

【paper-note1】unprocessing images for learned raw denoising

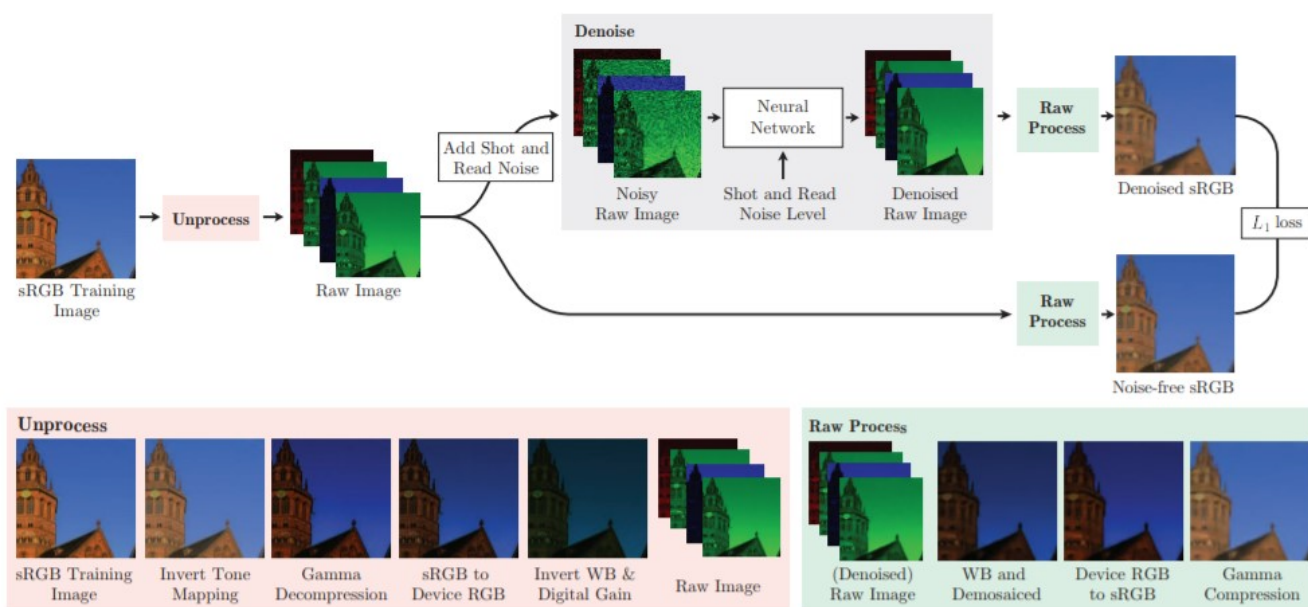
论文作者：Tim Brooks 谷歌研究院，UC Berkeley

收录：CVPR2019

核心思想

提出了 `unprocess` 的方法，把普通的jpg图片转化为raw格式文件，文章对相机isp算法pipeline进行建模，称之为 `Raw Process`，然后对pipeline中的每个操作都求逆操作，称之为 `unprocess`，以此可以把sRGB的训练图片转化为Raw Image。

这么做的原因是，在单图像去噪任务中，sRGB域的图像所含的噪声非常复杂，难以建模，但是相机传感器得到Raw文件的噪声容易建模，在该领域之前也有相当多的工作，如 [1, 2, 3, 4]。本文把相机传感器的噪声分为了 `shot noise` 和 `read noise` 两种噪声，具体在后面详述。得到噪声的建模后，该方法引入了U-net [5] 作为自己的训练网络，输入合成的噪声raw image，输出去噪后的raw image，并把unprocessed raw文件作为ground truth，最后计算输出和GT的L1 loss。如下图：



raw image pipeline

算法的重要内容就是对isp算法的求逆，分为7个步骤：

1. Shot and Read Noise

该步骤对raw文件的噪声进行建模，主要来源于文献 [1]。相机传感器的噪声主要来源于两个方面，第一：光子到达的统计数据（被称为“shot noise”），第二：读取电路的不准确性（被称为“read noise”）。

其中，`shot noise` 是一个服从泊松分布的随即变量，其均值为真实的光强 x （泊松分布的均值和方法相等），而 `read noise` 是一个服从近似高斯分布的随机变量，其均值为0，并拥有固定的方差。

把这两个噪声放到一起，得到一个单异方差高斯分布（single heteroscedastic Gaussian），观测值 y 是一个服从高斯分布的随机变量，其方差是 x 的函数：

$$y \sim \mathcal{N}(\mu = x, \sigma^2 = \lambda_{read} + \lambda_{shot}x).$$

$$\lambda_{read} = g_d^2 \sigma_r^2, \quad \lambda_{shot} = g_d g_a.$$

g_d 是数字增益， g_a 是模拟增益，这两个值是相继曝光值的直接函数， σ_r 是read noise的固定方差。

采样值：

$$\log(\lambda_{shot}) \sim \mathcal{U}(a = \log(0.0001), b = \log(0.012))$$

$$\log(\lambda_{read}) \mid \log(\lambda_{shot}) \sim$$

$$\mathcal{N}(\mu = 2.18 \log(\lambda_{shot}) + 1.2, \sigma = 0.26).$$

建模结果：

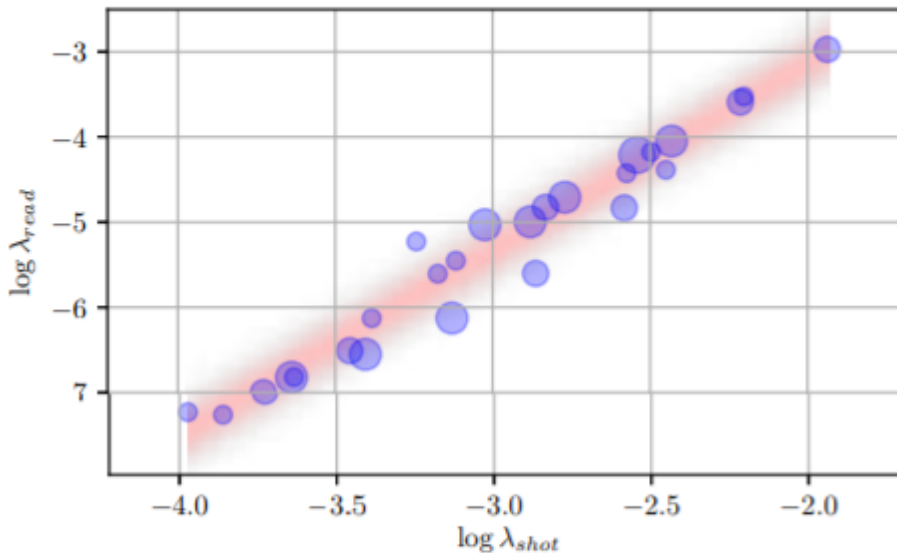


Figure 3. Shot and read noise parameters from the Darmstadt dataset [31]. The size of each circle indicates how many images in the dataset shared that shot/read noise pair. To choose the noise level for each synthetic training image, we randomly sample shot and read noise parameters from the distribution shown in red.

2. 去马赛克

该步骤很简单，raw—>jpg：双线性插值法，jpg—>raw：根部bayer阵列，每个像素的三个颜色中丢掉其中两个颜色。

3. 数字增益

数字增益来源于自动曝光算法，大部分相机的自动曝光算法是个黑盒，很难建模，本文就假设图像强度服从不同的指数族分布：

$$p(x; \lambda) = \lambda e^{-\lambda x}$$

对于 $x > 0$ ，得到 λ 的极大似然估计为样本均值的倒数，这就意味着放大 x 即为缩小 λ 。

数据集的放大比率为1.25，则

$$\lambda = 1/1.25 = 0.8, \text{ 去正态分布, 均值为0.8, 方差为0.1, } [0.5, 1.1].$$

4. 白平衡

图像=光照颜色*物体颜色。

白平衡求逆是个分段函数，主要是为了解决高光缺失问题，因为逆白平衡增益通常比真实的要小。

$g > 1$ and $x > t, t = 0.9$:

$$\alpha(x) = \left(\frac{\max(x - t, 0)}{1 - t} \right)^2$$

$$f(x, g) = \max \left(\frac{x}{g}, (1 - \alpha(x)) \left(\frac{x}{g} \right) + \alpha(x)x \right)$$

$x \leq t$,

$$f(x, g) = x/g$$

$g \leq 1$,

$$f(1, g) = 1$$

如图：

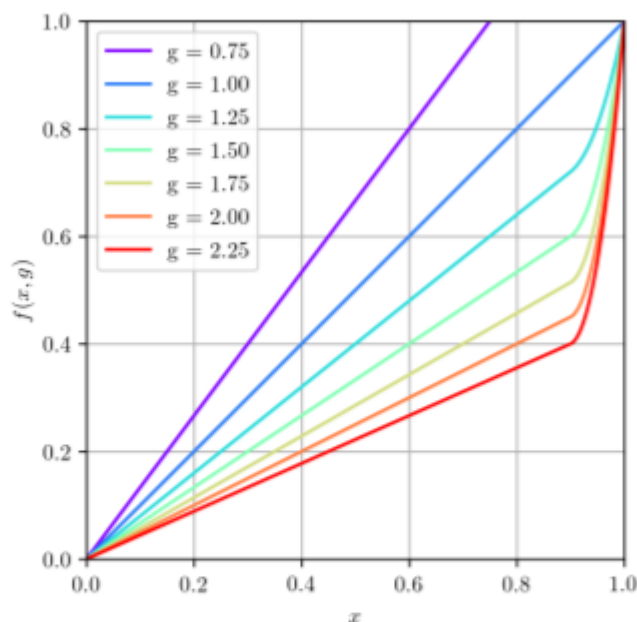


Figure 4. The function $f(x, g)$ (defined in Equation 6) we use for gaining down synthetic image intensities x while preserving highlights, for a representative set of gains $\{g\}$.

5. 色彩校正

把device RGB转变为sRGB需要用到3x3的色彩校正矩阵（CCM），每个相机的CCM都不一样，本文对四个相机的CCM做了凸组合，并求逆，得到device RGB的图像。

6. gamma 压缩

标准gamma曲线：

$$\Gamma(x) = \max(x, \epsilon)^{1/2.2}$$

简单求逆：

$$\Gamma^{-1}(y) = \max(y, \epsilon)^{2.2}$$

7. 色调映射

为了迎合胶片的特性，应用S形曲线。

简单假设色调映射曲线是一个“smoothstep”曲线：

$$\text{smoothstep}(x) = 3x^2 - 2x^3$$

求逆：

$$\text{smoothstep}^{-1}(y) = \frac{1}{2} - \sin\left(\frac{\sin^{-1}(1 - 2y)}{3}\right)$$

至此，本文的主要方法以阐述完毕，剩下的就是用U-net建模，详细的可以看论文。

参考文献

- [1] Samuel W. Hasinoff. Photon, poisson noise. In Computer Vision: A Reference Guide. 2014.
- [2] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. IEEE TPAMI, 30(2):299–314, 2008. 2, 3
- [3] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. IEEE TIP, 17(10):1737–1754, 2008. 3
- [4] H. J. Trussell and R. Zhang. The dominance of Poisson noise in color digital cameras. In IEEE ICIP, pages 329–332, 2012. 3, 8
- [5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. MICCAI, 2015.