Comparison of Mosaic Patterns for Spectral Filter Arrays

Davit Gigilashvili, Jon Yngve Hardeberg, and Jean-Baptiste Thomas
The Norwegian Colour and Visual Computing Laboratory, Department of Computer Science
NTNU – The Norwegian University of Science and Technology, Gjøvik, Norway
davit.gigilashvili@ntnu.no

***Abstract*—Spectral Filter Arrays allow snapshot multispectral acquisition within a compact camera. While Bayer filter mosaic is a widely accepted standard for color filter arrays, no single mosaic pattern is considered dominant for spectral filter arrays. We compare different patterns for 8-band mosaics, and their overall performance in terms of spectral reconstruction, as well as color and structure reproduction. We demonstrate that some mosaics having overrepresentation of certain filters perform better than the ones with overrepresentaiton of other filters, while the arrays having all filters equally represented perform better than the arrays with overrepresentation.**

***Index Terms*—SFA, MSFA, spectral imaging, filter array, demosaicing**

I. Introduction and Background

Using Spectral Filter Arrays (SFA) [1] is a promising technique for multispectral imaging that enables acquisition of the multispectral information in a single shot. In its core sense, it is an extension of the widely used and accepted Color Filter Arrays (CFA) [2] that are currently extensively used in modern digital color cameras. In contrast to arrangement of three, "Red", "Green" and "Blue" filters in CFAs, SFA consists of several different filters that after demosaicing process, enables obtention of the multispectral information. While demosaicing is a relatively well-studied and understood process, the impact SFA pattern has on the overall performance still remains an open question. Wang [3] proposes several methods for demosaicking and refers to design considerations as well, but question about optimal arrangement still remains open. In another work, Amba *et al.* [4] propose that random arrangement performs better than the regular one for CFAs. It is point of interest whether the conclusion can be extended to the multispectral arrays or not. In this work, we investigate, whether any particular pattern or group of patterns can clearly outperform the others, which factors influence the performance of patterns, and whether overrepresentation of any particular filter is an advantage for the mosaic. The paper is arranged as follows: in the next section, we introduce the workflow and experimental setup. The preliminary study using 1000 patterns to check the need of normalization is introduced in Section III, and the results are summarized in Section IV, followed by the discussion and summary.

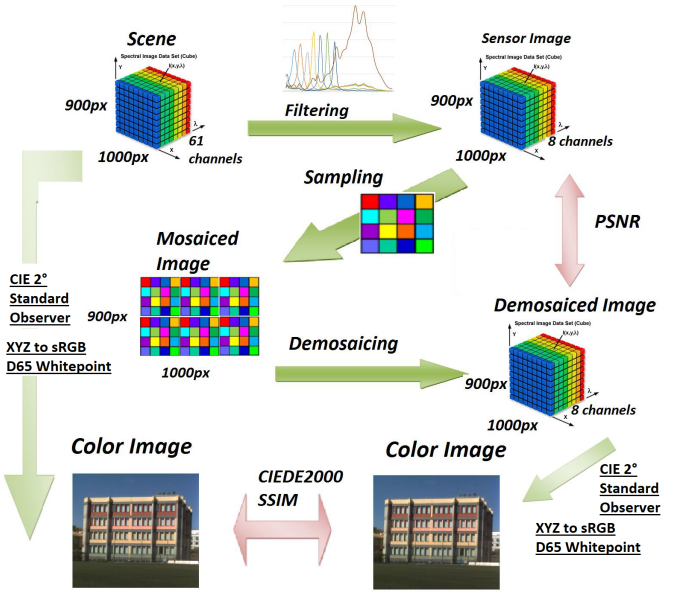

Fig. 1. Overall Framework. Sensor image is generated using scene and filter data. It is sampled using SFA mosaic pattern and mosaiced image is generated. LMMSE demosaicing is used to reconstruct the original sensor image. Demosaiced image and original sensor image are compared to evaluate the performance. Color images generated using original scene and demosaiced image data are also used for performance assessment.

II. Framework and Experimental Setup

A. General Framework and Workflow

The general framework of the work carried out is illustrated on Fig.1. In this work, we simulate acquisition with the real measured data combined according to an imaging model, where the sensor image generated by scene data multiplied with the filter data sampled over 10nm steps. The general imaging model we used is given in Equation 1, where f is image R,G,B values; e(λ) is illuminant (equal-energy radiator - Illuminant E is assumed throughout the experiment), r(λ) is surface reflectance of the scene object, and c(λ) are sensor sensitivity functions.

$$f = \int_{\omega} e(\lambda) r(\lambda) c(\lambda) d\lambda \tag{1}$$

2018 Colour and Visual Computing Symposium (CVCS)
978-1-5386-5645-7/18/$31.0 © 2018 IEEE

The hyperspectral cube of the scene image, with spatial resolution 1000x900 pixels and 61 spectral bands within 400-1000nm range is used for the experiment. It is multiplied by and integrated over ”filters” data that include spectral transmission data of 8 bandpass filters and the sensitivity of the sensor. After this step, so called ”sensor image” is obtained that has the same spatial resolution but only 8 channels, each of which is a result of the integration using each of the 8 filters present in the system. After that a pattern, i.e. arrangement of the filters within a SFA is generated and the sensor image is masked using this pattern. Therefore, we obtain a mosaiced image, where in a particular pixel, information of just one channel is present. Thus, we have a two dimensional image. On the next stage of the process, demosaicing algorithm is applied to the mosaiced image and 8-channel image is reconstructed. In order to evaluate the reconstruction, i.e. demosaicing accuracy, two cubes, an original, pre-sampling cube I and demosaiced one $\hat{\mathrm{I}}$ are compared against each other using Peak-Signal-to-Noise-Ratio (PSNR) metric. Besides, a color image is generated using the very initial cube and using the demosaiced one and performance in sense of color reproduction and structure preservation in intensity images are evaluated with CIEDE2000 and SSIM metrics respectively. By using the initial 61-band spectral cube for color image generation, we enable assessment of the full framework and not only demosaicing. In order to generate XYZ color images, we used CIE 2 degree standard observer functions on 380-780nm spectral range (visible spectrum) sampled with 10nm spacing. Further conversion was made from XYZ to sRGB colorspace assuming D65 whitepoint. It is worth mentioning that a sensor image is generated just once, while other steps are repeated within several iterations and a different pattern of the SFA is applied for sampling during each iteration of the loop.

B. Scenes

Two hyperspectral images with dimensions 1000×900 and 61 spectral channels with 10 nanometers of spectral resolution within 400-1000 nm range were used. One image was used as a reference image for testing purposes while another was used for training of the demosaicing model (discussed below). The images were obtained from the database of the Colorlab of the University of Granada [6]. As performance of the demosaicing algorithm is highly content dependent, we tried to use the scenes with more or less similar visual look and color constitution (refer to Fig. 2). Besides, the images are interesting as they include natural scenes - vegetation, as well as building and artificial objects.

C. Filters

We used the data from a paper by Thomas *et al.* [7] that is already a product of bandpass filter transmission and sensor sensitivity data. There are 8 different filters that cover the spectral range from 380 to 1100 nanometers, but as the scene data is available only within 400-1000nm range, filter data is utilized only for this range as well. It is worth mentioning that

Fig. 2. Reference image on the left and training image on the right.

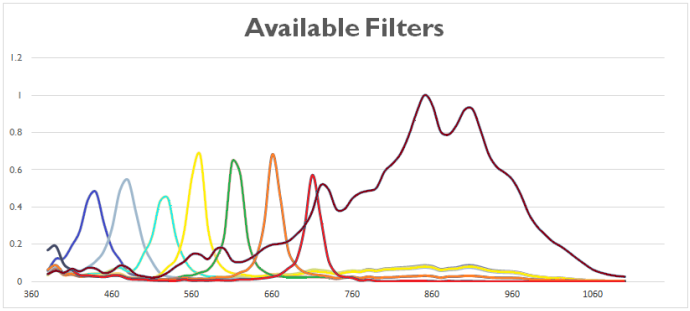

Fig. 3. Spectral transmittance curves of the 8 filters used. It is worth mentioning that for illustration purposes, the image shows the bands interpolated, while actually, the images are downsampled to the sampling of the filters.

except for seven visible range filters, one filter has significant transmission in near infrared (NIR) part of the spectrum as well. The spectral curves of the filters are illustrated in Fig. 3.

D. Sampling and Pattern Generation

We used 4×4 moxel for sampling. In other words, the repetitive pattern, a.k.a. moxel (mosaic element) in the multispectral filter array consists of 16 pixels, arranged as 4×4 square shapes. Theoretically, arrangement of 8 filters within 4×4 matrix, gives possibility to generate 8^{16} different combinations that is more than 281 trillion different patterns. But it is quite obvious that not all patterns need to be examined and that the number can be substantially decreased. For instance, there would be no sense of examination at all, if all the pixels of the array consist of just 1 type of the relatively narrow-band filters, as the whole information about the rest of the spectrum will be discarded. A binary tree-based method for array generation was suggested by Miao and Qi in their paper [8]. In order to make the patterns as meaningful, as possible, they impose some constraints to the generation process. Namely, the authors consider spectral consistency and uniformity of the filter distribution, in order to make sure that each pixel has all types of other filters available at a reasonable distance that makes demosaicing accurate. We decided to use and examine two different types of architectures that were also present in Wang's work [3].

1. An array where each type of filter has equal probability of presence, i.e. we have 2 pixels for each type of filter in 16-pixel array (Fig. 4 (a))

2. One type of filter has 50% probability of appearance, i.e. it takes a half of all pixels within a moxel, while another type

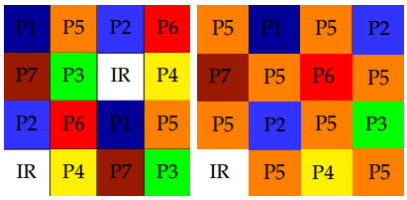

Fig. 4. (a) Moxel Architecture I. (b) Moxel Architecture II.

of filter covers just $\frac{1}{8}$ of the moxel, i.e. is represented by two pixels and the rest of the filters are present just in one pixel of the moxel. (Fig. 4 (b))

Applying Miao's method decreased the number of possible combinations dramatically, down to 80640 instances.

E. Demosaicing

The pivotal step of the work was demosacing, i.e. reconstructing a sensor image from a sampled mosaic image. The algorithm used for demosaicing was Linear Minimum Mean Square Error estimation [9], used in Wang's work [3]. Its successful performance for SFAs has been also demonstrated in [4] [5]. As Wang explores in his work, LMMSE estimation takes demosaicing as image restoration problem. If we refer to Equation 2, X is a mosaiced image, while Y is an initial spectral image and P is a matrix projecting Y on X.

$$X = PY \tag{2}$$

Demosaicing is actually, inverting the above described process. Let's refer to Equation 3, where $\hat{Y}$ is a demosaiced image, X is a mosaiced image and Q is a reconstruction matrix. As P_r is not invertible, Q cannot be obtained directly from P_r. Therefore, some a priori information is used to estimate Q.

$$\hat{Y} = QX \tag{3}$$

We use training data to estimate Q. We pass a sensor image and its mosaiced version to the model, as well as just a mosaic version of the reference image, and the model tries to estimate Q, so that the mean square error between Y and $\hat{Y}$ is minimized. Q matrix that generates $\hat{Y}$ with minimal mean squared error is considered optimal for demosaicing. Using LMMSE has two substantial advantages:

1. Its computational cost is quite low.

2. It enables performance comparison among even dramatically different patterns that is not the case for many other demosaicing algorithms. For instance, biinear interpolation is favorable to some specific arrangements, while LMMSE may help to identify other good proposals too, as the success of the demosaicing is in itself dependant on the neighborhood, i.e. the architecture examined.

F. Evaluation Metrics

Three evaluation metrics were used: Peak-Signal-to-Noise-Ratio - PSNR that takes a sensor image as a perfect reference and compares the degradation between a reference sensor image and reconstructed demosaiced image. CIEDE2000[1] [10] [11], for colorimetric performance and Structural Similarity SSIM [12] for examining the preservation of image structures were also used.

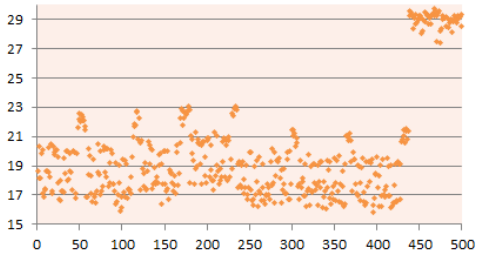

Fig. 5. Plot of PSNR as a function of the pattern number for Architecture II

III. Preliminary Experiments and Normalization

Considering computational cost of the framework, the preliminary experiments were conducted to outline the performance trends. 500 Architecture I and 500 Archiceture II mosaic patterns have been analyzed by the preliminary study. After preliminary examination, Architecture I did not demonstrate any significant trends. Although equal presence of all filters makes it too complicated to identify "similar" patterns and arrange them in a logical order in 1-40320 pattern number range, we could still observe the individual instances. However, all values belonged to the same very narrow range and none of them stood out significantly. On the other hand, Architecture II showed clear indications of difference among the patterns (Fig. 5). The patterns with the same type of filter covering the half of the array were given numbers from the same range, while all patterns were numbered from 1 to 500. This enabled us plot metric values as a function of the pattern number. Fig. 5. illustrates that some group of the patterns clearly stands out of the rest with its PSNR values. This is the group of the arrays, where Infrared pixels cover the half. Predictably, that difference faded away for CIEDE2000 and SSIM metrics, as IR information is not used when generating color images, as the standard observer functions are beyond 780nm. If we refer to spectral sensitivities of the filters on Fig. 3, we will see that the 8^{th} IR filter covers widest range of the spectrum, as well as the largest area under the curve.

Looking at these results, it was not possible to distinguish whether this result was due to magnitude difference created by the differences in sensitivities between the spectral bands (i.e. NIR is very sensitive, while others exhibiting less sensitivity), or the spectral nature of the bands - in particular transversal sensitivities of the NIR band in the visible range, which may act as an attenuation neutral filter, which increases spectral correlation, e.g. improves demosaicing. In order to disambiguate between those cases and identify the reason of the difference in PSNR values, we propose to normalize the images based on

[1]Considering the computational cost of the whole framework, color difference is found between the mean colors of the scenes, instead of pixel-wise calculation followed by pooling.

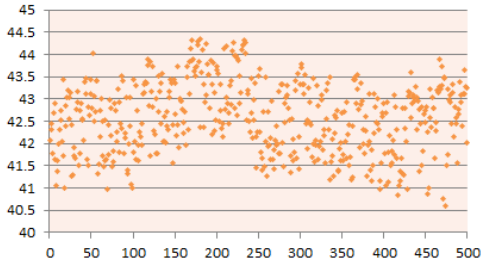

Fig. 6. Plot of PSNR as a function of the pattern number for Architecture II after normalization.

the sensitivities of the filters (in other words, divide the sensed values by the integrals of the filters). This has been shown to improve the demosaicing task by Mihoubi Et al. [13] It is worth mentioning that we did not consider illumination during normalization, as we are using Illuminant E that is already neutral. The Fig.6. illustrates that very obvious difference in PSNR values has disappeared after normalization by filter sensitivities. This makes us conclude that the results will be biased unless normalized and thus, used normalized data for the rest of the study.

IV. Results

80640 SFA patterns generated with Miao's method [8] have been analyzed (40320 per architecture).

A. Architecture I

The results for Architecture I are illustrated on Fig. 7. The best performing ones for Architecture I are shown on Fig. 11. First and foremost, we have to mention that the range of all those data is very small. Therefore, the best performing and the worst performing samples differ with a small margin. The range for PSNR is 2.126, for CIEDE2000 - 0.079, and for SSIM - 0.00004. Secondly, it is difficult to see any trends from any of the three plots. While there are some outliers across the whole horizontal axis, the vast majority of the data evenly fits within the very small range. If we have a look at Fig. 9, we will see that there is hardly any visual difference between best- and worst-performing ones in terms of colorimetric difference. On the other hand, color degrades significantly for all patterns, as we see in comparison with the original (right) on Fig. 9. One of the explanations of this could be the leakage of the NIR filter into the visible range. However, structural similarity is almost perfectly kept in all images. All SSIM values are close to 1 and differ insignificantly, only by the 5^{th} decimal.

B. Architecture II

The results for Architecture II are illustrated on Fig. 8. The data range is larger than that of the Architecture I. This is logical and intuitive, as all Architecture I arrays differ in just spatial allocation of the filters - having the equal number of pixels for each filter. On the other hand, Architecture II patterns differ in filter composition, and their performance varies more. The range for PSNR is 5.211, for CIEDE2000 - 0.304, and for SSIM - 0.00012. We see clear trends for all three metrics. The pattern number should be interpreted in a following way: patterns from 1 to 5040 have 8 pixels of Filter1, patterns from 5041 to 10080 have 8 pixels of Filter2, from 10081 to 15120 have 8 pixels of Filter3, and so on with intervals of 5040. Within each of those intervals, first 720 patterns have 2 pixels of Filter1 (or Filter2, if Filter1 is already present in 8 pixels, e.g. within 1-5040 interval), next 720 patterns have two pixels for Filter2 (or Filter3, if Filter2 is already present in 8 pixels), and so on. For example, patterns from 1 to 720 have 8 pixels for Filter1 and 2 pixels for Filter2; the patterns from 5761 to 6480, have 8 pixels for Filter2 and 2 pixels for Filter3. On the Fig. 8.(a), we see that there is an increasing trend in the first half of the range, while the value suddenly drops when we switch from 8-pixel-for-Filter4 to 8-pixel-for-Filter5 patterns. Besides, there are peaks on the boundaries of the intervals that can be an indication that within each 5040 pattern range, the ones with 2 pixels of Filter8, perform better than the others within the same range. For instance, if we have a closer look on the PSNR values within 20100 and 20200 range (refer Fig. 10), we can see that the drop point coincides exactly the pattern group boundary, 20160 (5×5040). Besides, it is worth mentioning that the last group of the patterns, with 8-pixel-for-Filter8 has many points with relatively low values than the others. There is an indication that abundance of the filters peaking around 560nm improve the performance of the Architecture II patterns.

The three best performing patterns for Architecture II in terms of PSNR are illustrated on Fig. 12 and the three worst performing ones on Fig. 13. It is worth mentioning that 96 out of 100 best performing patters have abundance of 4^{th} filter, peaking around 560nm. On the other hand, the majority of poorly performing filters have abundance of the NIR filter. The filters with the best and the worst colorimetric performance are given on Figures 14 and 15. It is interesting that both best and worst performing ones are the ones with abundance of the IR filter. Therefore, we can assume that IR filter is the most unstable one. This can be explained with the fact that it has varying transmittance throughout the whole spectrum and highly depends on the presence of other filters. Even though we see some trend in SSIM values, the range of the values is extremely small and the best- and the worst-performing ones differ only on 4^{th} decimal level.

C. Comparison: Architecture I vs. Architecture II

Another interesting point is to compare Architecture I and Architecture II. If we refer to the boxplot in Fig. 16(A), we will see that Architecture I mostly outperforms Architecture II by the small margin of the PSNR. Although Architecture II has some outliers that perform either better in terms of CIEDE2000, or worse either in terms of CIEDE2000, or SSIM, very small difference, as well as the overall range of the data, make it difficult to conclude which one performs better.

V. Discussion

Several indications can be seen in the results above. First of all, we have seen that having all filters equally represented is

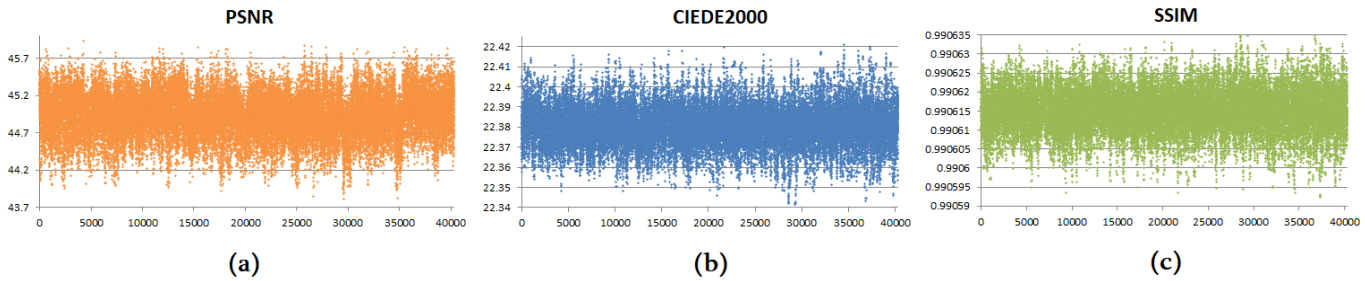

Fig. 7. Results for Architecture I. (a) Plot for PSNR values. (b) Plot for CIEDE2000 values. (c) Plot for SSIM values. Horizontal axis corresponds to the SFA pattern number. Vertical axis corresponds to the metric value.

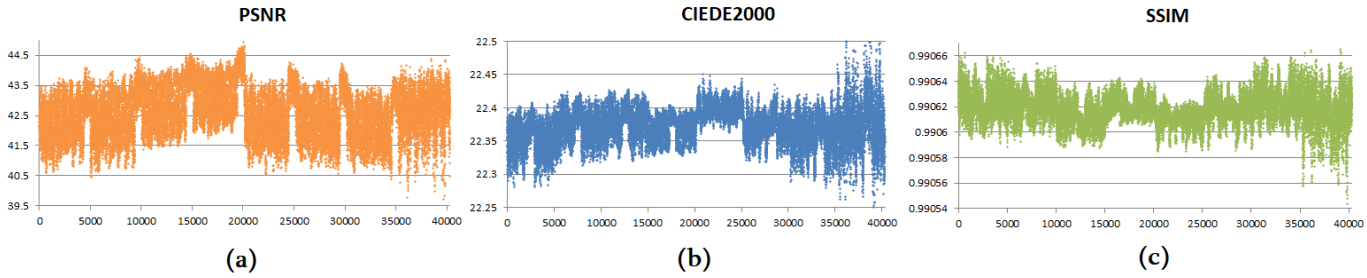

Fig. 8. Results for Architecture II. (a) Plot for PSNR values. (b) Plot for CIEDE2000 values. (c) Plot for SSIM values.

Fig. 9. There is no noticeable difference between the best (left) and the worst (middle) performing ones by CIEDE2000, while both significantly differ from the original (right).

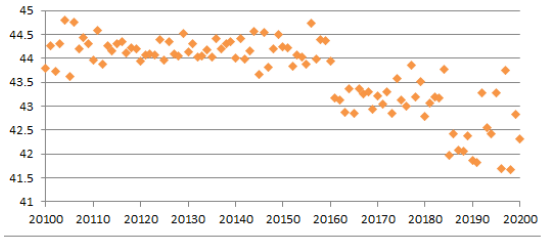

Fig. 10. PSNR values for Architecture II within 20100 - 20200 pattern number range.

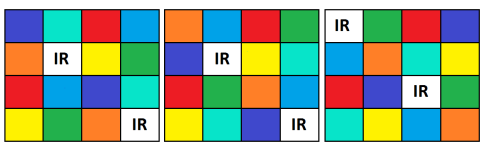

Fig. 11. Three best performing patterns for Architecture I by PSNR (left), CIEDE2000 (middle), and SSIM (right).

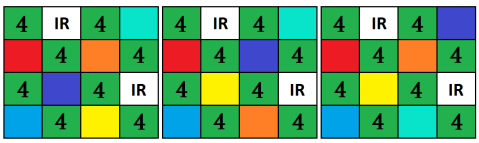

Fig. 12. Three best performing patterns for Architecture II in terms of PSNR.

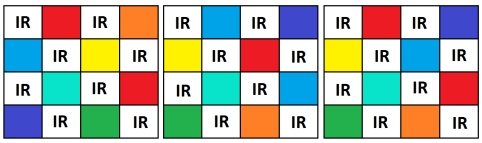

Fig. 13. Three worst performing patterns for Architecture II in terms of PSNR.

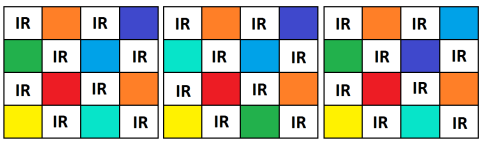

Fig. 14. Three best performing patterns for Architecture II in terms of CIEDE2000.

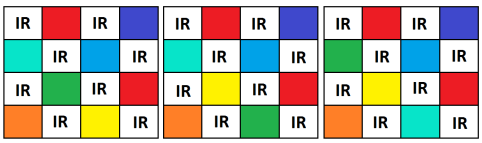

Fig. 15. Three worst performing patterns for Architecture II in terms of CIEDE2000.

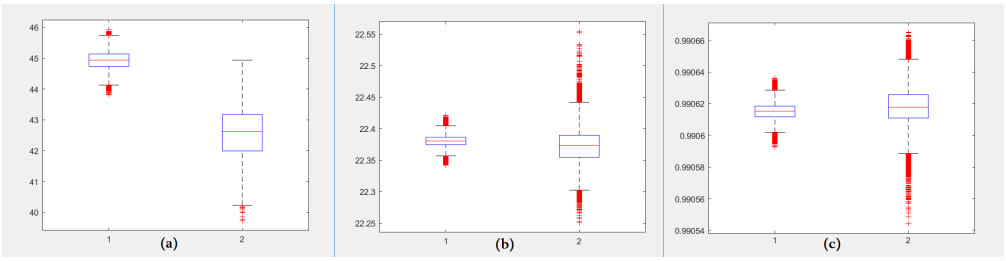

Fig. 16. 1 - Architecture I. 2 - Architecture II. Box plots: (a) PSNR. (b) CIEDE2000. (c) SSIM. The central mark is the median, bottom and top edges of the box are 25^{th} and 75^{th} percentiles, respectively. Whiskers extend to the extreme data points excluding outliers, while outliers are plotted as the red '+' symbol.

better than abundance of any filters in terms of PSNR. On the other hand, having eight filters peaking around 560nm seems to be the best solution, while having half of the moxel covered with NIR filter can be less reasonable. This can be explained with the composition of our test and training images. Both scenes have significant area covered by vegetation. Therefore, we need to consider that these results can be content-dependent and may not be generalized for other images. Another factor that makes generalization difficult is extremely small differences of metric values between worst- and best performing ones, especially, for Architecture I. This makes us think that as long as the moxel is covered with equal number of filters, the spatial allocation does not have significant impact. On the other hand, having the moxel architecture with different number of pixels allocated per filter, the performance can be highly content dependent. Based on above-presented data, we conclude that all patterns preserve structural similarity almost perfectly. In total, colorimetric performance of the framework is poor, spectral reconstruction is reasonably good, while visual structures are kept intact.

Besides, this work has several shortcomings that is not addressed in this paper: First of all, subjective psychometric evaluation, as well as further objective metrics can be used for performance evaluation. Secondly, due to limited computational resources, we used one image pair for reference and training. As LMMSE is a content dependent algorithm, larger dataset is needed for generalization of the findings. Besides, the experiment assumed perfect conditions, e.g. no noise, and 100% fill factor of the sensor pixel. This is hardly ever the case in reality. Thus, to make the framework more robust in practice, the conditions closer to real-life need to be considered. Finally, the number of filters is a compromise that could provide a good spectral reconstruction, but may not be optimal for color images. The pollution from the NIR range onto the visible bands exists, and that will clearly impact colorimetric accuracy. Thus, CIEDE2000 could be unreliable indicator of the overall performance.

VI. Summary and Conclusion

We compared different 8-channel Spectral Filter Array mosaic patterns. Having very small range of evaluation metric values, as well as just single pair of images for testing and training makes it difficult to draw the firm conclusions. However, there are indications that the architecture with equal number of pixels for each filter outperforms the architecture having half of the array covered with single type of filters and is more stable, less impacted by the spatial allocation of the filters within the moxel. On the other hand, performance of the patterns differs, when some of the filters are overrepresented within the moxel. Those hypotheses need further examination with larger datasets and other evaluation metrics.

References

[1] P.J. Lapray, X. Wang, J.B. Thomas, and P. Gouton, 2014. "Multispectral filter arrays: Recent advances and practical implementation." Sensors, 14(11), pp.21626-21659.

[2] R. Lukac, and K.N. Plataniotis, 2005. "Color filter arrays: Design and performance analysis." IEEE Transactions on Consumer Electronics, 51(4), pp.1260-1267.

[3] X. Wang, "Filter array based spectral imaging: Demosaicking and design considerations", PhD Thesis, NTNU, Gjøvik, Norway, 2016

[4] P. Amba, J. Dias, D. Alleysson, "Random Color Filter Arrays are better than Regular Ones", Journal of Imaging and Science Technology, 60(5), 2016

[5] P. Amba, JB Thomas and D. Alleysson (2017), "N-LMMSE Demosaicing for Spectral Filter Arrays", Journal of Imaging Science and Technology. Vol. 61(4), pp. 40407-1-40407-11.

[6] Database of ColorLab of the University of Granada: http://colorimaginglab.ugr.es/pages/Data Accessed on: 27/03/2018

[7] J.B. Thomas, P.J. Lapray, P. Gouton, and C. Clerc, 2016. "Spectral characterization of a prototype SFA camera for joint visible and NIR acquisition." Sensors, 16(7), p.993.

[8] L. Miao, H. Qi, and W.E. Snyder, 2004, October. "A generic method for generating multispectral filter arrays." In Image Processing, 2004. ICIP'04. 2004 International Conference on (Vol. 5, pp. 3343-3346). IEEE.

[9] L. Zhang and X. Wu, 2005. "Color demosaicking via directional linear minimum mean square-error estimation." IEEE Transactions on Image Processing, 14(12), pp.2167-2178.

[10] M.R. Luo, G. Cui, and B. Rigg, 2001. "The development of the CIE 2000 colour difference formula: CIEDE2000." Color Research & Application, 26(5), pp.340-350.

[11] G. Sharma, W. Wu, and E.N. Dalal, 2005. "The CIEDE2000 color difference formula: Implementation notes, supplementary test data, and mathematical observations." Color Research & Application, 30(1), pp.21-30.

[12] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Qualifty Assessment: From Error Visibility to Structural Similarity." IEEE Transactions on Image Processing. Vol. 13, Issue 4, April 2004, pp. 600612.

[13] S. Mihoubi, J.B. Thomas, O. Losson, L. Macaire, 2017, November. "Illumination-robust multispectral demosaicing." In Image Processing Theory, Tools and Applications (IPTA), 2017 Seventh International Conference on (pp. 1-6). IEEE.