CAMERA-SPECIFIC IMAGE QUALITY ENHANCEMENT USING A CONVOLUTIONAL NEURAL NETWORK

Anselm Grundhöfer and Gerhard Röthlin

Disney Research

ABSTRACT

We propose a simple method to enhance the image quality of modern Bayer pattern based cameras that are offering an optional sub-pixel accurate sensor shift to capture full rgb images of static scenes. By capturing a series of image pairs of the same, unaltered scene, once captured with the Bayer pattern and once with the full rgb image data, a database of corresponding images can be generated in which the former contains artifacts resulting from the spatial interpolation which is absent in the latter. Using this data, we train a convolutional neural network (CNN) to generate a camera dependent image processing operation which reduces the image artifacts and enhances the image quality to approximate the quality of the full rgb image within a single exposure, even if moving scenes are captured. We present a simple, do-it-yourself method to capture and pre-process the data, train the network, and to enhance the images. An evaluation using several image quality assessment methods shows the effectiveness of the proposed method.

Index Terms— Image Enhancement, Moiré, Convolutional Neural Network

1. INTRODUCTION AND MOTIVATION

The Bayer pattern [1] and related spatial color filter arrays (CFA) [2, 3] have become a de-facto standard component in modern cameras. Especially at the consumer and prosumer levels, the trade-off between quality, cost, and size savings is mostly tolerated. However, these spatial color filters lack image quality compared to full rgb image capture systems. Since some modern cameras, however, offer the option to shift the sensor with the Bayer pattern to generate full-rgb color images requiring no spatial color interpolation, these limitations can be avoided. But due to the required acquisition of multiple, individually exposed images, this can only be applied when capturing completely static scenes. Since most real-world photographs are captured hand held and usually contain scene motion, it is desired to overcome this limitation.

We propose to use a camera and lens specific image enhancement method by training a CNN[4] to transform the single shot captured and demosaicked images into an approximation of the appearance of the full rgb image. This can be achieved by capturing a training data set of corresponding Bayer pattern and full rgb images of static scenes using cameras which are capable of doing so.

Although several method exist to solve the given problem of removing Bayer pattern based image artifacts, our approach differs within in the following points: We present:

- A simple and computationally efficient method to enhance image quality of cameras with a controllable moving sensor such as Pentax K-3 II, the Hasselblad H5D-200c MS or the Olympus E-M5 II
- A simple CNN architecture and an image registration tool to generate a data set and to apply a straightforward do-it-yourself training for a specific camera
- A quality comparison of the results with the internal pixel shift resolution enhanced multi-exposure image.

The remainder of the paper is organized as follows: We will give a short overview of the most relevant existing literature in Sec.2 and will describe the proposed image enhancement method in Sec.3. A prototypical implementation is summarized in Sec.4 followed by an image quality evaluation (Sec.5) and a final a conclusion section 6.

2. BACKGROUND AND RELATED WORK

Enhancing the image quality of photographs is an ongoing and quite active field of research [5]. With the rising trend of applying CNNs for such kind of tasks, even more effort has been made to enhance images by applying learning algorithms to, for example, generate better quality super-resolution images, as, for example, presented in [6, 7, 8].

In contrast to related approaches, we are focusing on training a specific camera and lens configuration which is able to also capture full rgb image data to optimally overcome its inherent limitations resulting from the CFA and demosaicking operation. Furthermore, our presented method can be seen as a do-it-yourself approach since it can be carried out in a straightforward manner using consumer hardware.

3. IMAGE ENHANCEMENT OF CONSUMER CAMERAS USING CNNS

As already mentioned in the introduction, we are focusing on correcting image artifacts resulting from the spatially color multiplexed capture process using a CFA and the according necessary demosaicking process. These artifacts can be mainly subdivided into a reduction of image sharpness, color fringes on high contrast edges as well as moiré artifacts [9]. The latter is most often suppressed using optical anti-aliasing within the optical path, but since this leads to an additional reduction in resolution, more and more cameras are available in which this filter is removed, which, however, leads to an increased likelihood to capture such moiré artifacts. Fig. 1 illustrates examples of the mentioned image degradations.

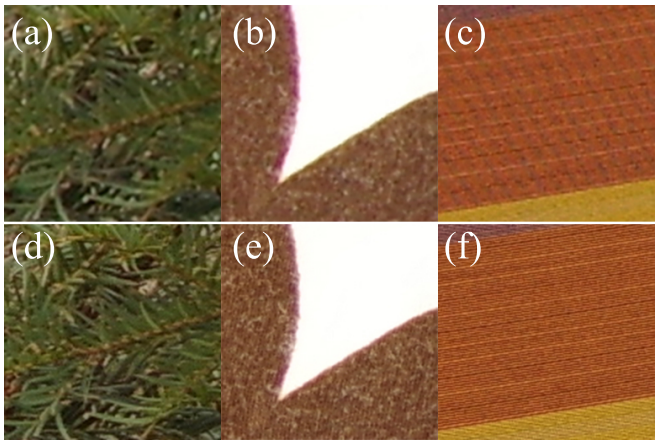

Fig. 1. Upper Row: Sample image artifacts of images generated using a Bayer pattern (a): blurriness, (b) color fringes, (c) moiré. Lower row (d)-(f): Corresponding full rgb images free of these artifacts. The shown images are crops of the actual training data set used in the evaluation described in Sec.5.

In the following section we are describing our proposed simple method on how these artifacts can be corrected for, using a specifically trained CNN.

3.1. Definition of the CNN

The CNN design is straight forward, consisting of a 3 channel (rgb) input layer, followed by N internal layers, and a single output layer transforming the tensor data back into a rgb representation. All layers are connected by convolutions with a $KxKxLxM$ weight tensor. L is the number of channels in the source layer, i.e. 3 rgb channels for the input layer, M is the number of channels in the destination layer and K is the size of the convolution Kernel. To not shrink the image in each layer, padding is added before the convolution operation by mirroring the input tensor. Activations are performed using the *elu* function [10] and a bias term. Most of the image enhancements we want to correct are mainly focused on very local image manipulations, however, also artifacts covering a larger spatial area resulting from moiré should be reduced with this approach. Therefore we decided to configure the CNN to work with a fixed generic kernel size K of 11 and used 5 hidden layers. The depth, i.e. the number of channels M in each hidden layer was set to 256.

The loss function consists of three error terms: Firstly, the output layer is compared with the expected output via a simple $\ell 1$ norm of the rgb differences:

$$L^{\ell_1}(P) = \frac{1}{N} \sum_{p \in P} |x(p) - y(p)| \tag{1}$$

where N is the number of pixels, $x(p)$ is the expected, i.w. the according full rgb color value and $y(p)$ the according pixel value of the network output. Secondly a custom single-scale SSIM [11] loss is applied:

$$L^{SSIM}(P) = \frac{1}{N} \sum_{p \in P} 1 - SSIM(p) \tag{2}$$

The combination of $\ell 1$ loss and SSIM loss are used since in [12] this was shown to produce superior results compared to using only a single loss term. Finally an additional regularization loss is added with an $\ell 2$ norm of all the weight tensors:

$$L^{reg}(P) = \sum_{w \in W} w^2 \tag{3}$$

Where W is the number of weights w. The influence of all three terms are adjusted using three individual weighting factors $\omega_{L^{\ell_1}}$, $\omega_{L^{SSIM}}$, and $\omega_{L^{reg}}$. The network training is performed using an adam optimization [13].

The following section summarizes how this network was trained in a prototypical realization and describes the required pre-processing steps to achieve that goal.

4. PROTOTYPICAL REALIZATION

The proposed idea was prototypically realized using a *Pentax K3-II* DSLR which contains no anti-aliasing filter and a movable sensor allowing to capture full per-pixel rgb information of still imagery. To apply our CNN based image enhancement we used the tensorflow framework [14]: In a first step, a set of images of the same scenes have to be acquired, perfectly registered, subdivided, and used for training the CNN. The individual steps of how this has been realized will be summarized in the following.

4.1. Training Data Set Generation

To generate an appropriate data set for training, care has to be takes that the data actually contains a sufficient variety of the image artifacts which should be removed. Therefore, images were taken of scenes containing mainly well-focused, high-frequency components.

4.1.1. Image Acquisition

To acquire the image data, the used Pentax K3-II camera was set to fully manual mode and all values were fixed for each image pair acquisition[1]. Both images of each scene were captured in rapid succession to minimize any potential light changes or movements. The aperture was set to $f/8$ to ensure a large depth of field but at the same time avoiding unnecessary blurring due to the diffraction limit [15].

4.1.2. Pair Registration

Although the images were all captured using a tripod and self-timer, small image shifts due to the unavoidable mirror movement and vibrations lead to a required software alignment step to ensure that each image pair is aligned as accurate as possible. Therefore, an automated feature detection and matching using AKAZE [16] followed by a homography based image warping step was applied to register each pair. To minimize quality degradations due to interpolation artifacts, the Bayer patterned image was always warped to the unaltered high-resolution image.

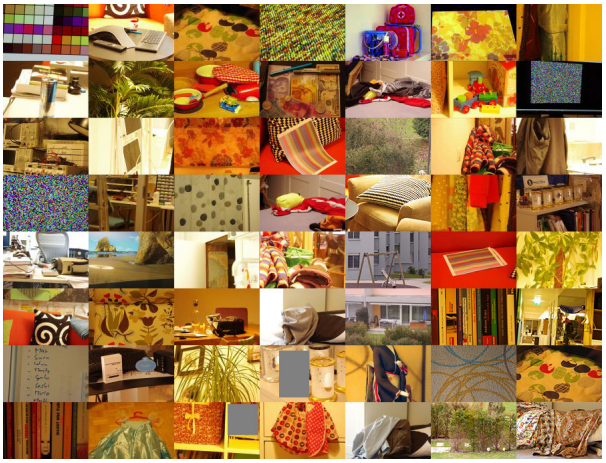

Fig. 2. Thumbnails of the samples used for training the network (each is a 24MP image). They were rotated and randomly cropped into approx. 170000 64^2 sized pieces.

4.1.3. Patch Generation

Having registered the input images, the patch pairs for training the CNN are generated. They were all cropped to square image regions of 64 pixels width. To focus on relevant patches containing at least some local variations, only image regions with rgb variances above a given threshold were chosen as training patches. Registered patches which contained color differences above a certain threshold were excluded from further processing since this measure might indicate some undesired scene motion within the two captures which would lead to a flawed training result.

[1]I.e. shutter, ISO, aperture, white balance, focusing and light metering

Figure 2 shows thumbnails of the used training data. From this, 170000 patches with a dimension of $64x64$ pixels were extracted with random image rotations. Since it is close to impossible to capture all potential colors of the full color cube, 32^3 uniformly colored patch pairs sampling the whole rgb color cube were added to the dataset to help guiding the network to accurately map colors which were not available in the training samples.

4.2. CNN Training

During the training phase of the CNN, the registered image patches were loaded into the tensorflow memory. All weights and biases of the CNN were randomly initialized with values of the expected variance and mean as proposed in [17].

The depth as well as the network layers were varied between different training runs. The best results were finally achieved using a depth of 256 with 5 layers and a $11x11$ kernel size. The loss weights were set to $\omega_{L^{\ell_1}} = 1.0$, $\omega_{L^{SSIM}} = 0.25$, and $\omega_{L^{reg}} = 0.0005$. Due to the limited available GPU memory, batches of 64 patch pairs were used in each training iteration. Each batch was chosen as a random subset of the remaining dataset. This selection was repeated until all patches were chosen once and then the process was started over again. The training was stopped after reaching an average loss of a $\ell 1$ rgb error of ~3.6 and an average SSIM of ~0.93.

4.3. Image Processing

Having finished the training, the CNN is ready to being used for enhancing the image quality of captured, demosaicked images. to avoid out-of-memory issues, this can be realized by loading an image into memory and sequentially processing overlapping sub-regions of the image which then are combined again afterwards to generate the final output image.

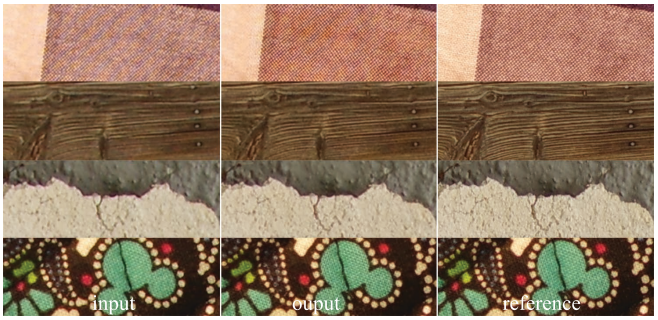

Fig. 3. Close-up of evaluation samples (cf. Fig. 4). The left column shows the input images, the center the generated outputs and the right the full-rgb references. In the upper row sample $VIII$ shows restored details and reduced moiré. In the 2nd, sample XI also shows restored details. In row three, the cyan color fringe of sample V is widely reduced and on the bottom, more details are reconstructed in sample XV.

Fig. 4. Set of 18 evaluation samples, ranging from rockwork to puppets, plants, wood and various synthetic and natural fabrics.

Table 1. Image quality enhancement evaluation of 18 images ($I - XVIII$) using four different error metrics. Our proposed method enhances the image quality in almost all four metrics in the whole test data set.

	SSIM		ΔE CIE2000		PSNR		HSV-PSNR	
	Test input	CNN output	Test input	CNN output	Test input	CNN output	Test input	CNN output
X-mas	0.944007	**0.950489**	1.11712	**0.832606**	78.7071	**80.4185**	29.3112	**32.6035**
jacket	0.907736	**0.918532**	1.14043	**1.0036**	80.9524	**81.3878**	32.6664	**33.3055**
bark	0.872082	**0.888763**	2.17739	**1.82226**	76.0238	**77.2159**	26.5725	**28.5022**
flower	0.941716	**0.948977**	1.43438	**1.39873**	83.6123	**84.8097**	34.2643	**36.7468**
wall	0.84925	**0.86149**	1.0483	**0.867731**	80.4174	**81.2209**	32.4495	**34.7386**
clothing	0.912729	**0.913741**	1.99531	**1.81123**	78.5794	**80.0373**	28.4387	**30.8832**
bricks	0.835692	**0.872536**	2.32035	**1.9269**	74.3123	**75.3655**	29.6526	**32.3468**
shirts	0.91714	**0.932181**	1.02573	**0.655952**	76.0085	**76.2307**	30.1278	**30.8785**
puppet	0.942483	**0.950971**	1.83497	**1.73843**	84.8439	**85.1673**	36.4138	**36.7897**
rocks	0.754907	**0.782223**	1.37835	**1.29047**	77.8367	**77.8893**	**30.061**	29.776
saloon	0.796942	**0.824613**	2.20546	**2.07741**	76.6698	**76.8616**	**29.0629**	29.002
lampshade	**0.90175**	0.899741	**2.06854**	2.15735	79.8131	**80.768**	31.3292	**33.1141**
roof	0.903924	**0.913559**	0.920708	**0.604267**	79.6357	**81.2469**	30.2431	**33.4454**
chalkboard	0.740472	**0.785859**	2.44952	**2.2924**	75.0645	**75.6871**	29.6817	**31.4046**
oven	0.895396	**0.900563**	1.8391	**1.75718**	78.1296	**78.859**	29.1039	**30.3405**
sign	0.800228	**0.815991**	1.43918	**1.31718**	78.0259	**78.5775**	30.7148	**32.1658**
basket	0.813939	**0.819401**	3.3232	**3.23489**	**73.6173**	73.246	22.472	**21.9328**
bamboo	0.836253	**0.865376**	2.49861	**2.13859**	74.8621	**75.7512**	28.0688	**29.9176**

5. EVALUATION

To evaluate the effectiveness of the proposed camera specific image enhancement method, we applied the weights of the network to process a set of 18 sample evaluation images (cf. Fig.4) which, again, were captured image pairs of static scenery, taken once in Bayer pattern mode and applied in-camera demosaicking and once with the full rgb information. After registering them as described in Sec.4.3, the demosaicked image was processed by the trained network and the results as well as the unprocessed inputs were compared to the full rgb reference using the SSIM[11], PSNR-HVS[18], PSNR as well as the ΔE CIE2000 error metrics. The quantitative results are shown in Tab. 1 and visual close-ups of the improvements are shown in Fig. 3. On average, all four metrics showed a significant enhancement in image quality with average improvements of 1.9% (SSIM), 0.94% ($PSNR$), 4.8% (PSNR-HVS), and 11.6% (ΔE CIE2000).

6. CONCLUSIONS

We presented a simple but efficient camera specific image enhancement method to improve the image quality and reduce visual artifacts using a CNN. The method has to be trained for a specific camera type which offers the acquisition of full rgb images that do not require a demosaicking operation. It increases the perceived image quality, approximating a sub-pixel-shifted image capture with one, single capture. In addition, this method also suppresses color fringes and moiré artifacts caused by the CFA and the inherent demosaicking operation. Since the used CNN has a simple layout, it can be easily ported to GPUs or hardware and thus we envision that a method like this could be implemented as an additional in-camera post-processing step in the future. Our prototypical realization and evaluation showed that the method is able to increase the image quality of normally captured images to closer approximate full rgb high-resolution images taken with the same camera. Using a relatively small set of registered image pairs for training already led to a visible increase in image quality on all of the test samples. The approach is competitive to standard image post-processing operations while better preserving the image content and takes only several seconds to compute on standard hardware. The evaluation showed that although some moiré could be suppressed (cf. Fig. 3), more training samples of varying moiré artifacts might be needed to better eliminate such image degradations. In the future, we want to extend this approach in also applying it directly to the demosaicking operation, similar as proposed in [19]. Furthermore, an in-depth evaluation of the optimal structure of the CNN, in particular by adding skip connections, making the network deeper and using smaller kernels to build a residual network is another direction of future investigation [20].

7. REFERENCES

[1] B.E. Bayer, "Color imaging array," July 20 1976, US Patent 3,971,065.

[2] E. Chang, "High-sensitivity infrared color camera," June 28 2007, US Patent App. 11/317,129.

[3] J. Hamilton and J. Compton, "Processing color and panchromatic pixels," Feb. 1 2007, US Patent App. 11/191,538.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[5] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 664–678, May 2008.

[6] Yaniv Romano, John Isidoro, and Peyman Milanfar, "RAISR: rapid and accurate image super resolution," *CoRR*, vol. abs/1606.01299, 2016.

[7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, *Learning a Deep Convolutional Network for Image Super-Resolution*, pp. 184–199, Springer International Publishing, Cham, 2014.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[9] Isaac Amidror, *The Theory of the Moire Phenomenons*, Springer, 2009.

[10] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015.

[11] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Trans. Img. Proc.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[12] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss Functions for Neural Networks for Image Processing," *ArXiv e-prints*, Nov. 2015.

[13] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[14] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[15] Erik Reinhard, Erum Arif Khan, Ahmet Oguz Akyz, and Garrett M. Johnson, *Color Imaging: Fundamentals and Applications*, A. K. Peters, Ltd., Natick, MA, USA, 2008.

[16] Adrien Bartoli Pablo Alcantarilla (Georgia Institute of Technolog), Jesus Nuevo (TrueVision Solutions AU), "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proceedings of the British Machine Vision Conference*. 2013, BMVA Press.

[17] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

[18] P. Gupta, P. Srivastava, S. Bhardwaj, and V. Bhateja, "A modified psnr metric based on hvs for quality assessment of color images," in *2011 International Conference on Communication and Industrial Application*, Dec 2011, pp. 1–4.

[19] Y. Q. Wang, "A multilayer neural network for image demosaicking," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 1852–1856.

[20] Hantao Yao, Feng Dai, Dongming Zhang, Yike Ma, Shiliang Zhang, and Yongdong Zhang, "Dr^2-net: Deep residual reconstruction network for image compressive sensing," *CoRR*, vol. abs/1702.05743, 2017.