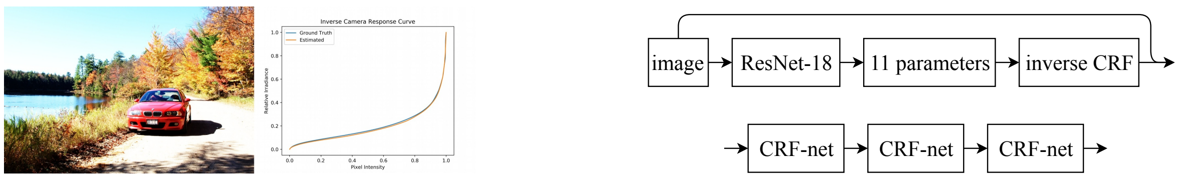
Progressive-CRF-net: Single Image Radiometric Calibration Using Stacked CNNs

Yi-Lung Kao
National Taiwan University
dragonkao730@cmlab.csie.ntu.edu.tw

Yu-Sheng Chen
National Taiwan University
nothinglo@cmlab.csie.ntu.edu.tw

Ming Ouhyoung
National Taiwan University
ming@csie.ntu.edu.tw



(a) (b)

Figure 1: (a) Input image and corresponding inverse camera response curves. (b) The proposed method.

ABSTRACT

A camera is a good instrument for measuring scene radiance. However, to please the human eye, the resulting image brightness is not linear to the scene radiance, so solving the mapping function between scene radiance and image brightness is very important. We propose a Progressive-CRF-net for radiometric calibration. By stacking multiple networks and using the pre-trained weights, this approach can reduce the training time and reach better performance than that of previous work. Our experiments show a significant improvement based on PSNR and SSIM.

CCS CONCEPTS

• **Computing methodologies → Camera calibration**;

KEYWORDS

Radiometric Calibration, Camera Response Function, Convolutional Neural Network

ACM Reference Format:

Yi-Lung Kao, Yu-Sheng Chen, and Ming Ouhyoung. 2018. Progressive-CRF-net: Single Image Radiometric Calibration Using Stacked CNNs. In *Proceedings of SIGGRAPH '18 Posters.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3230744.3230770

1 INTRODUCTION

Radiometric calibration is an important problem in the fields of computer vision and computer graphics. When photographers utilize cameras for image creation, the camera itself will directly affect the results of the shooting. Each camera vendor applies different camera response curves and color renditions to various cameras, so experienced consumers need a lot of time to carry out the task of color correction and color grading. The whole process is very annoying and tedious. Therefore, several of radiometric calibration algorithms try to help artists to finish their job in a skillful way.

In 1997, Debevec and Malik [Debevec and Malik 1997] proposed an algorithm to reconstruct High Dynamic Range (HDR) Image by fusing different exposure images with estimated camera response function (CRF). At that time, most calibration approaches are based on the ideas of exposure stack [Mitsunaga and Nayar 1999] or taking photos with a specific color chart [Chang and Reid 1996].

In past few years, there are certain methods that try to use a single image to do the radiometric calibration without an additional color chart in the scene [Lin et al. 2004; Lin and Zhang 2005; Ng et al. 2007]. Unsurprisingly, predicting the camera response function with a single image is very difficult. But with the development of deep learning technology, the calibration problem can be dealt with in a different way. Li and Peers [Li and Peers 2017] conducted the radiometric calibration with a CRF-net by using a convolutional neural network (CNN). Due to the power of CNN, the result is very robust to the images with suitable exposure, however, it has still shown performance on too dark or too bright images.

We propose an approach to improve over the CRF-net. Our Progressive-CRF-net duplicates CRF-net a number of times and each CRF-net rectifies the input photo to be closer to linear space. Through this way, we can directly apply the same convolutional weight to each Progressive-CRF-net component from a single pre-trained CRF-net.

2 OUR APPROACH

2.1 CRF-net

Basically, we follow the original CRF-net [Li and Peers 2017] architecture. However, we modify the output space because we need to transfer the input image from non-linear space to linear space. Specifically, the original CRF-net [Li and Peers 2017] outputs an estimate of the camera response function [Grossberg and Nayar 2003] in the form of an 11-parameter EMoR model[Grossberg and Nayar 2003]. We alter the outputs to predict the "inverse" camera response function. Then we can directly linearize the input image

SIGGRAPH '18 Posters, August 12-16, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5817-0/18/08.
https://doi.org/10.1145/3230744.3230770

to the linear space image. Figure 1b upper part shows the pipeline of our CRF-net.

2.2 Progressive-CRF-net: stacked CRF-nets

To improve the performance of the CRF-net, the simplest way is by increasing the total number of network parameters. However, altering the network architecture means we have to train the new network from scratch. In our experience, training a CRF-net from the ground up takes almost one week with a Nvidia Tesla K80, which is a very time consuming process. We chose an elegant method: we concatenate three CRF-nets and initialize each net with the weight from a single pre-trained CRF-net. The idea comes from FlowNet 2.0 [Ilg et al. 2017]. To achieve better performance, it needs a bigger network architecture than the original FlowNet. Hence, FlowNet 2.0 stacks multiple FlowNets since the optical flow result can be iteratively refined. Similarly, the result of radiometric calibration can also be iteratively refined. Simultaneously, by using the pre-trained weight for each component, the training process can be accelerated. For instance, training the Progressive-CRF-net with the pre-trained weight can converge in less than one day. We can see the entire architecture in Figure 1b lower part.

2.3 Loss Function

The original CRF-net [Li and Peers 2017] directly minimize the Eq. 1, the Euclidean distance between the ground truth and output CRF(camera response function).

$$L_{CRF} = \frac{1}{N+1} \sum_{n=0}^{N} \left\| f(\frac{n}{N}) - \hat{f}(\frac{n}{N}) \right\|^2 \tag{1}$$

where f and $\hat{f}$ are the labels and the prediction of camera response function in a discrete form respectively, and N+1 is the total number of the CRF discrete points. However, this loss function is not reasonable for an image which is too bright or too dark. For example, if the input image is too bright, it lacks the information for response function in the dark region and thus is impossible to figure out the response function in the dark region, vice versa. Therefore, we minimize the mean square error between the ground truth linear image and our output rectified image. Specifically, we minimize the loss in the Eq. 2:

$$L_{pixel} = \frac{1}{P} \sum_{p=1}^{P} \left\| Y_p - \hat{f}^{-1}(X_p) \right\|^2 \tag{2}$$

where X and Y are the input image and the linear space image of the input respectively, and P is the total number of the pixels in the input data. By this way, it will not force the neural networks to predict the camera response curve across all the dynamic range which may not make sense for the input image. In other words, our loss function will only focuses on the error within the pixel space which is more rational to achieve.

2.4 Dataset

We collected HDR images from the Internet. For the training and testing inputs, we sampled the exposure value (EV) from -2 to 2 and applied camera response function from DoRF [Grossberg and Nayar 2003] to produce the non-linear low dynamic range input image, X, which can be formulated as:

$$X = f(clip(H \times t)) \tag{3}$$

where H is the HDR image, t is given as an exposure value, and the $clip$ function will clip the value within the range [0, 1]. The corresponding linear low dynamic range image, Y, will skip the non-linear mapping function which is:

$$Y = clip(H \times t) \tag{4}$$

Table 1: Test Result

Network	PSNR	SSIM	Loss Function
CRF-net [Li and Peers 2017]	26.34	0.8938	L_{CRF}
Our CRF-net	30.95	0.9300	L_{pixel}
Progressive-CRF-net	34.53	0.9653	L_{pixel}

3 CONCLUSION

We improve over the original CRF-net. Our network architecture can increase network parameters elegantly and speed up the training process. In our experience, training a CRF-net needs more than one week with a Nvidia Tesla K80. Our Progressive-CRF-net which contains triple CRF-nets only needs one day to converge. The overall testing performance is shown in Table 1.

ACKNOWLEDGEMENT

This project was partially supported by Ministry of Science and Technology (MOST), Taiwan: No. 106-3114-E-002-012, No. 105-2221-E-002-128-MY2, and Mediatek Inc. No. MTKC-2018-0167. Thanks "Mark Fairchild's HDR Photographic Survey" for providing high-quality HDR image dataset.

REFERENCES

Young-Chang Chang and John F Reid. 1996. RGB calibration for color image analysis in machine vision. *IEEE Transactions on image processing* 5, 10 (1996), 1414–1422.

Paul E Debevec and Jitendra Malik. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques.* ACM Press/Addison-Wesley Publishing Co., 369–378.

Michael D Grossberg and Shree K Nayar. 2003. What is the space of camera response functions?. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, Vol. 2. IEEE, II–602.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2.

Han Li and Pieter Peers. 2017. CRF-net: Single Image Radiometric Calibration using CNNs. In *Proceedings of the 14th European Conference on Visual Media Production (CVMP 2017).* ACM, 5.

Stephen Lin, Jinwei Gu, Shuntaro Yamazaki, and Heung-Yeung Shum. 2004. Radiometric calibration from a single image. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Vol. 2. IEEE, II–II.

Stephen Lin and Lei Zhang. 2005. Determining the radiometric response function from a single grayscale image. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 2. IEEE, 66–73.

Tomoo Mitsunaga and Shree K Nayar. 1999. Radiometric self calibration. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, Vol. 1. IEEE, 374–380.

Tian-Tsong Ng, Shih-Fu Chang, and Mao-Pei Tsui. 2007. Using geometry invariants for camera response function estimation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE, 1–8.