (references) and mosaiced data. The image restoration problem has long used a notation of two-dimensional (2D) images in 1D version, where the data are “vectorized” or “stacked” (2D → 1D) to facilitate computation. Let us define a 2D image $\mathbf{Y}$ with a single band of dimensions $N \times N$ by[24]

$$\mathbf{Y} = [\,\mathbf{y}_1 \quad \mathbf{y}_2 \quad \dots \quad \mathbf{y}_N\,], \tag{1}$$

where $\mathbf{y}_k$ are the column vectors of length N. The vectorized version of $\mathbf{Y}$ of dimension $N^2 \times 1$ is

$$\mathbf{y} = [\,\mathbf{y}_1^t \quad \mathbf{y}_2^t \quad \dots \quad \mathbf{y}_N^t\,]^t. \tag{2}$$

If the image $\mathbf{Y}$ has C channels (spectral, polarization, or spectropolarimetric), the channels can be stacked in the same way as the columns of the 2D image.

To demosaic an image with LMMSE, a training step is first needed to get a linear model, which is then used to demosaic the image. The imaging model can be written algebraically as

$$\mathbf{x} = \mathbf{H}\mathbf{y}, \tag{3}$$

where the vectors $\mathbf{x}$ and $\mathbf{y}$ are the vectorized versions of the mosaiced and full-resolution images, respectively, and $\mathbf{H}$ represents the mosaicing operator (but can also be considered as an image degradation function) that selects pixels from $\mathbf{y}$ to form each column of $\mathbf{x}$. The dimension of $\mathbf{H}$ is $N^2 \times (N^2 \times C)$. As the mosaicing is a linear process, it is natural to consider a linear inverse solution for demosaicing. The estimation of $\mathbf{y}$ from the LMMSE can be done by[21]

$$\hat{\mathbf{y}} = \mathbf{R_y H^t}[\mathbf{H R_y H^t}]^{-\mathbf{1}}\mathbf{x}, \tag{4}$$

where $\mathbf{R_y}$ is the autocorrelation matrix of $\mathbf{y}$. It is generally calculated on a set of full-resolution images and is therefore the expectation of the correlation for several observations of the signal.

The above model involves the manipulation of matrices of comparable size to images, which makes the algorithm difficult to implement. Because of the block shift invariance of the sensor, the restoration can be done independently per super-pixel (a periodic group of pixels in an image). Without loss of generality, the 1D rearrangement in Eq. (2) can be applied to a super-pixel instead of the full image. This advantage allows the algorithm to be usable for any image definition. The demosaicing of a super-pixel in an image can therefore be done such as

$$\hat{\mathbf{y}}_j = \mathbf{D}\mathbf{x}_j, \tag{5}$$

where j indicates the position of the super-pixel and $\mathbf{D}$ the matrix which performs the demosaicing of a super-pixel. This matrix is the same for all super-pixels in the image. The vectors $\hat{\mathbf{y}}_j$ and $\mathbf{x}_j$, i.e., the matrixed and demosaiced super-pixel, are in a “vectorized” version.

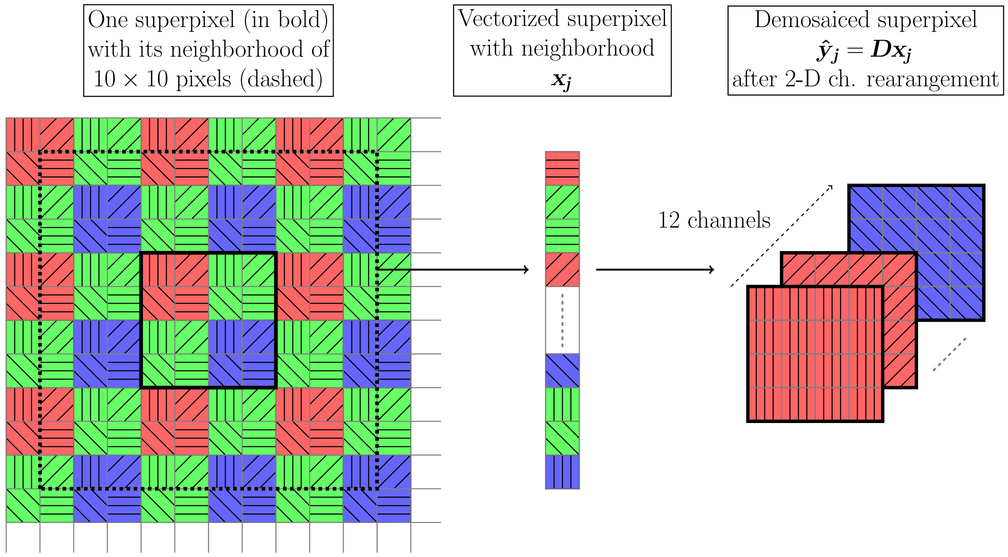

Fig. 2 Demosaicing with the LMMSE algorithm. Each superpixel of the image at position j is vectorized with its neighborhood, before applying the linear model $\mathbf{D}$.

In the case of CPFA arrangement, and to stabilize the estimate, a neighborhood of 10×10 pixels[28] around the super-pixel $\mathbf{x}_j$ is considered (and the same goes for obtaining the matrix $\mathbf{D}$, see below). In the case of CPFA with a super-pixel of 4×4 to be demosaiced into 12 bands, $\mathbf{D}$ is of size 192×100, $\mathbf{x}_j$ is of size 100×1, and $\hat{\mathbf{y}}_j$ is of size 192×1. Figure 2 shows the operations to demosaic a super-pixel.

To obtain the model $\mathbf{D}$, as for Eq. (4), the algorithm uses a training over a dataset

$$\mathbf{D} = \mathbf{V_1} \mathbf{R}_{\mathbf{y}_1} \mathbf{H}_1^t [\mathbf{H_1} \mathbf{R}_{\mathbf{y}_1} \mathbf{H}_1^t]^{-1}, \tag{6}$$

where the index $_1$ means that a neighborhood is considered for all reference data, i.e., that $\mathbf{y}_1$ contains all the vectorized super-pixels with their respective neighborhood. In this way, we can estimate the demosaiced super-pixel from the mosaiced super-pixel, taking into account its neighborhood. The matrix $\mathbf{V_1}$ is a constant matrix compound of 0 and 1 allowing neighborhood removal ($\mathbf{y} = \mathbf{V}_1 \mathbf{y}_1$).

The reconstructed images with LMMSE can produce high-frequency artifacts for some channels, e.g., grid effect distortions around edges, similar to Moiré patterns. Specifically, the grid effect becomes particularly visible in homogeneous (flat) areas of the image due to the strong frequency content of the polarization signal reverberating in the reconstructed channels. It becomes even more visible when looking at reconstructed polarization parameter images such as degree of linear polarization (DOLP) or angle of linear polarization (AOLP) [computed as in Ref. 29, see respectively Figs. 7(i) and 7(p)]. Reducing these artifacts is a way to improve visual image quality resulting from LMMSE demosaicing. To this end, we propose to enhance the quality of edges by applying an additional refinement step after LMMSE demosaicing, i.e., the anisotropic diffusion.

3 Anisotropic Diffusion for Demosaicing

Here, we explain the refinement step for demosaiced images, which is based on an anisotropic diffusion model. It comes directly after the demosaicing procedure, from which it acts as an edge restoration. This was originally produced to enhance the edges of intensity images[30] and then has been adapted to RGB images demosaiced by the Poisson algorithm.[31] We propose to readapt this method to the specific case of spectropolarimetric data, i.e., on 12-channel images instead of three.

As explained in Ref. 31, to couple the 12 channels to take into account the correlations across the channels and preserve edges better, we use a 2×2 structure tensor. This tensor of the image is defined in each position of the image[32]

$$\boldsymbol{S}(\hat{\mathbf{y}}(w,h)) = \begin{pmatrix} \Sigma_C \left(\frac{\partial \hat{y}^C}{\partial w}\right)^2 & \Sigma_C \frac{\partial \hat{y}^C}{\partial w}\frac{\partial \hat{y}^C}{\partial h} \\ \Sigma_C \frac{\partial \hat{y}^C}{\partial w}\frac{\partial \hat{y}^C}{\partial h} & \Sigma_C \left(\frac{\partial \hat{y}^C}{\partial h}\right)^2 \end{pmatrix}, \tag{7}$$

where $\hat{\mathbf{y}}(w,h)$ indexes the pixel value and C the channel ($C \in \{R_0, R_{45}, R_{90}, R_{135}, G_0, G_{45}, G_{90}, G_{135}, B_0, B_{45}, B_{90}, B_{135}\}$).

The local diffusion coefficient is defined as shown in Eq. (8), inspired by Perona and Malik[30] and generalized for the anisotropic case

$$d(\lambda_+) = \frac{1}{1 + \kappa \times \lambda_+^2}. \tag{8}$$

With λ_+ and λ_-, the eigenvalues of the gradient tensor 7 and κ a chosen constant.

The diffusion tensor $\mathbf{A}$ is defined as shown in Eq. (9), with $\mathbf{E}$ the matrix with eigenvectors of the structure tensor corresponding to the eigenvectors $\lambda_\pm$ as columns

$$\mathbf{A} = \mathbf{E}^{\mathbf{T}} \operatorname{diag}(d(\lambda_+), d(\lambda_-))\mathbf{E}. \tag{9}$$

Equation (10) is the anisotropic diffusion equation

$$\frac{\partial \hat{\mathbf{y}}^C}{\partial t} = \nabla \times (\mathbf{A}\nabla \hat{\mathbf{y}}^C). \tag{10}$$

For the discretization, we use the explicit Euler method for the time integration and 3×3 convolutions with the following kernels to compute the gradients

$$\mathbf{f_w} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{f_h} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{11}$$

where $\mathbf{f_w}$ and $\mathbf{f_h}$ are the horizontal and vertical gradient kernels, respectively.

The anisotropic diffusion algorithm has three parameters:

- Number of iterations: number of times that the diffusion operation is repeated.
- α is the time step that affects diffusion; the higher α, the greater the diffusion effect.
- κ is the impact edge preservation; the higher κ is, the less there is diffusion across the edges.

Anisotropic algorithm takes four inputs, the demosaiced image and the three parameters. First, it computes the diffusion tensor $\mathbf{A}$ taking into account the value of κ. Next, the diffusion equation is applied several times (depending on the number of iterations). For each time, the factor α is applied to increase or decrease the effect of the diffusion. The final image is then obtained.

4 Experimental Protocol

4.1 Database

A recent and up-to-date review of polarization imaging datasets is available.[33,34] There are only three RGB polarization image datasets that do not use a mosaic sensor to capture the full-resolution spectropolarimetric images.[6,11,35] Nevertheless, datasets from Wen et al.[11,35] exhibit non-uniform degree of focus among all channels (demonstrated in Ref. 10), probably due to the optical configuration of the three-charge coupled device (CCD) acquisition setup. This is somehow undesirable for the simulation of mosaiced data because there are no reasons to have this behavior in CPFA images. Thus, we decided to use the database of images from Morimatsu et al.[6] It is composed of 40 spectropolarimetric images of 768×1024 pixels, each of them having 12 full-resolution channels. The channels are a combination of three color channels ($c \in \{R, G, B\}$) and four polarization angles of analysis, equally distributed between 0 and 180 deg ($p \in \{0, 45, 90, \text{and } 135 \text{ deg}\}$). Channels are defined by $I_{c,p}$. Scenes of this database have been captured with a three-CCD RGB camera with a rotating polarizer with four orientations.

4.2 Model Selection for LMMSE

Algorithm 1 is developed to evaluate the efficiency of the algorithm when the number of images increases. In the following, we will refer to this method as the convergence test method. The

Algorithm 1 Convergence test method

INPUT: Database

OUTPUT: PSNR (μ, σ)

Select 12 images for learning

Select 28 random images for test

for $i = 1:12$ **do**

  Learn **D** matrix with i images

  Demosaic the 28 test images

  Compute PSNR for the test group

  Compute the PSNR (μ, σ)

end for

principle is to divide the dataset into two groups, one for learning and one for testing. Images intended for training should be different from those used for testing to avoid any bias in the evaluation. For the first iteration, the training is done with one image, and then the 28 test images are demosaiced. At each iteration, the number of training images is increased by 1, up to 12, and the demosaicing is performed at each iteration with the same 28 images.

To confirm that 12 images for learning are enough, we use a second cross-validation method based on the K-fold technique.[36] In the following, we will refer to this method as the K-fold method. This cross-validation makes it possible to draw several sets of validation from the same database and thus obtain a robust evaluation. The principle is to divide the dataset of images into K groups. $K-1$ groups are used to train the algorithm, and one group is used to test the algorithm. There are K iterations so that each individual group serves once as a test group. In our case, we take $n = 24$ random images from the database. To vary the number of images used for training, we vary $K \in \{2, 3, 4, 6, 8, 12, 24\}$, which gives respectively 12, 16, 18, 20, 21, 22, and 23 learning images. The last case where $K = n = 24$ is a special case of the K-fold and corresponds to a leave-one-out cross-validation. Algorithm 2 shows the different steps of the K-fold experiment.

Algorithm 2 K-fold method

INPUT: Database

OUTPUT: PSNR (μ, σ)

Select $n = 24$ random images

for $k = 2:24$ **do**

  if $24/k$ is an integer **then**

    Create k groups of $24/k$ images

    for $i = 1:k$ **do**

      Select the group i for test and other(s) for learning

      Learn **D** matrix with the learning group(s)

      Demosaic the test group

      Compute PSNR for the test group

    end for

    Compute the PSNR (μ, σ)

  end if

end for

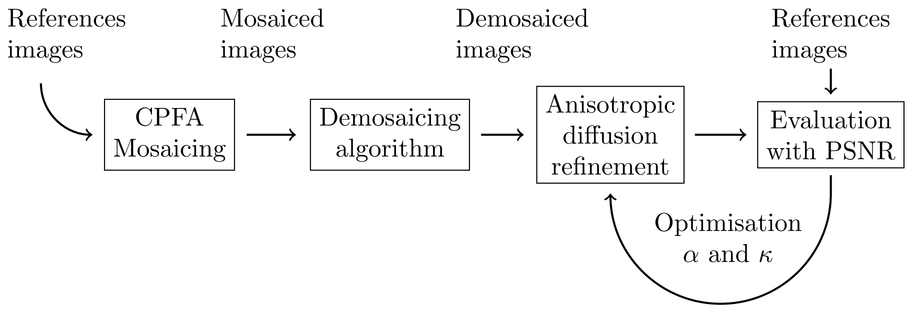

Fig. 3 Processing pipeline used to optimize the anisotropic diffusion parameters.

4.3 Parameter Selection for Anisotropic Diffusion

Figure 3 shows the processing pipeline we use to optimize the anisotropic diffusion algorithm. For parameter selection, we only use images of the training dataset. The first step is to mosaic the full-resolution reference images to obtain mosaiced images. Figure 1 shows the spatial arrangement of the filters used to mosaic the images. The second step applies a demosaicing algorithm. In our case, we selected three algorithms: bilinear, EARI,[6] and LMMSE. The third step is to apply the anisotropic diffusion algorithm. The fourth step computes the mean PSNR over a set of refined images. This step allows optimizing the parameters (α and κ) of the anisotropic diffusion algorithm. The optimization of κ and α is realized using the MATLAB function *fminsearch*,[37] by minimizing the cost function that is the inverse of the PSNR function.

We have chosen to fix the number of iterations because we experimented that it has a similar impact as α. The number of iterations is fixed to 50 for the three algorithms, this is the same number used in Ref. 31. The initial values of α and κ are respectively 0.24 and 1000 (the same for the three algorithms). We chose these values because they were the ones used in the article.[31] Steps 3 and 4 are repeated to obtain the values of α and κ, which give the best PSNR value.

5 Results and Analysis

5.1 Metric

To measure the demosaicing quality, two methods are often used: PSNR and structural similarity index measure (SSIM). The two methods use a reference to assess the demosaicing quality. By definition, the LMMSE algorithm is optimized for PSNR. To verify the impact of the identified bias, we have computed the correlation between the PSNR and SSIM results. The aim is to verify that the results obtained by either measuring PSNR or SSIM are correlated.

We compute the correlation between PSNR and SSIM on a set of color and spectropolarimetric images (PSNRAll and SSIMAll) and gray level images (PSNRGRAY and SSIMGRAY) for 28 test images. The values of PSNRAll and SSIMAll correspond to the means over the 12 channels. The RGB images are computed from the $S_0 = \frac{I_0+I_{45}+I_{90}+I_{135}}{2}$ intensity components for each spectral channel. To convert RGB images to gray level, the MATLAB function *rgb2gray* is used.

Figure 4 shows the correlation results. It is computed over the 28 images. The correlation between PSNR and SSIM for spectropolarimetric images is 0.76, and the p-value is 3×10^{-6}. For the gray version of images, the correlation is 0.77, and the p-value is 1.4×10^{-6}. These results verify that the PSNR and SSIM values are correlated for the set of images, i.e., the PSNR or the SSIM can be used interchangeably to assess the quality of the results. That is why we thought reasonable to use only PSNR to evaluate the LMMSE demosaicing results in the following.

5.2 Quantitative Results

Figure 5 shows the average PSNR values and standard deviations obtained with the convergence test method. We can see that the PSNR values stabilize with a relatively low number of images. The conclusion is that 12 images for learning are enough with the Morimatsu dataset. The PSNR values are higher for the four green channels than for other channels, probably because the green channels are spatially oversampled. This oversampling allows more information to be available to reconstruct the data lost during mosaicing.

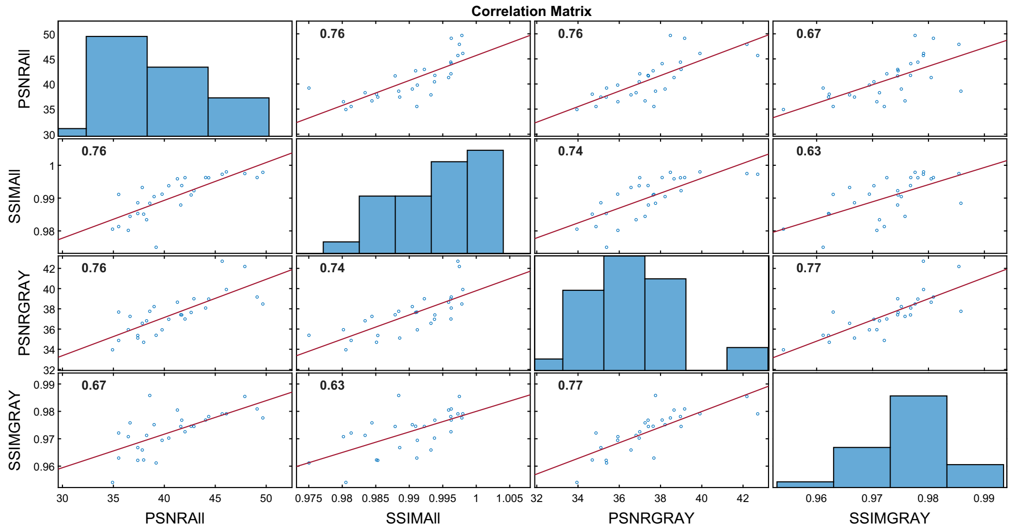

Fig. 4 Correlation between PSNR and SSIM.

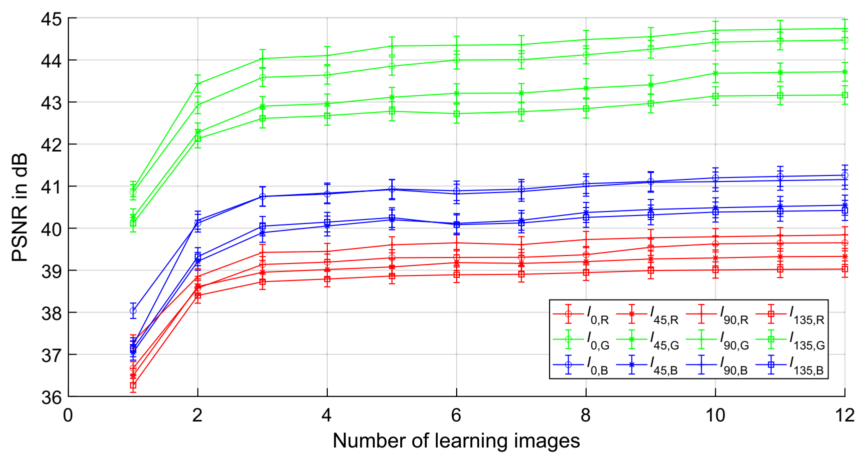

Fig. 5 PSNR as a function of the number of images to train the LMMSE model. A factor of 0.05 is applied to the standard deviation for better readability.

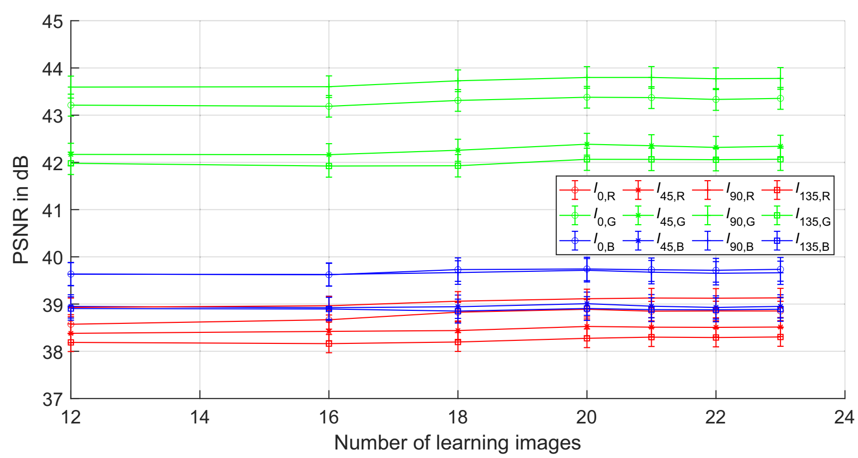

Fig. 6 PSNR as a function of the number of images to train the LMMSE model. A factor of 0.05 is applied to the standard deviation for better readability.

Figure 6 shows the average PSNRs and standard deviations obtained with the K-fold method. It can be seen that the variations in PSNRs are small for numbers of training images from 12 to 23. This means that the model has already converged. The PSNRs for the green channels are supposed to be higher than those of the blue and red channels because the green channels are sampled more densely than the others. This oversampling allows more information to be available to reconstruct the data lost during mosaicing. This confirms that the algorithm converges quickly and therefore does not need many images to perform well. It should be noted that the global PSNR averages are different compared with the convergence test method. As the scenes used for the tests are different for the two methods, the results vary depending on the image statistics.

Table 1 shows the optimal values of κ and α obtained after the optimization step for bilinear, EARI, and LMMSE. We can note that, in the case of the bilinear algorithm, the value of α (the time-step for solving the numerical equation) is near the theoretical upper limit for the stability

Table 1 Optimal parameter values for the refinement step (α and κ—anisotropic diffusion) applied to each of the following demosaicing methods: bilinear, EARI, and LMMSE.

Parameter	Bilinear	EARI	LMMSE
α	0.2695	0.057	0.008
κ	1.2431×10^4	1.6781×10^7	2.7891×10^3

Table 2 Average μ and standard deviation σ of PSNR for bilinear, EARI, and LMMSE algorithms without and with anisotropic diffusion refinement.

							With refinement—anisotropic diffusion (optimized with PSNR)					
	Bilinear		EARI		LMMSE		Bilinear		EARI		LMMSE	
Channel	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
$I_{0,R}$	35.36	4.45	38.15	4.48	39.65	3.85	**36.51**	4.23	**38.32**	4.45	**39.92**	3.91
$I_{0,G}$	38.11	4.38	43.58	4.53	44.47	4.19	**40.51**	4.44	**43.99**	4.48	**44.66**	4.18
$I_{0,B}$	36.10	4.88	40.99	5.14	41.26	4.77	**37.54**	4.95	**41.18**	5.03	**41.55**	4.84
$I_{45,R}$	35.04	4.43	37.64	4.55	39.33	3.82	**36.12**	4.24	**37.78**	4.55	**39.53**	3.89
$I_{45,G}$	37.66	4.35	42.49	4.64	43.72	4.45	**39.91**	4.46	**42.82**	4.63	**43.94**	4.46
$I_{45,B}$	35.71	4.92	40.29	5.28	40.55	4.77	**37.08**	5.02	**40.45**	5.22	**40.92**	4.86
$I_{90,R}$	35.47	4.42	38.23	4.48	39.84	3.91	**36.58**	4.17	**38.38**	4.45	**39.95**	3.94
$I_{90,G}$	38.20	4.43	43.80	4.63	44.75	4.27	**40.64**	4.53	**44.26**	4.59	**44.87**	4.26
$I_{90,B}$	36.10	4.92	41.11	5.27	41.16	4.60	**37.62**	5.07	**41.33**	5.26	**41.48**	4.63
$I_{135,R}$	35.10	4.48	37.68	4.49	39.03	3.89	**36.16**	4.25	**37.82**	4.43	**39.22**	3.94
$I_{135,G}$	37.71	4.41	42.42	4.70	43.17	4.41	**39.86**	4.54	**42.66**	4.66	**43.35**	4.42
$I_{135,B}$	35.75	4.89	40.32	5.42	40.42	4.79	**37.13**	5.05	**40.52**	5.37	**40.75**	4.95
S_0	37.32	4.35	41.38	4.57	42.34	4.31	**38.73**	4.26	**41.42**	4.52	**42.53**	4.37
S_1	43.34	4.37	47.21	4.61	48.73	3.84	**45.35**	4.72	**48.13**	4.70	**49.15**	3.78
S_2	41.75	4.69	44.97	4.90	45.70	4.18	**43.14**	4.91	**45.27**	4.97	**46.00**	4.19

Best mean values by channel and algorithms are highlighted in bold fonts. PSNR is computed on the image without the borders (border size of 4 pixels).

condition (0.25 for a three-dimensional image with a pixel spacing of 4). This value is higher than for EARI and LMMSE. It can be explained by the fact that the demosaicing quality of bilinear is lower than for EARI or LMMSE. The κ value for the bilinear algorithm is lower than that for EARI. Indeed, as the EARI algorithm better reconstructs high frequencies, the value of κ is high, which helps preserve details. The κ value of LMMSE is low because LMMSE produces zipping artifacts. To mitigate these artifacts, κ value should have a lower value.

Table 2 shows the average PSNRs and standard deviations for bilinear, EARI, and LMMSE algorithms, without and with anisotropic diffusion refinement. We can see that the best values of the PSNR average are given by the LMMSE with anisotropic diffusion refinement. For LMMSE, on average, the anisotropic diffusion algorithm improves the PSNR value of 0.2 dB. We can also see that the anisotropic diffusion refinement improves the quality of demosaicing bilinear and EARI algorithm in terms of PSNR. The quality improvement in terms of PSNR is significant, probably due to the low PSNR values given by the bilinear method.

For the runtime to process one image of size 768×1024 pixels without anisotropic diffusion refinement, we report 0.19 s for bilinear, 10.1 s for EARI, and 0.89 s for LMMSE. With the refinement, we have 11.83 s for bilinear, 21.72 s for EARI, and 12.46 s for LMMSE. These runtimes are obtained using a laptop with 16 GB RAM, a processor Intel core I5-8265U, and without any GPU usage.

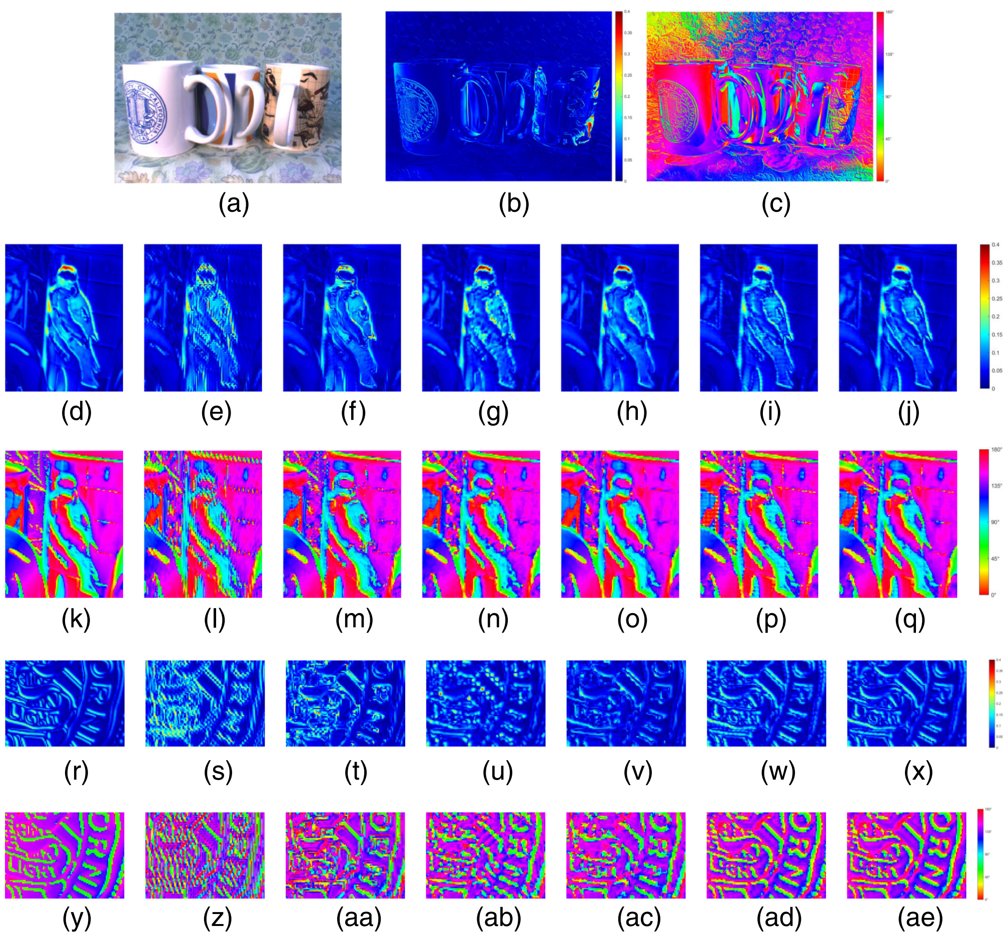

Fig. 7 Visualization results for the scene called “cup2” from the Morimatsu dataset.[6] Two zoomed areas, representing the DOLP and the AOLP, are visualized in false colors. An-di stands for anisotropic diffusion. (a) Reference S_0. (b) Reference DOLP G. (c) Reference AOLP G. (d) Zoomed reference. (e) Bilinear. (f) Bilinear with An-di. (g) EARI. (h) EARI with An-di. (i) LMMSE. (j) LMMSE with An-di. (k) Zoomed reference. (l) Bilinear. (m) Bilinear with An-di. (n) EARI. (o) EARI with An-di. (p) LMMSE. (q) LMMSE with An-di. (r) Zoomed reference. (s) Bilinear. (t) Bilinear with An-di. (u) EARI. (v) EARI with An-di. (w) LMMSE. (x) LMMSE with An-di. (y) Zoomed reference. (z) Bilinear. (aa) Bilinear with An-di. (ab) EARI. (ac) EARI with An-di. (ad) LMMSE. (ae) LMMSE with An-di.

5.3 Qualitative Results

Figure 7 shows the result of LMMSE, bilinear, and EARI demosaicing without and with refinement for image "cup2." For LMMSE, on the zoom of DOLP and AOLP images, we can see that the anisotropic diffusion refinement reduces the impact of zipping artifacts [see the bird's wing in Figs. 7(i) and 7(j) for DOLP and Figs. 7(p) and 7(q) for AOLP]. For the first zoomed image, the refinement improves the PSNR's value by 0.3 dB. For the full image, the PSNR's value without refinement is 41.61 dB, whereas it is 41.65 dB with refinement. The given values of PSNR correspond to the mean of PSNR's values over the 12 bands.

We can also see that the visual improvement for bilinear is important for both DOLP Figs. 7(e) and 7(f) and AOLP Figs. 7(l) and 7(m). In terms of PSNR, the improvement is also important, for the first zoomed image, where the PSNR increased by 2.56 dB with the refinement. For the full image, the PSNR without refinement is 34.44 dB and with refinement is 36.18 dB. This result confirms the visual observation.

For the EARI, the visual improvement is also visible on the bird's wing in Figs. 7(g) and 7(h) for DOLP and Figs. 7(n) and 7(o) for AOLP. In terms of PSNR, for the first zoomed image, the refinement increases the PSNR's value by 0.19 dB. For the full image, the PSNR's value without the refinement is 38.10 dB and with is 38.43 dB.

On the second zoomed image, for the three algorithms, we see that the refinement step succeeds with keeping details [see Figs. 7(s)–7(x) for DOLP and Figs. 7(z)–7(ac) for AOLP]. We can also observe that LMMSE offers a better reconstruction of the text.

Figure 8 shows the result of LMMSE, bilinear, and EARI demosaicing without and with refinement for image "doll2." As for Fig. 7, we see that applying anisotropic diffusion to the output of bilinear CPFA demosaicing greatly improves the visual results. For EARI, we can see a better reconstruction of the central circle in DOLP images in Figs. 8(g) and 8(h) and reduction of some artifacts for AOLP in Figs. 8(n) and 8(o). For LMMSE, we can see that the reconstruction of the circle is more accurate in DOLP [Figs. 8(i) and 8(j)] and reduction of some artifacts for AOLP.

6 Conclusion

In conclusion, we adapted a machine learning demosaicing algorithm (LMMSE) to the specific case of color polarization filter array images. We study the impact of the number of training

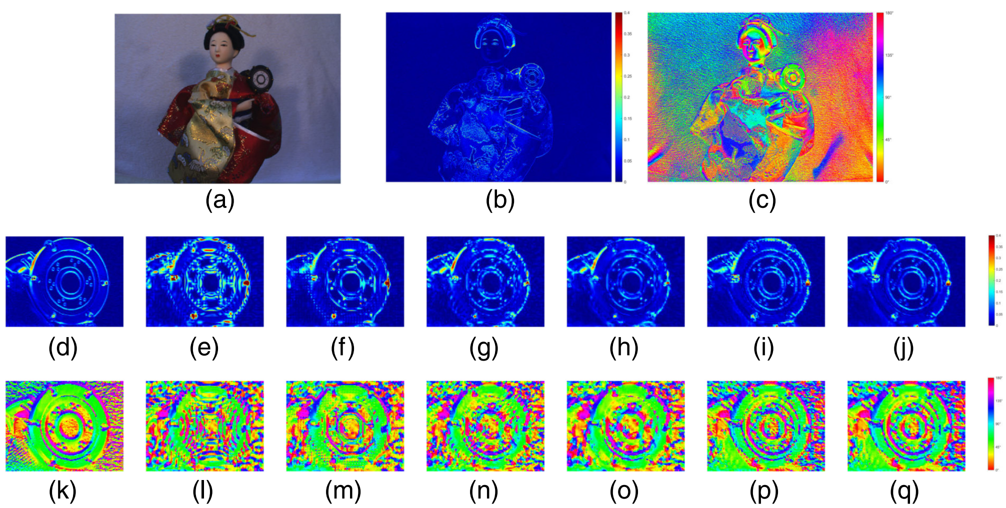

Fig. 8 Visualization results for the scene called "doll" from the Morimatsu dataset.[6] Two zoomed areas, representing the DOLP and the AOLP, are visualized in false colors. An-di stands for anisotropic diffusion. (a) Reference S_0. (b) Reference DOLP R. (c) Reference AOLP R. (d) Zoomed reference. (e) Bilinear. (f) Bilinear with An-di. (g) EARI. (h) EARI with An-di. (i) LMMSE. (j) LMMSE with An-di. (k) Zoomed reference. (l) Bilinear. (m) Bilinear with An-di. (n) EARI. (o) EARI with An-di. (p) LMMSE. (q) LMMSE with An-di.

images and optimize an anisotropic diffusion refinement algorithm to improve the visual quality of LMMSE demosaicing by reducing the zipping artifact and preserving the details. In terms of PSNR, the refinement increases the mean values by 0.2 dB. Depending on the image, the improvement of the complete proposed algorithm (LMMSE + refinement) is between 0.01 and 0.5 dB on a database of 28 images. The refinement improves PSNR and visual quality even for filtering-based algorithms (EARI algorithm), which suggests that this algorithm can also be a good direct extension for improving the best state-of-the-art methods for demosaicing.

As future work, we could investigate other possibilities for the computation of the diffusion tensor than the gradients from the 12 bands. We could also study the possibility to parallelize LMMSE on dedicated hardware compatible with video streams.

Disclosures

The authors declare that there are no financial interests, commercial affiliations, or other potential conflicts of interest that could have influenced the objectivity of this research or the writing of this paper.

Code and Data Availability

Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Acknowledgments

This research was partly funded by the Research Council of Norway over the project ICVIO (Grant No. 287209). This work was also supported by the ANR JCJC SPIASI Project (Grant No. ANR-18-CE10-0005) of the French Agence Nationale de la Recherche.

References

1. J.-B. Thomas et al., "Spectral characterization of a prototype SFA camera for joint visible and NIR acquisition," *Sensors* **16**(7), 993 (2016).
2. P.-J. Lapray et al., "Multispectral filter arrays: recent advances and practical implementation," *Sensors* **14**(11), 21626–21659 (2014).
3. Sony, "Polarization image sensor," tech. rep., Polarsens (2018).
4. T. Okawa et al., "A 1/2inch 48m all PDAF CMOS image sensor using 0.8 μm quad Bayer coding 2×2OCL with 1.0 LUX minimum AF illuminance level," in *Int. Electron Devices Meet.*, IEEE, pp. 16.3.1–16.3.4 (2019).
5. C. S. Chun, D. L. Fleming, and E. Torok, "Polarization-sensitive thermal imaging," *Proc. SPIE* **2234**, 275–286 (1994).
6. M. Morimatsu et al., "Monochrome and color polarization demosaicking using edge-aware residual interpolation," in *IEEE Int. Conf. Image Process. (ICIP)*, IEEE, pp. 2571–2575 (2020).
7. N. Hagen et al., "Fourier-domain filtering analysis for color-polarization camera demosaicking," *Appl. Opt.* **63**, 2314 (2024).
8. S. Qiu et al., "Linear polarization demosaicking for monochrome and colour polarization focal plane arrays," *Comput. Graph. Forum* **40**(6), 77–89 (2021).
9. A. Spote et al., "Joint demosaicing of colour and polarisation from filter arrays," in *Color and Imaging Conf.*, Society for Imaging Science and Technology, Vol. 2021, pp. 288–293 (2021).
10. R. Dumoulin et al., "Impact of training data on LMMSE demosaicing for colour-polarization filter array," in *16th Int. Conf. Signal-Image Technol. & Internet-Based Syst. (SITIS)*, IEEE, pp. 275–280 (2022).
11. S. Wen et al., "Convolutional demosaicing network for joint chromatic and polarimetric imagery," *Opt. Lett.* **44**(22), 5646–5649 (2019).
12. V. Nguyen et al., "Two-step color-polarization demosaicking network," in *IEEE Int. Conf. Image Process. (ICIP)*, IEEE, pp. 1011–1015 (2022).
13. Y. Sun, J. Zhang, and R. Liang, "Color polarization demosaicking by a convolutional neural network," *Opt. Lett.* **46**, 4338 (2021).
14. Y. Luo et al., "Learning a non-locally regularized convolutional sparse representation for joint chromatic and polarimetric demosaicking," *IEEE Trans. Image Process.* **33**, 5029–5044 (2024).
15. Y. Zheng et al., "Color polarization imaging demosaicing based on Stokes vector information complementation and fusion," *Opt. Express* **32**, 44049 (2024).

16. M. Morimatsu et al., "Monochrome and color polarization demosaicking based on intensity-guided residual interpolation," *IEEE Sens. J.* **21**(23), 26985–26996 (2021).
17. G. G. Stokes, "On the composition and resolution of streams of polarized light from different sources," *Trans. Cambridge Philos. Soc.* **9**, 399 (1851).
18. D. Kiku et al., "Residual interpolation for color image demosaicking," in *IEEE Int. Conf. Image Process.*, IEEE, pp. 2304–2308 (2013).
19. G. Courtier et al., "Correlations in joint spectral and polarization imaging," *Sensors* **21**(1), 6 (2021).
20. N. Galatsanos and R. Chin, "Digital restoration of multichannel images," *IEEE Trans. Acoust. Speech Signal Process.* **37**(3), 415–421 (1989).
21. H. Trussel, M. Sezan, and D. Tran, "Sensitivity of color LMMSE restoration of images to the spectral estimate," *IEEE Trans. Signal Process.* **39**(1), 248–252 (1991).
22. M. A. Wober and F. R. Soini, "Method and apparatus for recovering image data through the use of a color test pattern," US Patent 5,475,769 (1995).
23. D. Taubman, "Generalized Wiener reconstruction of images from colour sensor data using a scale invariant prior," in *Proc. Int. Conf. Image Process. (Cat. No.00CH37101)*, Vol. 3, pp. 801–804 (2000).
24. H. J. Trussell and R. E. Hartwig, "Mathematics for demosaicking," *IEEE Trans. Image Process.* **11**(4), 485–492 (2002).
25. L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.* **14**(12), 2167–2178 (2005).
26. P. Amba, J. Dias, and D. Alleysson, "Random color filter arrays are better than regular ones," *J. Imaging Sci. Technol.* **24**, 294–299 (2016).
27. P. Amba, J. B. Thomas, and D. Alleysson, "N-LMMSE demosaicing for spectral filter arrays," *J. Imaging Sci. Technol.* **61**(4), 40407 (2017).
28. D. Alleysson and P. Amba, "Method for reconstructing a colour image acquired by a sensor covered with a mosaic of colour filters," US Patent 11,057,593 (2021).
29. P.-J. Lapray et al., "An FPGA-based pipeline for micropolarizer array imaging," *Int. J. Circuit Theor. Appl.* **46**(9), 1675–1689 (2018).
30. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990).
31. J.-B. Thomas and I. Farup, "Demosaicing of periodic and random color filter arrays by linear anisotropic diffusion," *J. Imaging Sci. Technol.* **62**(5), 050401 (2018).
32. S. Di Zenzo, "A note on the gradient of a multi-image," *Comput. Vision Graph. Image Process.* **33**(1), 116–125 (1986).
33. L. Bigué, "Polarization imaging datasets," GitHub (13 July 2023). https://github.com/laurent-bigue/Polarization-imaging-datasets (accessed 28 July 2023).
34. L. Bigué, A. Foulonneau, and P.-J. Lapray, "Production of high-resolution reference polarization images from real world scenes," *Proc. SPIE* **12690**, 126900B (2023).
35. S. Wen, Y. Zheng, and F. Lu, "A sparse representation based joint demosaicing method for single-chip polarized color sensor," *IEEE Trans. Image Process.* **30**, 4171–4182 (2021).
36. S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Stat. Surv.* **4**, 40–79 (2010).
37. J. C. Lagarias et al., "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optim.* **9**(1), 112–147 (1998).

Ronan Dumoulin is a PhD student at the Université de Haute-Alsace. He received his master's degree in signal and image processing from the Université de Clermont Auvergne, France, in 2023. His current research interests include polarimetric imaging and spectro-polarimetric imaging.

Alexandra Spote: Biography is not available.

Pierre-Jean Lapray received his engineer's degree in 2010 and his PhD in 2013 from Université de Bourgogne, France. Since 2016, he has been an associate professor at the Université de Haute-Alsace. He was on the development of spectral and polarimetric imaging systems and the associated pre-processing.

Jean-Baptiste Thomas received his bachelor's degree in applied physics and his master's degree in optics from Université Jean Monnet, France, and his PhD from Université de Bourgogne. Since 2010, he has been an associate professor at Université Bourgogne Europe. In 2015, he was a guest researcher at EPFL. From 2016 and 2021, he was at the Norwegian University of Science and Technology (NTNU) on sabbatical. He has focused extensively on the development

of spectral filter array technology for which he has contributed to the definition of the imaging pipeline to enable spectral imaging for computer vision. Since 2016, he has been focusing on understanding material appearance and its measure using imaging systems.

Ivar Farup received his MSc degree in technical physics from the Norwegian Institute of Technology (NTH) in 1994 and his PhD in applied mathematics from the University of Oslo (UiO), Norway, in 2000. He is currently a professor of computer science at NTNU, mainly focusing on color science and image processing.