2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) | 978-1-6654-6495-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/SITIS57111.2022.00031

Impact of training data on LMMSE demosaicing for Colour-Polarization Filter Array

Ronan Dumoulin
Université de Haute-Alsace
IRIMAS UR 7499
Mulhouse, France

Pierre-Jean Lapray
Université de Haute-Alsace
IRIMAS UR 7499
Mulhouse, France
0000-0003-2230-0955

Jean-Baptiste Thomas
Université de Bourgogne, dpt IEM
IMVIA EA 7535
Dijon, France
jean-baptiste.thomas@u-bourgogne.fr

Ivar Farup
Department of Computer Science
Norwegian University of Science and Technology (NTNU)
Gjøvik, Norway
0000-0003-3473-1138

***Abstract*—Linear minimum mean square error can be used to demosaic images from a colour-polarization filter array sensor. However, the role of training data on its performance is yet an open question. We study the model selection using cross-validation techniques. The results show that the training model converges quickly, and that there is no significant difference in training the model with more than 12 images of approximately 1.5 megapixels. We also found that the selected trained model performs better compared to a dedicated Colour-Polarization Filter Array demosaicing algorithm in terms of Peak Signal-to-Noise Ratio.**

***Index Terms*—Color-polarization imaging, Polarization filter array, demosaicing, spatial interpolation, linear minimum mean squared error.**

I. Introduction

Color-polarization image sensors capture images with a spatial sampling based on a mosaic of Color Polarization Filter Array (CPFA). It senses a specific filtered signal by pixel, relatively to one spectral band and one polarization direction. The most common CPFA camera is a 12-channel sensor, which combines three color channels and four polarization angles of analysis, equally-distributed between $0°$ and $180°$. The SONY IMX250 MYR [10] is one realization which is commercially available, and its spatial arrangement is shown in Figure 1.

Demosaicing of filter arrays images is necessary to get a full resolution image that is easier to handle by computer vision algorithms. Like for the typical case of color filter arrays that uses knowledge on spatio-spectral correlation [5], CPFA demosaicing benefits from specific correlations present in the data to perform image reconstruction [3], [6]. The Linear Minimum Mean Square Error (LMMSE) was successfully used to demosaic colour and polarisation images, with results competitive to the state-of-the-art [11]. However, the role of training data on its performance is yet an open question. Thanks to the recent availability of a large image database, this kind of analysis is now practically feasible.

This work was supported by the ANR JCJC SPIASI project, grant ANR-18-CE10-0005 of the French Agence Nationale de la Recherche.

In this article, we study the performance of the LMMSE demosaicing algorithm relatively to the amount of images used for training. To this end, we use cross-validation techniques applied to a dataset of spectral and polarimetric images. The LMMSE algorithm is introduced in the next Section. Then, we define an experiment for model selection over the largest existing database. Results demonstrate that the learning model reaches convergence with a limited number of training images, and that the corresponding trained algorithm performs statistically better compared to a dedicated CPFA algorithm, i.e. the Edge-Aware Residual Interpolation algorithm (EARI) [7], in terms of PSNR.

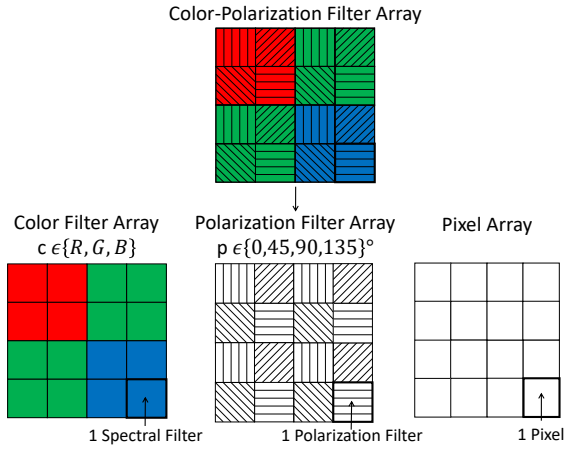


Fig. 1: A CPFA sensor architecture. A superpixel (4×4 pixels), composed by filter and pixel arrays. The spatial arrangement of filters used in this work is that of the SONY IMX250 MYR sensor.

II. LMMSE demosaicing

LMMSE is a supervised learning algorithm, which means that a linear model must be trained to demosaic images. The

978-1-6654-6495-6/22/$31.00 ©2022 IEEE
DOI 10.1109/SITIS57111.2022.00031

training is performed on a set of images that contains both mosaiced and full resolution images, called references. The mosaiced images are generated with the Equation 1, where $\boldsymbol{X}$ is the mosaiced image, $\boldsymbol{Y}$ is the reference image, and $\boldsymbol{M}$ is the mosaicing matrix that simulates the effect of a CPFA filter :

$$\boldsymbol{X} = \boldsymbol{M}\boldsymbol{Y} \tag{1}$$

The size of $\boldsymbol{Y}$ is PHW with number of channels $P = 12$, the height H, and the width W. The size of $\boldsymbol{X}$ is HW.

For the proper functioning of the algorithm, the image data are rearranged in one dimensional vector. The reference rearranged image is noted $\boldsymbol{y}$, and the mosaiced rearranged image $\boldsymbol{x}$.

$$\hat{\boldsymbol{y}} = \boldsymbol{D}\boldsymbol{x} \tag{2}$$

Equation 2 gives the relationship between the mosaiced image $\boldsymbol{x}$ and the estimate of the reference image $\hat{\boldsymbol{y}}$. Thus, demosaicing can be considered as an inverse problem, which admits a solution based on the criterion of minimizing the Mean Square Error (MSE) between $\boldsymbol{y}$ and its estimate $\hat{\boldsymbol{y}}$ derived from $\boldsymbol{x}$. The demosaicing matrix $\boldsymbol{D}$ is computed as follows:

$$\boldsymbol{D} = E_i\{\boldsymbol{y}\boldsymbol{x}^t(\boldsymbol{x}\boldsymbol{x}^t)^{-1}\} \ . \tag{3}$$

E is the expectation, and $i \in [1, n]$ indexes the image in a database of n images.

To stabilize the solution of $\boldsymbol{D}$, we use a neighbourhood of 10×10 pixels. This ad-hoc choice offered the best trade off between performance and computational complexity [1], [11]. In the following, matrices that contain neighbouring pixels will be denoted by the index 1. The matrix $\boldsymbol{S_1}$ is a constant matrix with zeroes and ones, for removing neighbours from $\boldsymbol{y_1}$. Taking into account the neighbours, we can rewrite the above equations as follows:

$$\begin{cases} \boldsymbol{y} & = \boldsymbol{S_1}\boldsymbol{y_1} \\ \boldsymbol{x}_1 & = \boldsymbol{M_1}\boldsymbol{y_1} \\ \boldsymbol{D} & = \boldsymbol{S_1}\boldsymbol{R}\boldsymbol{M_1}^t(\boldsymbol{M_1}\boldsymbol{R}\boldsymbol{M_1}^t)^{-1} \\ \hat{\boldsymbol{y}} & = \boldsymbol{D}\boldsymbol{x_1} \end{cases} \tag{4}$$

with

$$\boldsymbol{R} = \frac{1}{\frac{HW}{hw}k} E_i\{\boldsymbol{y_1}\boldsymbol{y_1^t}\}, \tag{5}$$

where hw is the size of superpixels. $\boldsymbol{R}$ is the mean of autocorrelation of $\boldsymbol{y_1}$ over the $\frac{HW}{hw}$ superpixels of each learning image and over the k learning images of the database.

III. Experiments

To study the performance of the LMMSE training, we used two methods:

1) Method 1: the K-Fold cross-validation technique, in which we vary K from 2 to 24, with a among a total of $n = 24$ images.
2) Method 2: we vary the number of training images from 1 to 12 and compare the demosaicing results.

To assess the demosaicing quality, we use the Peak Signal to Noise Ratio (PSNR).

Then, we compared the quality of the LMMSE demosaicing with a demosaicing algorithm dedicated to CPFA, the Edge-Aware Residual Interpolation algorithm (EARI).

A. Database

In our experiments, we use the database of images from Wen et al. [12], which is available online. It is composed of 105 colour and polarimetric images of 1456×1088 pixels, each of them having 12 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 $180°$ ($p \in \{0°, 45°, 90°, 135°\}$). Each channel is defined by $I_{c,p}$. The images have been captured with a three-CMOS prism-based RGB camera, and with a linear polarizer rotated in front of the camera. This is currently the largest available database of colour and polarimetric images. This criterion is important because we must be sure to have enough images to train and test the algorithm. In addition, to avoid bias in the evaluation procedure using learning methods, the images used for training should not be reused for testing.

By examining individually each band visually, we noticed that the blue channels at $0°$, $45°$, $90°$, and $135°$ polarization orientations are blurred. To measure it, we used the Helmli and Scherer method [8] to estimate the degree of focus for each of the 12 channels, and for the 105 images in the database. This is an indicator for the relative measurement of blur among channels. For comparison, we did the same process with another dataset, which is the Monno et al. database [7] (40 images).

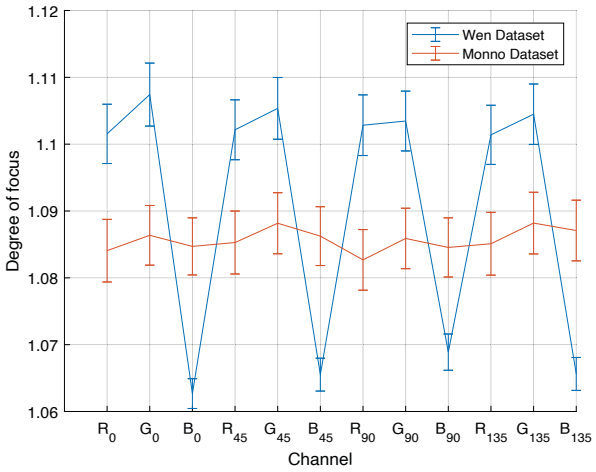


Fig. 2: Degree of focus (mean and standard deviation) according to the channel. A factor of 0.1 is applied to standard deviations for better readability.

Figure 2 shows the degree of focus computed on the two databases for each of the 12 channels. It is noticeable that the

measurements for the four blue channels of the Wen database are lower relatively to the other channels. The differences for the Monno database are much smaller. This blur study will serve as an element of understanding for the result analysis in the rest of the article. This does not call into question the use of this database, since 1-the algorithms must be robust to this kind of defect present in imperfect sensors, and 2-the demosaicing results are compared by channel and relatively to the same set of reference data. Moreover, Monno's database only contains 40 images of 768×1024 pixels, so Wen's database allows us to have much more data to conduct this study. Moreover, other existing database, like Qiu et al. [9] (40 images of 512×512 pixels) or Lapray et al. [4] (10 images of 368×496 pixels) have a smaller amount of data than the Wen database.

B. Cross-validation using K-Fold

The first evaluation method (called Method 1) used in this article is based on the K-Fold method [2]. 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 (LOOCV). The Algorithm 1 shows the different steps of the K-Fold experiment.

Algorithm 1 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
```

The limitation of this method is that we can not evaluate for a few training images, where the number of testing images can be too small to be representative. In our case, it would be interesting to study the convergence of the learning procedure for a few images. This is why we used another method to complete this study.

C. Training with i images

The second method (called Method 2) consists of selecting i images to train the algorithm and randomly taking 50 images for testing the algorithm. The images for training must be different from those for testing to avoid bias. For the first iteration, the training is done with 1 image and then the 50 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 50 images. The Algorithm 2 shows the different steps of Method 2.

Algorithm 2 Second method

```
INPUT : Database
OUTPUT : PSNR (μ,σ)

Select 12 images for learning
Select 50 random images for test
for i = 1 : 12 do
    Learn D matrix with i images
    Demosaic the 50 test images
    Compute PSNR for the test group
    Compute the PSNR (μ,σ)
end for
```

IV. Results and Discussion

A. Model selection

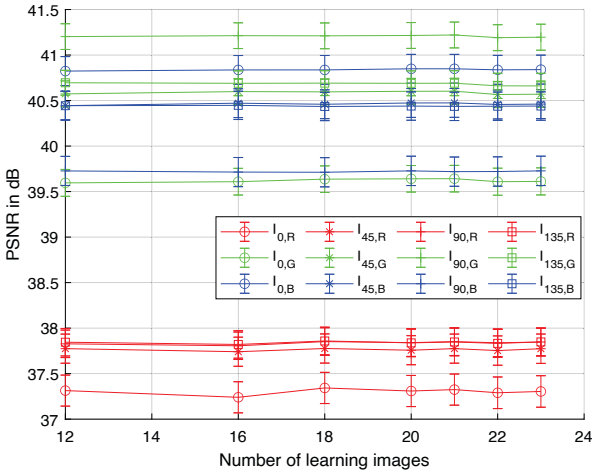


Fig. 3: PSNR as a function of learning images number with Method 1. A factor of 0.05 is applied to standard deviation for better readability.

The Figure 3 shows the average PSNRs, and standard deviations obtained with the Method 1 (K-Fold cross-validation described in Subsection III-B). 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. We can also see that the PSNR results for the blue and green channels are close. This is due to the fact that the blue channels are blurred as explained in the Section III-A. The PSNRs for the green channels are supposed to be higher than those of the blue and red channels, because the green channels are spatially oversampled compared to the others. This oversampling allows more information to be available to reconstruct the data lost during mosaicing. It is clear that the channel dependency of blur effect is actually impacting the PSNR results, especially for algorithms that assume spatial correlation in the image, such as LMMSE or EARI. The blur effect acts as a low pass filter, and thus increases the spatial correlation in the image. Blurred channels are relatively better reconstructed than sharper channels. This helps explain the good PSNR results for the four blue channels.

The Figure 4 shows the average PSNR values, and standard deviations obtained with Method 2 (described in Subsection III-C). It can be seen that the PSNR values increase monotonically throughout the iteration process from 1 to 12 learning images. The PSNR values stabilize with a relatively low number of images. This means that the algorithm converges quickly and therefore does not need a large amount of images to perform well. It should be noted that the global PSNR averages are different compared to Method 1. As the scenes used for the tests are different for the two methods, the results varies depending on the image statistics. Moreover, we remark that if the images are of lower resolution, it will take more images for the algorithm to converge, which validates the dependency on the amount of data. As for Method 1, the PSNR values of the blue channels are relatively high (close to the oversampled green channels). This is due to the blur present in the blue channels.

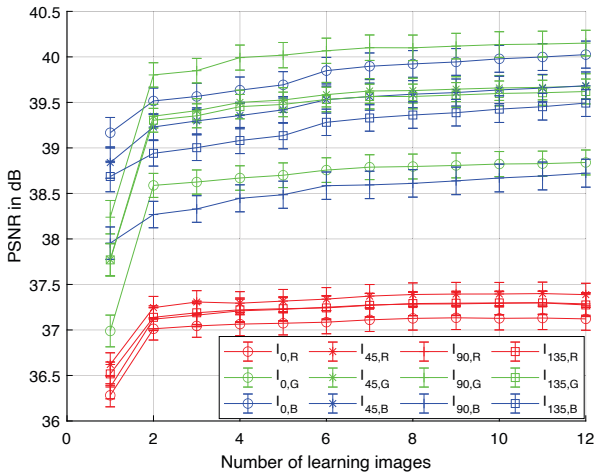


Fig. 4: PSNR as a function of learning images with Method 2. A factor of 0.05 is applied to standard deviation for better readability.

To summarize, we can conclude with Method 1 that with 12 images, the model has already converged, and there does not seem to be a statistically significant difference in using more images. With Method 2, we showed that the trend is rapidly converging, and that only need a handful of images to train the model.

For the comparison between the LMMSE and the EARI, we chose the trained model with 12 images for the implementation.

B. LMMSE versus EARI

We compare the quality of the LMMSE demosaicing on the dataset with the EARI algorithm. For LMMSE, we have kept the trained model based on $i = 12$ images from the Method 2. We also do the same test with a model with $i = 55$ images for comparison.

For the evaluation, we randomly select 50 images (none of which were used for training to avoid bias). Then, these 50 images are demosaiced by both EARI and LMMSE, and the averages and standard deviations are computed by channel.

The Table I shows the average PSNRs, and standard deviations for the EARI and the LMMSE. It can be seen that the LMMSE gives better PSNR values than the EARI, which was confirmed by a Wilcoxon signed rank test with a p-value of 8.3742e-90, computed over the PSNR of all bands for all images reconstructed by LMMSE (with 12 images) in front of EARI, demonstrating the statistical significance of the result. The PSNR results are very close between training with 12 images and 55 images. This means that from a certain number of images (in our case 12) there is no point in learning with more images.

It is important to note that EARI is an algorithm that does not need to be learned, it is operational as is and will always give the same result for a given image. An image demosaiced with EARI will therefore always have the same PSNR. Unlike the EARI, the LMMSE needs a learning step to demosaic an image. This means that the demosaicing results can vary depending on the number of images used for training and the images themselves. For the same image, the LMMSE can give different PSNR values.

It should be noted that we use a method which optimizes the result for MSE, then we study the result relatively to the PSNR, which is also based on MSE. There is a known bias here. One way to escape bias is to use another metric for the evaluation. At the moment, we don't know of any other metric that can quantify image quality, including the polarimetric aspect.

The Figure 5 shows the S_0 reference image and their demosaiced versions with EARI and LMMSE, with a zoom on an area of interest (a writing). S_0 is a reconstructed color image which contains the total intensity for each spectral channel c, computed from the four polarization channels as follows:

$$S_{0,c} = \frac{I_{0,c} + I_{45,c} + I_{90,c} + I_{135,c}}{2} \ . \tag{6}$$

It can be seen that the demosaicing performed by the LMMSE allows a better reconstruction at the level of the writings.

TABLE I: Results for the experiment conducted in Section IV-B. Average μ and standard deviation σ of PSNR for EARI and LMMSE algorithms. Best mean values by channel are highlighted in bold fonts.

	EARI		**LMMSE** $i = 12$		**LMMSE** $i = 55$	
	μ	σ	μ	σ	μ	σ
$I_{0,R}$	35.96	2.42	**37.12**	2.50	37.02	2.53
$I_{0,G}$	38.02	2.95	**38.84**	2.74	38.76	2.71
$I_{0,B}$	38.87	3.02	**40.02**	2.97	40.00	2.94
$I_{45,R}$	36.02	2.43	**37.39**	2.49	37.37	2.51
$I_{45,G}$	38.61	2.96	**39.68**	2.74	39.64	2.71
$I_{45,B}$	38.51	3.00	**39.68**	3.02	39.65	2.95
$I_{90,R}$	36.03	2.48	37.29	2.50	**37.30**	2.51
$I_{90,G}$	39.10	3.07	**40.15**	2.82	40.13	2.79
$I_{90,B}$	37.87	3.07	**38.72**	3.07	38.70	3.02
$I_{135,R}$	36.10	2.49	37.27	2.56	**37.28**	2.56
$I_{135,G}$	38.70	2.97	**39.62**	2.74	39.56	2.71
$I_{135,B}$	38.43	2.95	**39.49**	2.94	39.47	2.91

The Figure 6 shows the Degree Of Linear Polarisation ($DOLP$) images, computed by:

$$DOLP_c = \frac{\sqrt{S_{1,c}^2 + S_{2,c}^2}}{S_{0,c}} \ , \tag{7}$$

where S_1 is the intensity difference between the $0°$ and $90°$ polarization images, and S_2 the intensity difference between the $45°$ and $135°$ polarization images. Only the green channels of $DOLP$ are shown, as all color channels are very similar. The zoomed images highlight that the EARI creates more artefacts than the LMMSE. Moreover, it can also be seen that the LMMSE tends to smooth the image.

The Figure 7 shows the Angle Of Linear Polarisation ($AOLP$), computed by:

$$AOLP_c = 0.5 \arctan\left(\frac{S_{2,c}}{S_{1,c}}\right) \ . \tag{8}$$

As with the $DOLP$, it can be seen that the EARI creates more artefacts than the LMMSE and that the LMMSE tends to smooth the image.

V. Conclusion

In conclusion, we observed that the LMMSE learning process converges quickly. With Wen's database, a handful of images are enough to get better results than the state of the art. We also observed that the LMMSE creates less artifacts than EARI and allows for better demosaicing, specifically on written content.

As future work, we could evaluate the results for random batches of superpixels from the database rather than full frames. In this way, we can identify if potential biases could be introduced by image content. Moreover, it would be interesting to compare LMMSE with deep learning-based demosaicing methods, in terms of performance for a given amount of data.

Acknowledgment

This work was supported by the ANR JCJC SPIASI project, grant ANR-18-CE10-0005 of the French Agence Nationale de la Recherche.

References

[1] David Alleysson and Prakhar Amba. Method for reconstructing a colour image acquired by a sensor covered with a mosaic of colour filters, July 6 2021. US Patent 11,057,593.

[2] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[3] Guillaume Courtier, Pierre-Jean Lapray, Jean-Baptiste Thomas, and Ivar Farup. Correlations in joint spectral and polarization imaging. *Sensors*, 21(1), 2021.

[4] Pierre-Jean Lapray, Luc Gendre, Alban Foulonneau, and Laurent Bigué. Database of polarimetric and multispectral images in the visible and nir regions. In *Unconventional Optical Imaging*, volume 10677, pages 666–679. SPIE, 2018.

[5] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: a systematic survey. In William A. Pearlman, John W. Woods, and Ligang Lu, editors, *Visual Communications and Image Processing 2008*, volume 6822, pages 489 – 503. International Society for Optics and Photonics, SPIE, 2008.

[6] Sofiane Mihoubi, Pierre-Jean Lapray, and Laurent Bigué. Survey of demosaicking methods for polarization filter array images. *Sensors*, 18(11), 2018.

[7] Miki Morimatsu, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Monochrome and color polarization demosaicking using edge-aware residual interpolation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2571–2575. IEEE, 2020.

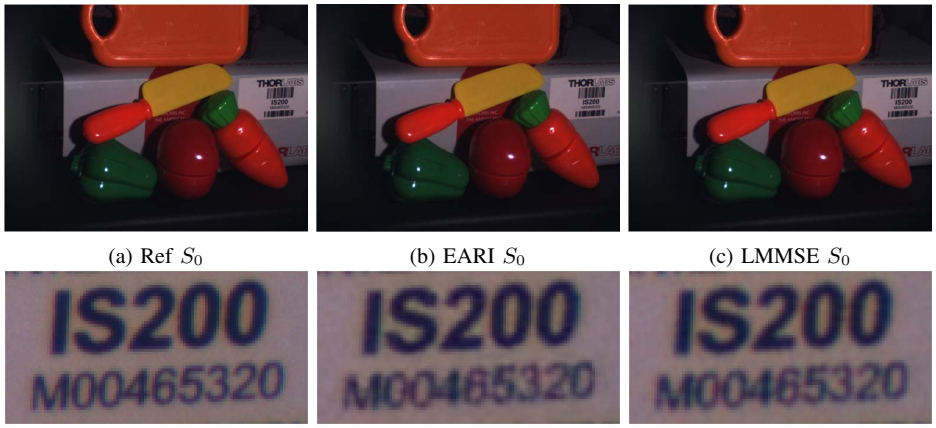
[8] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013.

[9] Simeng Qiu, Qiang Fu, Congli Wang, and Wolfgang Heidrich. Linear polarization demosaicking for monochrome and colour polarization focal plane arrays. *Computer Graphics Forum*, 40(6):77–89, 2021.

[10] Sony. Polarization image sensor. Technical report, Polarsens, 2018.

[11] Alexandra Spote, Pierre-Jean Lapray, Jean-Baptiste Thomas, and Ivar Farup. Joint demosaicing of colour and polarisation from filter arrays. In *Color and Imaging Conference*, volume 2021, pages 288–293. Society for Imaging Science and Technology, 2021.

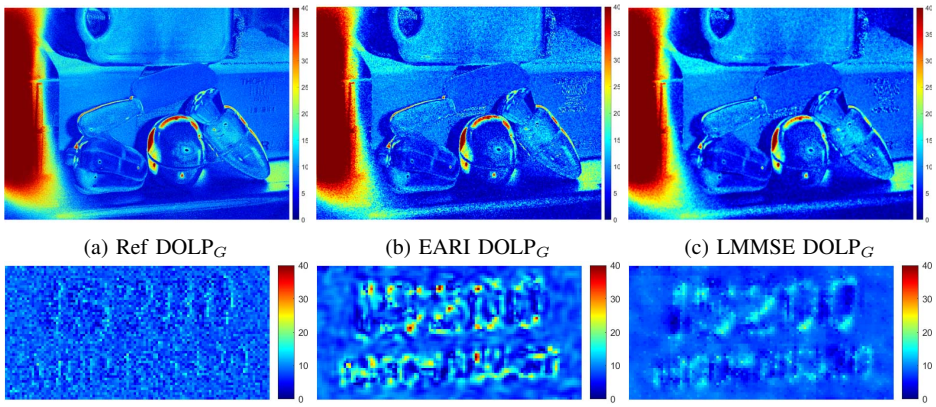
[12] Sijia Wen, Yinqiang Zheng, Feng Lu, and Qinping Zhao. Joint chromatic and polarimetric demosaicing via sparse coding. *arXiv preprint arXiv:1912.07308*, 8, 2019.



(a) Ref S_0 (b) EARI S_0 (c) LMMSE S_0

(d) Ref S_0 zoomed (e) EARI S_0 zoomed (f) LMMSE S_0 zoomed

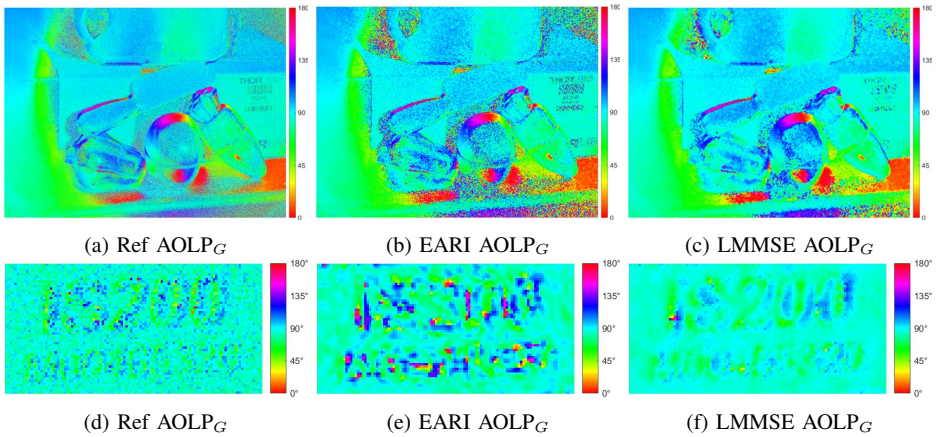
Fig. 5: (a)-(c) Visualization of the S_0 images for reference image and demosaiced images with EARI and LMMSE. (d)-(f) Zoomed versions.



(a) Ref DOLP$_G$ (b) EARI DOLP$_G$ (c) LMMSE DOLP$_G$

(d) Ref DOLP$_G$ (e) EARI DOLP$_G$ (f) LMMSE DOLP$_G$

Fig. 6: (a)-(c) Visualization of the $DOLP$ images (green channel) for reference image and demosaiced images with EARI and LMMSE. (d)-(f) Zoomed versions.



(a) Ref AOLP$_G$ (b) EARI AOLP$_G$ (c) LMMSE AOLP$_G$

(d) Ref AOLP$_G$ (e) EARI AOLP$_G$ (f) LMMSE AOLP$_G$

Fig. 7: (a)-(c) Visualization of the $AOLP$ images (green channel) for reference image and demosaiced images with EARI and LMMSE. (d)-(f) Zoomed versions.