

Local Linear Learned Image Processing Pipeline

Steven Lansel and Brian A. Wandell

Electrical Engineering Department, Stanford University, Stanford, CA-94305, USA.

Psychology Department, Stanford University, Stanford, CA-94305, USA.

slansel@stanford.edu, wandell@stanford.edu

Abstract: The local linear learned (L^3) algorithm is presented that simultaneously performs the demosaicking, denoising, and color transform calculations of an image processing pipeline for a digital camera with any color filter array.

© 2011 Optical Society of America

OCIS codes: 040.1490, 100.2000.

1. Introduction

The recent increase in the number of megapixels in digital cameras has resulted in images that have a higher spatial resolution than is required by most imaging applications. This excess of pixels offers an opportunity to redesign imaging sensors by altering the color filter array (CFA) to achieve improvements in other aspects of photography than spatial resolution. The CFA is a mosaic of optical filters overlaying each photosensitive site on the sensor that controls the spectral sensitivity of the pixel. Figure 1 shows four possible CFA designs. The Bayer CFA shown in Figure 1(a) is in almost all cameras today. Figure 1(b) shows a CFA where half of the green pixels from the Bayer CFA are replaced with white pixels for low light imaging [1].

For cameras with new CFA designs, there is a challenge in designing the image processing pipeline, the series of calculations that transforms the raw output from the sensor into a desirable image. In general, one needs to perform demosaicking to estimate a full color image from the mosaicked image where only a single band is measured at each pixel. Denoising is required to suppress noise that exits due to photon shot noise and imperfections in the sensor electronics. Surveys of existing demosaicking and denoising algorithms are provided in [2] and [3], respectively. Finally, a transformation is needed from the input color space defined by the sensors spectral sensitivities to a desired output color space such as XYZ . The output images can then be converted to a standard color space such as $sRGB$. The algorithm can estimate output images in any color space, which is required for multispectral imaging.

The Local Linear Learned (L^3) algorithm is presented that simultaneously performs demosaicking, denoising, and color conversion for an arbitrary CFA. Although there exist numerous image processing algorithms for camera pipelines, very few have the ability to operate on any CFA or can perform any of these calculations simultaneously. The L^3 algorithm learns from a training set of images how to distinguish between flat and texture regions and optimal linear filters for these two regions. As a result, the algorithm can be trained for a specific CFA with a training set appropriate for a particular imaging application. The algorithm enables rapid design and testing of future CFAs by automatically generating the image processing pipeline. An overview of the algorithm is presented in Figure 2.

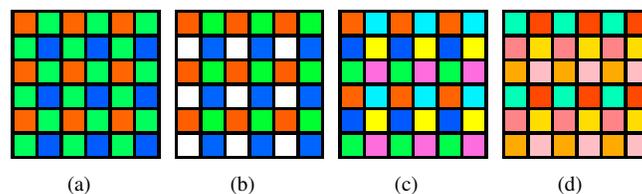
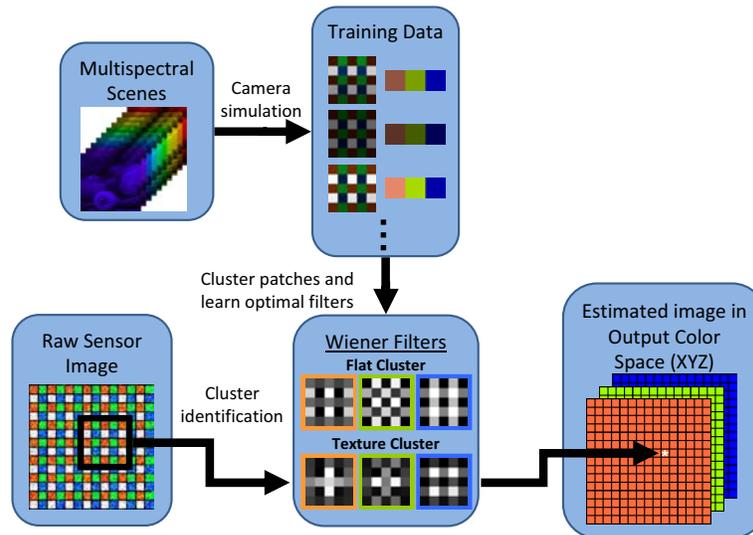


Fig. 1. **Example CFA patterns.** (a) Bayer pattern, (b) $RGBW$ CFA with translucent pixel for low light imaging, (c) 6-band CFA for improved color or multispectral imaging, and (d) 6-band CFA for possible medical use.

IMC3.pdf

Fig. 2. Overview of L^3 pipeline.

2. Generating Training Data by Simulation

The training set contains a collection of CFA measurements that contain little or no noise and the corresponding desired output images, which may be calculated, designed, or measured as desired. The learned pipeline will try to match the desired output images as closely as possible. A large number of images are not required since each image contains many pixels, but the training images should be representative of data for the intended application.

To generate training data and evaluate algorithms, digital cameras were simulated using the Image Systems Evaluation Toolkit (ISET) [4]. The simulations accurately account for the scene, optics, and sensor. Computer simulations have the advantage that a physical camera prototype is not needed, which enables fast design optimization and testing. The scene is modeled using a specified illuminant and multispectral images that provide the reflectance at each spatial location for the different wavelengths of light.

3. Algorithm Description

The L^3 pipeline classifies each pixel in the sensor image as being in a flat (uniform) or texture (contrast) patch and learns filters that use nearby measurements to estimate the output colorbands at that pixel. These filters are optimized over the training set and adapt to the expected noise in the measurements by depending on the scene's luminance.

The algorithm is spatially localized: each pixel in the final image is a function only of the nearby sensor measurements. This enables parallel processing. We use a square neighborhood of $m \times m$ pixels, referred to as an image patch. Since only the center pixel output values are estimated, the number of patches that must be processed equals the number of image pixels. For an RGB array there are four types of patches (R, B and two G s).

Patch Classification: In order to make the algorithm adaptive (local instead of global), patches are classified into flat and texture clusters. The flat patches are relatively uniform image areas that contain only low spatial frequencies. Texture patches contain higher frequencies and appear as edges or textures. To classify patches as flat or texture, the CFA values at the center pixel are estimated. For each pixel in the patch, subtract the estimate that corresponds to that pixel's color, which removes the overall color of the patch. If the patch is uniform, the transformed patch will now be identically 0. The amount that each value in the transformed patch deviates from 0 is a measure of the presence of texture in the patch. Therefore, the absolute sum of the transformed patch is calculated and compared to a threshold to determine if the patch is flat or texture.

Learning Filters: Next, we find optimal filters that estimate the desired output colors at the center pixel of each patch in each cluster. Separate Wiener filters are found for the flat and texture clusters, patch type, and scene luminance. Optimal filters for flat clusters are spread over space to reduce noise by averaging with little risk of blurring the signal. Optimal filters for texture clusters are relatively compact to avoid combining measurements across an edge.

The training data for each cluster, patch type, and scene luminance consists of k chosen from a set of training

IMC3.pdf

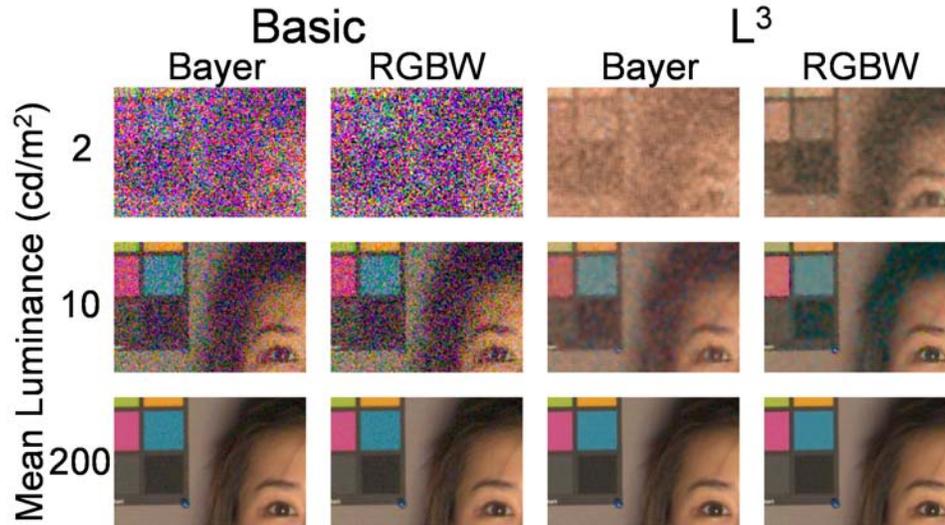


Fig. 3. **Estimated images under various light levels.** Images are the result of reconstructing a scene from simulated camera measurements. Images are cropped to 240x280 pixels. Original image is omitted because it is very similar to the image on the bottom-right.

images. Let the columns of $Y \in \mathbb{R}^{m^2 \times k}$ be vectorized CFA patches from the sensor and the columns of $X \in \mathbb{R}^{o \times k}$ be the ideal values of the o output color bands at the center pixel of each patch. Although we assume Y is noise-free for training, we need the filtering to be robust to measurement noise. Let $N \in \mathbb{R}^{m^2 \times k}$ be a random variable representing measurement noise so that $X + N$ is observed. We assume the columns of N are independent of X and Y and identically distributed with mean 0 and autocorrelation R_n even though this is not true when testing.

We wish to learn the linear estimator $W \in \mathbb{R}^{o \times m^2}$ so that the estimate $\hat{X} = W(Y + N)$ is most similar to X . Generally $m^2 < k$, so we cannot obtain perfect estimation. Instead the sum of the squares of the errors of the estimates is minimized for convenience even though the metric may not match visual perception. Under these assumptions, the optimal linear filter is given by the Wiener filter, which can be found by solving $W(Y Y^T + k R_n) = X Y^T$.

4. Results

To test the pipeline, the capture of a scene was simulated using both Bayer and *RGBW* CFAs. The images were processed using the L^3 pipeline and a basic pipeline consisting of bilinear demosaicking and a linear transformation from the CFA color space to *XYZ*. The *XYZ* images were then converted to *sRGB* for display in Figure 3. The L^3 pipeline was trained on six similar portrait scenes using 9×9 patches and a threshold that results in 88% of training patches being flat.

Notice that there is more noise in the images from the basic pipeline than the L^3 pipeline. The images from the darkest scenes are very poor except for the L^3 processing of the *RGBW* CFA, which shows the advantage of the algorithm and white pixels. For such a dark scene, the colors are washed out due to the reliance on the white channel because it has better signal to noise ratio.

References

1. M. Parmar and B. A. Wandell, "Interleaved imaging: an imaging system design inspired by rod-cone vision," in *Digital Photography V, SPIE*, Vol. 7250, 2009.
2. X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: a systematic survey," in *Visual Communications and Image Processing 2008, SPIE*, Vol. 6822, 2008.
3. M. Motwani, M. Gadiya, R. Motwani, and F. Harris, Jr., "A Survey of Image Denoising Techniques," in *Proceedings of GSPx 2004*, 2004.
4. J. Farrell, P. Catrysse, and B. Wandell, "The Digital Camera is an Imaging System," in *Imaging Systems, OSA technical Digest*, 2010.