

作业1-高动态范围成像与色调映射

介绍

本次作业的目的是探讨高动态范围（HDR）成像、噪声校准、色彩校准和色调映射。如我们在课上讨论的，HDR成像可以用于创建浮点精度的图像，这些图像能线性映射到场景辐射值。噪声校准是找出你的相机的噪声特性，并利用它们来提高HDR图像的过程。色彩校准确保你在图像中看到的颜色与一些基准RGB值相匹配。色调映射算法将HDR图像的动态范围压缩到8位范围，以便它们可以在显示器上显示。要获得满分，你需要将这些步骤应用于我们提供的一个曝光堆栈和你自己捕获的一个曝光堆栈。最后，作为额外加分，你可以研究在你同时改变快门速度和ISO从而改变曝光下的HDR成像。

在整个作业中，我们提及了一些在课程参考阅读资料中提到的关键论文。虽然作业和课堂幻灯片描述了你执行的大多数步骤，但我们强烈建议你阅读相关的论文。

在本文档的末尾有一个“提示和信息”部分，这部分可能会有所帮助。我们强烈建议你在开始处理作业之前完整阅读该部分。本次作业所需的Python包包括numpy、skimage、matplotlib和cv2（OpenCV，用于读取和写入HDR文件），你可以使用作业ZIP档案中的./src/cp_hw2.py文件中提供的函数。

注：该作业改编自CMU 2021年计算摄影学课程15-463的作业2，另外虽然所有参考信息都可以通过查询互联网和相关文献来得到（例如我在知乎专栏《计算摄影学》中撰写的文章），但其中一些我专门标注出来了可以参考第1季课程《从光线到图像-相机成像的原理》，我强烈建议你选修该课程以打好这一季作业的坚实基础，这有利于你更加顺畅的完成此次作业

1 HDR成像 (60 points)

对于这部分以及接下来的两个部分（色彩校正和色调映射），你将使用尼康 D3300拍摄的一个办公室中曝光图像堆栈。图像文件位于作业ZIP存档的./data/door_stack目录中。图1显示了两个曝光下的图像，以及一个经过色调映射的HDR合成图。

虽然场景不是特别美观，但它具有一些特点，使其成为HDR的良好示例：

- 首先，有两个具有非常不同的照明和动态范围的区域，没有单个曝光可以同时正确捕获。
- 其次，两个区域都包括颜色丰富的物品（明亮区域背景中的Toy Story海报，Plus-Plus块，SIGGRAPH杯和暗区的书籍封面），你可以用它们来评估结果的颜色呈现。
- 第三，焦点区域具有高细节特征（书籍封面上的字母和镜头/相机标记），你可以用它们来评估结果的分辨率。最后，场景包括一个色卡，你可以在附加问题中用它进行色彩校准。

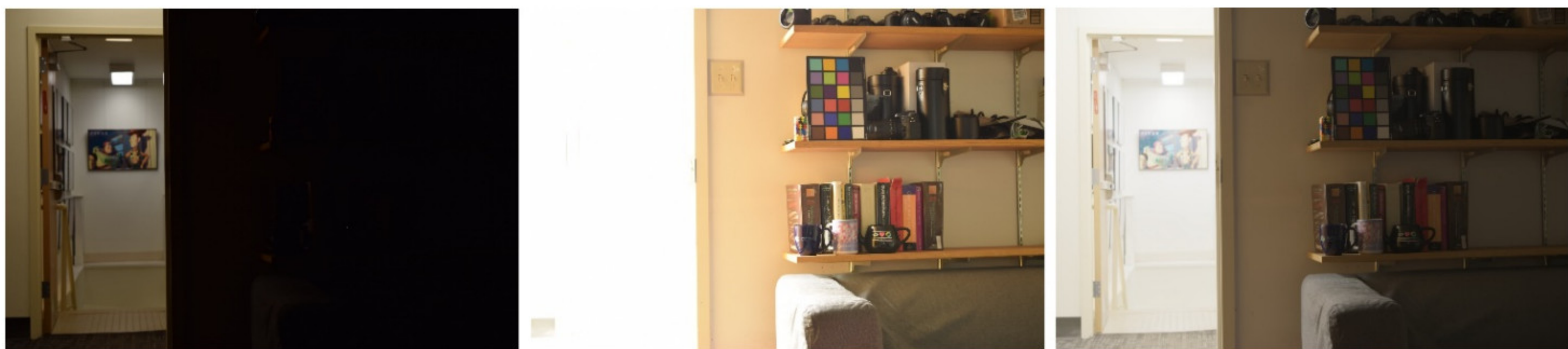


图1：从左到右：两个LDR曝光，以及使用摄影色调映射的HDR合成图。

你会注意到，在数据文件夹中，有两组图像，RAW (.NEF) 和RGB (.JPG)。正如我们在课堂上讨论的，HDR成像对于RAW和RGB图像是不同的。为了体验这种差异，在本次作业中，你将从两组图像中创建HDR图像。

作为参考，我们使用固定的光圈和ISO捕获了两个曝光堆栈，并且快门速度等于 $\frac{1}{2048} \cdot 2^{k-1}$ ，其中 $k \in \{1, \dots, 16\}$ 是图像文件名中的索引。

1.1 预处理RAW图像 (5 points).

使用dcraw将RAW.NEF图像转换为 线性 16-bit.TIFF图像。为此，你应该利用dcraw使用相机的白平衡配置文件进行白平衡，使用高质量插值进行解析，并使用sRGB作为输出颜色空间。阅读dcraw的文档，并找出进行此转换的正确标志集。确保在你提交解决方案的报告中报告你使用的标志。

dcraw的官网是<https://www.dechifro.org/dcraw/>，你需要从网络上查询相关软件的使用法。你也可以参考计算摄影学课程第1季《从光线到图像》的大课程作业以及第1季第4课《图像传感器与ISP》中的内容，了解更多的知识。

通过这一步，你将得到一系列TIFF图像，它们中的值是线性的，这有利于你之后做HDR图像的合成。

1.2 处理非线性化图像(25 points).

与线性的RAW图像不同，RGB图像中的值是非线性的。正如我们在课上看到的，在你可以将它们合并成HDR图像之前，你首先需要执行辐射校准以消除这种非线性。你将使用Debevec和Malik的方法[1]来完成这个任务。我们下面将介绍这个方法的工作原理，但是强烈建议你至少阅读这篇论文的第2.1节，该节解释了该方法。

图像 k 的像素 $\{i, j\}$ 上的强度 $I_{ij}^k \in \{0, \dots, 255\}$ 与某未知场景光通量值 L_{ij} 有关，如下式所示：

$$I_{ij}^k = f(t^k L_{ij}), \quad (1)$$

其中 t^k 是图像 k 的（已知）曝光时间，而 f 是相机应用的未知非线性函数。如果我们知道 f^{-1} ，我们可以将 I_{ij}^k 转换回线性度量。

你将找回函数 $g := \log(f^{-1})$ ，它将像素值 I_{ij}^k 映射到 $g(I_{ij}^k) = \log(L_{ij}) + \log(t^k)$ 。这是由人类视觉系统对于对数而非线性强度有响应的事实所导致的。由于 g 的定义域是离散的强度值 $\{0, \dots, 255\}$ ， g 本质上只是一个256维向量。

解决这256个值似乎不可能，因为我们既不知道 g 也不知道 L_{ij} 。然而，如果在捕获曝光堆栈时图像场景保持静态，我们可以利用 L_{ij} 在所有LDR图像中的值是常数的事实。然后，我们可以通过解决以下最小二乘优化问题来找回 g ，

$$\min_{g, L_{ij}} \sum_{i,j} \sum_k \{w(I_{ij}^k/255) [g(I_{ij}^k) - \log(L_{ij}) - \log(t^k)]\}^2 + \lambda \sum_{z=0}^{255} \{w(z/255) \nabla^2 g(z)\}^2 \quad (2)$$

方程 (2) 中的第一项是 **数据项**，它鼓励 g 和 L_{ij} 的值使得线性图像中的强度与曝光时间线性缩放。正如我们在课堂上讨论的，权重 w 与线性估计应该更多地依赖于曝光良好的像素而非曝光不足或曝光过度的像素有关。在问题1的后面部分 ("权重方案") 中看一下你应该使用什么权重。注意， w 的输入被除以255，因为权重的定义假定输入强度在范围 $[0, 1]$ 内，而你使用的强度在范围 $\{0, \dots, 255\}$ 内。

方程 (2) 中的第二项是一个 **正则化项**，它通过惩罚具有大二阶导数幅度的解 g 来鼓励 g 是平滑的。考虑到 g 是离散的，可以使用拉普拉斯滤波器来近似二阶导数，即 $\nabla^2 g(z) = g(z+1) - 2g(z) + g(z-1)$ 。参数 λ 控制这种正则化如何影响最终结果；它是一个超参数，你需要进行实验。注意，当使用需要知道曝光时间的光子最优权重 w_{photon} 时，你可以将_仅正则化项_的权重设置为常数（例如， $w(z) = 1$ ）。

通过将其表示为矩阵形式来解决方程 (2) 的 **最小二乘** 优化问题：

$$\|\mathbf{A}\mathbf{v} - \mathbf{b}\|^2, \quad (3)$$

其中 $\mathbf{A}$ 是一个矩阵， $\mathbf{v} = [g; \log(L_{ij})]$ 是未知数， $\mathbf{b}$ 是一个已知向量。然后，使用NumPy的求解器之一来找回未知数。（参见numpy函数 `numpy.linalg.lstsq`。）

尽管Debevec和Malik [1] 为每个颜色通道找回了不同的 g ，但是对于这次作业，我们建议你同时处理所有三个通道的像素，以找回所有通道的单一 g 。这有助于减少最终HDR合成图中的颜色伪像。

绘制你找回的函数 g ，然后用它将非线性图像 I_{ij}^k 转换为线性图像，

$$I_{ij,\text{lin}}^k = \exp(g(I_{ij}^k)). \quad (4)$$

注意，你不会使用从方程 (2) 中解出的值 L_{ij} 。

1.3 将曝光堆栈合并为HDR图像 (30分)

现在我们有两组（大致上）线性的图像，分别来自RAW和RGB文件，现在是时候将它们分别合并为各自的HDR图像了。这部分对于两组线性图像都是通用的。确保你创建的每个HDR图像只使用其中一组图

像，不要混用。

给定一组 k 个对应于不同曝光 t^k 的LDR线性图像，我们可以将它们合并为HDR图像，无论是在线性域还是对数域。线性合并的动机是物理上的精度，而对数合并的动机，则是人类视觉的感知。

我们首先引入一些符号。我们使用 $I_{ij,\text{LDR}}^k$ 来指代 k -th 原始 LDR输入图像的像素 $\{i, j\}$ 的强度值，直接从.JPG或.TIFF文件中读取。我们使用 $I_{ij,\text{lin}}^k$ 来指代 k -th 线性 LDR输入图像的像素 $\{i, j\}$ 的强度；这是方程 (4) 中的强度，当使用.JPG文件时，或者当使用.TIFF文件时与 $I_{ij,\text{LDR}}^k$ 相同。另外，从这点开始，我们假设 $I_{ij,\text{LDR}}^k \in [0, 1]$ 。因此，你需要将原始的LDR输入图像归一化到 $[0, 1]$ 范围，当使用.JPG文件时，你可以通过除以255来完成，当使用.TIFF文件时，你可以通过除以 $2^{16} - 1$ 来完成。

有了这个符号，当使用线性合并时，我们构造HDR图像如下：

$$I_{ij,\text{HDR}} = \frac{\sum_k w(I_{ij,\text{LDR}}^k) I_{ij,\text{lin}}^k / t^k}{\sum_k w(I_{ij,\text{LDR}}^k)}. \quad (5)$$

当使用对数合并时，我们构造HDR图像如下：

$$I_{ij,\text{HDR}} = \exp \left(\frac{\sum_k w(I_{ij,\text{LDR}}^k) (\log(I_{ij,\text{lin}}^k + \epsilon) - \log(t^k))}{\sum_k w(I_{ij,\text{LDR}}^k)} \right), \quad (6)$$

其中 ϵ 是一个小常数，以避免对数函数在0处的奇点。和前面一样，方程 (5) 和 (6) 中的权重 w 可以用来更多地强调曝光良好的像素，而减少对曝光不足或曝光过度的像素的强调。见下文关于使用什么权重。

为两个曝光堆栈实现线性和对数合并。然后，将结果HDR图像存储为.HDR文件，这是一个开源高动态范围文件格式。（参见./src/cp_hw2.py中提供的函数writeHDR）

1.4 权重方案

存在许多可能的权重方案选择 [3]。你需要实现下面四个：

$$w_{\text{uniform}}(z) = \begin{cases} 1, & \text{if } Z_{\min} \leq z \leq Z_{\max} \\ 0, & \text{otherwise} \end{cases},$$

$$w_{\text{tent}}(z) = \begin{cases} \min(z, 1 - z), & \text{if } Z_{\min} \leq z \leq Z_{\max} \\ 0, & \text{otherwise} \end{cases},$$

$$w_{\text{Gaussian}}(z) = \begin{cases} \exp\left(-4\frac{(z-0.5)^2}{0.5^2}\right), & \text{if } Z_{\min} \leq z \leq Z_{\max} \\ 0, & \text{otherwise} \end{cases},$$

$$w_{\text{photon}}(z, t^k) = \begin{cases} t^k, & \text{if } Z_{\min} \leq z \leq Z_{\max} \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

所有上述权重方案都假设强度值在范围 $z \in [0, 1]$ 内。你可以尝试使用不同的裁剪值 $Z_{\min}$ 和 $Z_{\max}$ ，但我们推荐 $Z_{\min} = 0.05$, $Z_{\max} = 0.95$ 。与其他方案不同，权重 w_{photon} 也取决于捕获像素时的曝光。

请注意，当从JPG堆栈创建HDR图像时，你需要在方程（2）（线性化）和方程（5）-（6）（合并）中使用相同的权重方案。

实现所有上述权重方案，并使用它们来创建HDR图像。总共，你将创建16个HDR图像：2组图像（RAW和RGB） $\times$ 2合并方案（线性和对数） $\times$ 4权重方案（均匀，帐篷，高斯和光子噪声最优）。

关于光子噪声的更多知识，请参看第二季第2课《图像噪声深入分析及处理》

1.5 做出你的选择

从你创建的十六个HDR图像中选择一个。例如，你可以选择你认为最美观的一个。在报告中清晰描述你为什么选择了你做的图像。请注意，由于你尚未对HDR图像进行色调映射，如果你直接显示它们，它们看起来不会很好看；请参见后面的"提示和信息"。

2 颜色校正和白平衡 (20分)

对于这部分，你将使用在第1部分末尾选择的HDR图像。如图2左侧所示，你的色调映射图像在房间的暗部分往往会呈现出橙色色调。这是因为房间内的非常低的光线和房间外光线的大对比度使得相机的自动白平衡失效。此外，即使白平衡工作得很完美，我们对各种图像合成所处的颜色空间也没有非常小心。

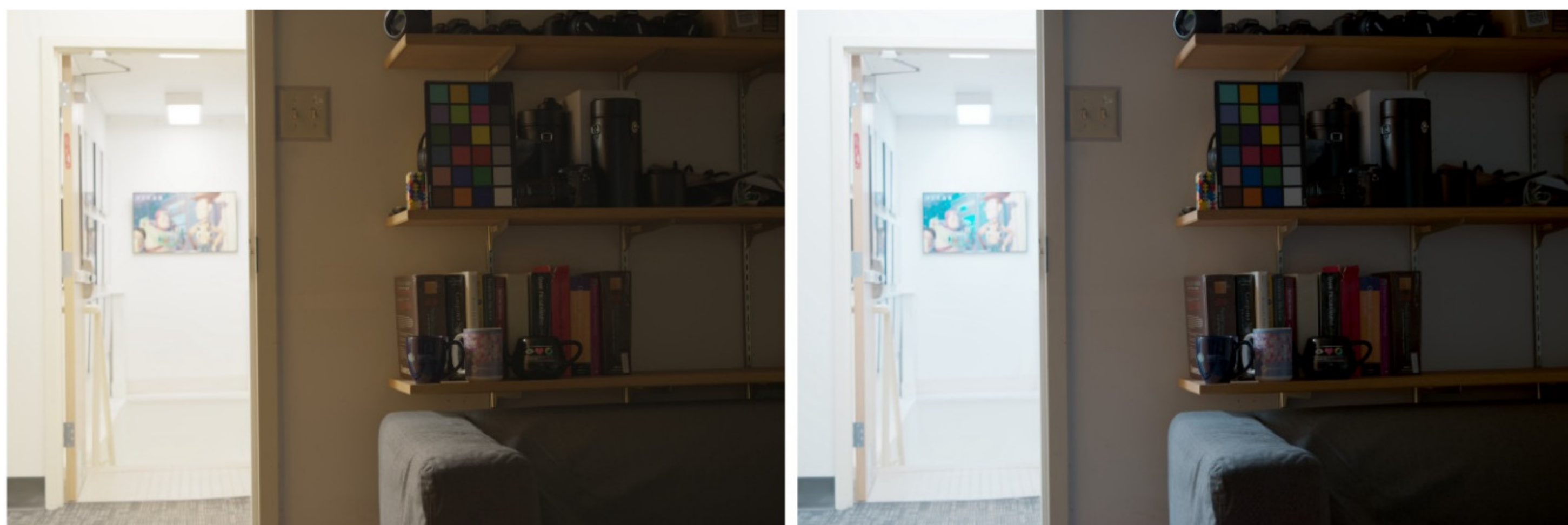


图2：未经颜色校正（左）和经过颜色校正（右）的色调映射HDR图像。

你可以应用我们在计算摄影学课程第1季《从光线到图像-相机成像的原理》中讨论的任何自动白平衡算法来改善这个问题。但是，由于图像中包含一个色卡（图3），我们可以做得更好，进行更准确的颜色校正。

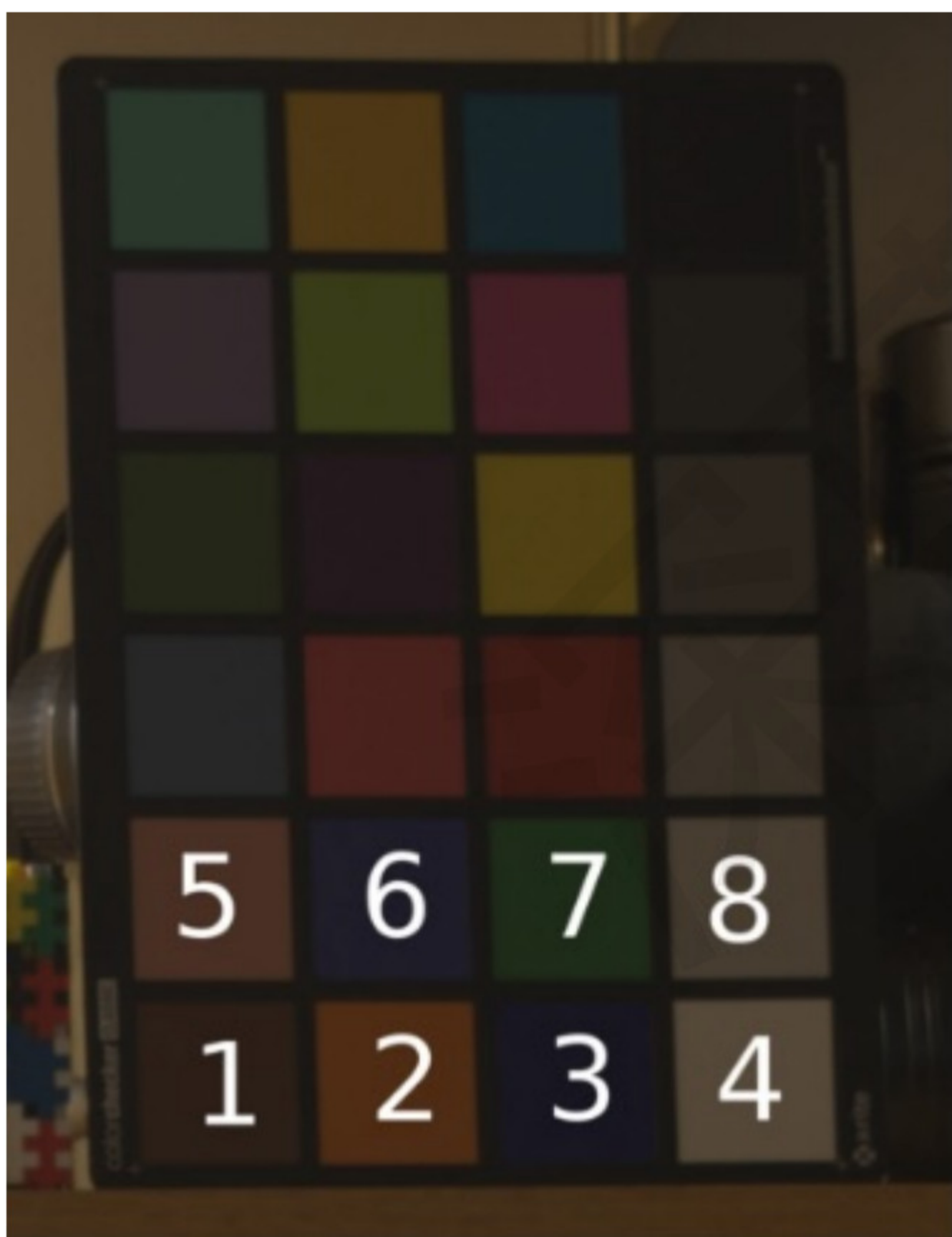


图3：色卡及其编号

特别要注意的是色卡是这样创建的，即当在标准光源（所谓的“D65”光源，大致对应于中午的日光）下观看色彩检查器时，其补丁在（线性）sRGB颜色空间中具有一组特定的RGB坐标。homework ZIP

archive 的 `./src/cp_hw2.py` 文件中提供的 `read_colorchecker_gm` 函数返回这些基准RGB坐标。

然后，为了使你的HDR图像正确显示颜色，你可以对其三个通道应用线性变换，以使图像中的色卡的RGB坐标尽可能接近基准坐标。你可以按照以下步骤执行此操作。

1. 对于每个色卡块，裁剪一个完全包含在补丁内的正方形。（参见 `matplotlib` 函数 `matplotlib.pyplot.ginput` 以交互式记录图像坐标。）确存储这些裁剪正方形的坐标，以便你可以重新使用它们。使用结果的24个裁剪来计算色卡的24个补丁的平均RGB坐标。
2. 通过将1作为它们的第四个坐标来将这些计算的RGB坐标转换为 齐次 4×1 坐标。
3. 解决一个最小二乘问题来计算一个 仿射变换，将测量的齐次坐标映射到基准齐次坐标。
4. 将计算出的仿射变换应用于你的原始RGB HDR图像。请注意，变换后的图像可能有一些负值，你应该将其剪裁为0。
5. 最后，应用一个额外的白平衡变换（即，用标量乘以每个通道），使得补丁4的RGB坐标彼此相等。这类似于手动白平衡，现在我们使用补丁4作为场景中的白色对象。这一步很像我们在第1季课程《从光线到图像-相机成像的原理》中讨论的手动白平衡方法，你可以参考第1季的课件和作业了解更多详情。

将颜色校正和白平衡后的HDR图像存储在一个.HDR文件中。你现在应该总共有两个HDR图像：第1部分的未经颜色校正的图像，以及你刚刚创建的图像。比较颜色校正的图像和原始图像，并讨论你最喜欢哪一个。

3 摄影色调映射（20分）

现在你已经有了几张HDR图像，你需要对它们进行色调映射以便可以显示它们。对于这一部分，你可以使用你在第2部分结尾处最喜欢的两张HDR图像中的任何一张。

你将实现由Reinhard等人[4]提出的色调映射操作符，这是从显示HDR图像开始时一个良好的基础算法。我们在下面描述了如何做到这一点，但强烈建议你至少阅读该文献的第2节和第3节，这些节解释了这个色调映射操作符的特定形式背后的基本原理、各种参数的效果，以及它与用于胶片冲印时使用的分区系统之间的关系。

给定线性HDR图像的像素值 $I_{ij,\text{HDR}}$ ，按照下列方式执行摄影色调映射：

$$I_{ij,\text{TM}} = \frac{\tilde{I}_{ij,\text{HDR}} \left(1 + \frac{\tilde{I}_{ij,\text{HDR}}}{\tilde{I}_{\text{white}}^2} \right)}{1 + \tilde{I}_{ij,\text{HDR}}}, \quad (8)$$

其中

$$\tilde{I}_{\text{white}} = B \cdot \max_{i,j} \left(\tilde{I}_{ij,\text{HDR}} \right), \quad (9)$$

$$\tilde{I}_{ij,\text{HDR}} = \frac{K}{I_{m,\text{HDR}}} I_{ij,\text{HDR}},$$

(10)

$$I_{m,\text{HDR}} = \exp \left(\frac{1}{N} \sum_{i,j} \log (I_{ij,\text{HDR}} + \epsilon) \right). \quad (11)$$

参数 K 是_键 (key) , 决定了色调映射呈现的明亮或暗淡程度。参数 B 是_燃烧 (burn) , 可用于抑制结果的对比度。最后, N 是像素数, ϵ 是一个小常数, 用于避免对数函数在0处的奇异性。

注意: 方程 (11) 与Reinhard等人[4]的对应方程 (1) 不同。这里给出的版本是正确的, 而论文中的版本是错误的。

以两种方式实现摄影操作符并将其应用于你的RGB HDR图像:

- 首先, 通过以相同的方式同时对所有颜色通道进行色调映射来应用它。
- 其次, 仅将其应用于亮度通道Y。

对于后者, 你可以使用作业附件提供的函数1RGB2XYZ将HDR图像从RGB转换为XYZ, 然后使用课堂上讨论的定义将其转换为xyY。在xyY中, 对亮度Y进行色调映射, 同时保持色度通道x, y不变。然后, 使用提供的函数XYZ21RGB反转颜色变换以返回到RGB。

尝试不同的K和B值。参数的一些合理起始值是 $K = 0.15$ 和 $B = 0.95$, 但要获得好的色调映射, 你需要探索不同的值。绘制RGB和亮度方法的代表性色调映射, 并讨论你的结果。清晰描述你最喜欢的色调映射, 以及为什么。

4 创建并对自己的HDR照片进行色调映射 (50分)

现在是时候将你上面实现的应用到你自己的照片中了。为了创建出比任何单一曝光都更好的结果, 你应该拍摄实际具有高动态范围的场景!

好的例子包括: 既有室内又有室外元素的场景 (带窗户的房间)、具有两种不同照明的室内场景 (如你在第1-3部分中使用的数据)、具有非常强烈逆光的场景, 或者在阳光强烈的阴影下的户外场景。请注意, 你必须对在此部分和第5部分中捕获的数据使用相同的相机设置; 请确保阅读“提示和信息”中关于使用什么相机设置的建议。

选择场景后, 以RAW和JPEG格式捕获曝光堆栈。我们建议使用在**对数域**中等间隔的曝光。例如, 从一些非常低的基础曝光开始, 然后使用等于基础的 $2\times$ 、 $4\times$ 、 $8\times$ 等的曝光。你可以穷尽曝光范围 (即从可能的最低快门速度开始, 然后以 $2\times$ 的步骤一直到最大快门速度), 或选择适合你的场景的曝光范围。

使用你捕获的曝光堆栈创建两个HDR图像，一个来自RAW图像，另一个来自JPEG图像。你可以使用你在第1部分中实现的任何HDR变体，无论你喜欢哪个。或者你可以尝试所有的，并决定哪个看起来最好。将这两个图像以.HDR格式存储。由于你没有色卡，你可以跳过色彩校准步骤。

然后，使用你在第3部分（摄影，RGB或仅亮度）中实现的色调映射算法处理这些图像。尝试不同的参数，显示一些代表性的色调映射，讨论你的结果，并确定你最喜欢哪个结果。

你将获得的这部分的总分数将取决于你创建的最终色调映射图像的视觉吸引力。

5 噪声校准和最佳权重 (50分)

注意：这一部分开始，我们要用到第2季第二课《图像噪声深入分析与处理》的知识，请在学习完第2课的视频和课件后开始这部分作业练习。

作为你HDR流水线的最后一步，你将尝试通过实施简单的噪声校准程序来进一步提高你的HDR合成图的保真度。我们将按照我们在课堂上讨论的噪声模型进行，但我们将假设暗电流项为零，可以忽略。

要开始噪声校准程序，你应该打印出一个亮度渐变图像，如图4所示。你可以使用numpy函数 `numpy.tile(numpy.linspace(0, 1, 255), (255, 1))` 生成这样的图像。你应该根据需要重新缩放图案，以便在打印后填满整页打印纸。



图4：噪声校准的渐变图像

5.1 噪声校准 (30 分)

在本节中，你应该使用单一的快门速度（使整个斜坡图像都能得到良好的曝光），并使用与第4部分相同的ISO和光圈设置。

首先，**带上镜头盖**拍摄约50张RAW图像。将这些RAW图像转换为_线性_ 16位.TIFF图像，如你在第1部分所做的那样。然后，将这些图像进行算术平均来得到黑帧。请确保保存黑帧，因为你很快就会用到

它。

现在取下镜头盖，拍摄斜坡打印图像的约 $N = 50$ 张RAW图像。将这些RAW图像转换为 **线性 16 位.TIFF图像**，如你在第1部分所做的那样。然后，从每个图像中减去上面计算的黑帧。对于剩下的噪声校准过程，你只能使用减去黑帧后的图像。

对于一些像素，绘制它们在你拍摄的各种图像中的值的直方图。讨论这些直方图大致具有什么形状，以及为什么。

然后，对于每个像素，计算其均值和方差，

$$\mu_{ij} = \frac{1}{N} \sum_n I_{ij}^n, \quad (12)$$

$$\sigma_{ij}^2 = \frac{1}{N-1} \sum_n (I_{ij}^n - \mu_{ij})^2. \quad (13)$$

将均值四舍五入到最接近的整数，这将导致多个像素具有相同的均值。为这个均值计算平均方差。

正如我们在课堂上讨论的，方差与均值的关系如下：

$$\sigma_{ij}^2 = \mu_{ij}g + \underbrace{\sigma_{\text{read}}^2 g^2 + \sigma_{\text{ADC}}^2}_{\sigma_{\text{additive}}^2}, \quad (14)$$

其中 g 是相机增益， σ_{read}^2 是读取噪声的方差， σ_{ADC}^2 是ADC噪声的方差。因此，如果你将上面计算的方差作为不同唯一均值的函数绘制出来，结果应该是一条近似直线。将直线拟合到你的唯一均值-方差点上，并用它来估算相机增益 g 和总附加噪声方差 $\sigma_{\text{additive}}^2$

5.2 使用最优权重合并 (20 分)

使用你在第4部分捕获的RAW曝光堆栈形成最后一个HDR图像，这次使用我们在课堂上讨论的噪声最优权重方案。

首先，执行黑帧减法，考虑到快门速度的差异：假设你通过使用快门速度 t_{nc} 执行噪声校准来计算黑帧 I_{dark} 。从你的曝光堆栈中的每个图像 I^k 中减去帧 $\frac{t^k}{t_{\text{nc}}} I_{\text{dark}}$ 。

然后，使用以下权重合并黑帧校正的曝光堆栈：

$$w_{\text{optimal}}(z, t^k) = \begin{cases} \frac{(t^k)^2}{gz + \sigma_{\text{additive}}^2}, & \text{if } Z_{\min} \leq z \leq Z_{\max} \\ 0, & \text{otherwise} \end{cases}. \quad (15)$$

将结果图像（经过色调映射后）与你在第4部分获得的最佳结果进行比较。噪声校准在你的图像的哪些部分产生了最大的不同？

6 额外任务：通过变化快门速度和ISO实现HDR (100 分)

正如我们在课堂上讨论的，我们可以通过变化快门速度，ISO或者两者都变化来创建曝光组合以组成HDR图像。当同时变化快门速度和ISO时，需要回答两个问题：

- 首先，我应该如何决定使用哪些快门速度和ISO的组合？
- 其次，我应该如何合并所得的曝光堆栈以创建HDR图像？

在回答这两个问题时，重要的是要考虑到这两种控制曝光的机制的不同噪声特性。

Hasinoff等人[2]提供了关于这种混合曝光法的详细分析。阅读这篇论文并尝试复现他们的算法，以捕捉并合并一个你同时变化快门速度和ISO的曝光组合。要获得全部学分，你需要捕捉同一场景的两个RAW曝光堆栈：一个只变化快门速度，另一个同时变化ISO和快门速度。然后，你应该使用论文中4.1节描述的过程，将这两个堆栈中的每一个合并成HDR图像，该过程对应于第5部分中的噪声最优权重。最后，你需要比较结果。为了公平比较，两个堆栈的总捕捉时间应该（大致）相同。你不需要实现4.2节：你可以使用论文中报告的ISO和快门速度，或者你自己想出的参数。

注意，为了在变化ISO的情况下实现黑帧减法和噪声最优混合，你需要单独估计等式(14)中的项 σ_{read}^2 和 σ_{ADC}^2 。我们之前在 $\sigma_{\text{additive}}^2$ 中合并了这些项，但是当我们使用ISO来变化增益 g 时，这已经不再足够了。为了获得全部学分，你需要想出并实现一个修改过的噪声校准程序，以便估计这两个项。但是，你可以通过合理的猜测（在你的报告中应该证明）来近似，或者在线搜索它们来获得部分学分

交付物

在提交你的作业解答时，请首先遵循深蓝学院的整体提交说明，你首先应该参考我们提供的基础说明信息，其结构如下，你应该基于此基础信息中的内容构建你的作业解决方案。

```
session2-assign1-HDR
├─ assign1-HDR-CN.pdf
├─ data
└─ src
```

你应该打包成一个ZIP文件提交，在你的ZIP文件中，你应该包括以下内容：

session2-assign1-HDR-solution

├─ Report.pdf
├─ data
└─ src

这里有：

- 一个PDF报告，解释你在每个问题中做了什么，包括对第1-5部分中提出的所有问题的答案，以及你选择做的任何选做问题。报告应包括你认为可能有帮助的任何图表和中间结果。确保包括任何可能阻止你完全解决作业的问题的解释，因为这将帮助我们确定部分学分。报告还应解释你在解决方案中包括的任何额外的图像文件（见下文）。
- 你的所有Python代码，包括选做问题的代码，以及一个解释如何使用代码的README文件。
- 你在第1部分（只是你在最后选择的那个）、2、4和5部分创建的HDR图像，以及第4部分中至少一个RAW和相应的JPG LDR图像。如果你认为它们显示了一些重要的内容，你也可以包括其他图像文件，LDR或HDR，用于各种实验（例如，使用不同的值进行色调映射）而不是你的最终实验。
- 如果你做了选做第6部分：在你的PDF报告中包括对你实现的Hasinoff等人的[2]部分的详细描述，你遇到的任何问题，以及在复制他们的算法时做出的任何近似或其他决定。此外，包括你创建的两个HDR图像，以及你为这个部分捕获的至少一个RAW图像。

提示和信息

- 确保下载并安装最新版本的dcraw。特别是，较旧的Windows版本中的默认版本不支持本课程中使用的相机，并且会产生强烈的紫色调。
- 当使用提供的和捕获的曝光堆栈时，你会注意到你的算法会使用**大量的内存**。这是处理现代相机捕获的照片时的常见问题，由于这些相机拥有非常大的像素数量。在2400万像素时，本作业使用的Nikon D3300处于百万像素的中等范围。然而，在这个分辨率下，一个3通道的HDR图像占用了超过0.5 GB的内存。这有两个含义。首先，你应该小心你在Python代码中创建了多少这些图像，否则你有可能会填满你的内存并使你的计算机陷入困境。其次，当处理图像时，你需要确保你使用可以并行处理所有像素的向量化代码，因为尝试使用双重循环一个接一个地处理所有2500万像素将花费很长时间。
- 特别是，在执行HDR合并时，请注意，方程（5） - （6）可以独立应用于每个 k 曝光图像。因此，你可以一次加载和处理一个图像，而不是一次加载整个曝光堆栈。另外，在每个图像中，方程（5） - （6）完全并行应用于每个像素。因此，你可以用一个向量化调用处理每个图像，而不是一个双重循环。当解决线性系统（3）以恢复非线性映射 g 时，无论你多么小心，你都会耗尽内存。按照Debevec和Malik [1]的建议，你应该在形成线性系统之前大大降低输入图像的采样率。请注意，你不应该使用skimage.transform.resize调整图像的大小，也不应该在降采样前尝试模糊它。为了推断 g ，你所要做的就是使用 $I[:, :N, ::N]$ 降采样输入图像 I ，对于某些 N 。我们建议使用 $N = 200$

。更一般地说，当你还在调试代码时，我们强烈建议你在降采样的图像上工作以加速开发过程。一旦你知道你的代码是正确的，你可以再次在完整分辨率的图像上运行它，以产生你的最终结果。

- 当将许多LDR图像合并为HDR图像时，你可能会遇到没有任何良好曝光值的像素（即，方程（5）-（6）的分母中的权重之和正好为0）。你可以将这些像素设置为你的HDR图像的最大或最小有效像素值，分别对于始终过曝或始终欠曝的问题像素。
- 即使进行了色调映射，你的图像可能还是显得太暗。实际上，在色调映射后，你仍然需要应用伽马编码以正确显示图像。其中，伽马编码是以下非线性运算符：

$$C_{\text{non-linear}} = \begin{cases} 12.92 \cdot C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1 + 0.055) \cdot C_{\text{linear}}^{\frac{1}{2.4}} - 0.055, & C_{\text{linear}} > 0.0031308 \end{cases}$$

(16)

你应该在脚本中实现这个，并在显示它们之前始终将伽马编码应用于色调映射或HDR图像。应用伽马编码也将有助于显示中间结果（见下文），你可以参考第1季课程《从光线到图像-相机成像的原理》中第4课课件和大作业来了解其中详情，亦可从互联网资源（如知乎专栏-《计算摄影学》）获得详细解释。

- 你可能会发现显示中间结果很有帮助。如果你直接显示创建的HDR图像，它们可能会显得非常亮（可能完全白色）或非常暗（可能完全黑色）。这不是问题：正如我们在课堂上讨论的，HDR图像与入射流量呈线性关系，但是由（有些）任意的缩放因子缩放。你所要做的就是将你的图像乘以你自己的适当缩放因子（如果图像非常亮，则小于1，否则大于1），应用伽马编码，然后将图像的值域限制在0-1之间然后使用imshow函数。你可能需要尝试一些不同的缩放因子值，直到找到正确显示图像的缩放因子。
- 如果你想查看创建的.HDR文件，请注意你_不能_使用标准图像查看器来执行此操作。相反，你应该使用专用于.HDR文件的查看器，例如OpenHDR。此查看器提供了用于控制曝光（你应用于图像的缩放因子）和伽马编码的交互式滑块，使查看HDR图像的好设置变得更容易。或者，你可以使用我们提供的代码中的readHDR函数打开并加载.HDR文件，然后按照前面的步骤显示它。
- 当单独对每个RGB通道应用摄影色调映射时，你可能会发现使用所有三个通道的相同标量 $I_{m,\text{HDR}}$ 和 $\tilde{I}_{\text{white}}$ 会得到更好的结果。你可以通过在方程（9）和（11）中使用所有三个通道的像素来实现这一点。此外，按照书面方程（11）进行评估（即，首先计算对数的平均值，然后取指数）可能会由于有限的数字精度而导致零，NaN或Inf值。你可能会通过认识到方程（11）相当于计算所有像素 $I_{i,j,\text{HDR}}$ 的几何平均数，并相应地更改你的实现，从而获得更稳定的结果。有关这种数值问题的更多信息，请在互联网上查询“几何平均数的对数平均形式”。
- 当使用你的相机捕捉自己的曝光堆栈和噪声校准数据时，你应确保将其白平衡选项设置为固定预设，以适应你选择的场景的照明，并将其输出颜色空间设置为sRGB。此外，如果你的相机支持，将其设置为为捕获的每个图像存储RAW和.JPG文件（Nikon D3300具有此选项）。这样，你将拥有完美配对的RAW和.JPG曝光堆栈，并且可以使用它们来比较使用其中一个或另一个进行HDR。
- 在捕获曝光堆栈和噪声校准数据时，除快门速度外，其他相机参数不得更改至关重要。因此，你应将相机设置为手动模式，并在捕捉期间禁用自动对焦。如果你不采取这些步骤，那么相机可能会自

动更改诸如光圈，ISO和对焦等参数，从而使你的数据无法使用。关于光圈，你应使用为你的曝光堆栈选择的场景提供良好景深的光圈设置。关于ISO，你应使用低ISO设置，因为这使噪声校准更可靠。关于对焦，你可以在确定你将用于曝光堆栈的场景时使用自动对焦，以确保你捕获的图像将是锐利的。一旦镜头聚焦，你可以禁用自动对焦，切换到手动，然后开始捕获曝光堆栈。你可以使用相同的程序在捕获噪声校准数据之前对焦镜头。如果你需要在捕获曝光堆栈和噪声校准数据之间更改镜头焦点，这是可以的；但是，你应确保除了对焦和快门速度（ISO，光圈，白平衡）之外，其他所有设置都相同。最后，当捕获噪声校准数据时，你应使用低于自动曝光为校准目标选择的快门速度，即使你的图像比理想的要暗一些也没关系。

- 正如我们在课堂和上述作业中讨论的那样，在捕获曝光堆栈时保持相机和场景静止非常重要。鉴于此，我们强烈建议你将相机安装在三脚架上，或者至少在非常稳定的表面上（例如，桌子）上拍摄图像。在捕获曝光堆栈时，你将需要多次调整相机的快门速度。手动执行此操作需要触摸相机以旋转快门速度拨盘。你还需要激活快门，这意味着需要进一步触摸相机并按下按钮。所有这些手动操作都可能导致相机移动很大，从而导致你捕获的LDR图像未对齐。使用三脚架不能保护你免受这种类型的相机运动。因此，我们强烈建议你连接相机到你的笔记本电脑，以便你可以电子控制其设置和快门释放，而无需触摸相机。很多单反或数码相机提供了这样的连接控制方式。要控制相机，你可以尝试使用相机制造商在其网站上提供的软件。作为替代方案，我们建议你尝试使用gphoto2。这是一个非常强大的命令行工具，可用于连接相机并实现非常复杂的捕获程序。例如，以下行将自动检测连接的相机，以快门速度1/2048捕获图像，然后从相机下载图像到你的计算机，并将其存储为文件名exposure1。如果你的相机设置为同时捕获RAW和.JPG，则此摘录将下载两个图像，并将其分别存储为exposure1.nef和exposure.jpg，尝试查询资料，看看你的相机是否倍支持。

```
gphoto2 --auto-detect gphoto2 --set-config-value /main/capturesettings/shutterspeed=1/2048
```

```
gphoto2 --capture-image-and-download --filename exposure1.%C
```

- 如果你正在使用尼康D3500相机，请注意gphoto2有一个已知问题，导致相机在几个捕获命令后失败。如果你遇到此问题，你可以通过终止gphoto2进程（按“Ctrl + C”两次）并运行gphoto2 --reset来解决它。有关更多详细信息，请参阅相应的问题页面。
- 在处理噪声校准数据时，删除高均值可能会有所帮助。在高光通量区域，泊松噪声变得占主导地位，高均值具有很高的方差，可能会产生强烈的异常值。这些异常值可能会使你拟合的线倾向于具有负截距。如果发生这种情况，你可以将用于拟合的均值范围减少到较低的均值，这些均值对于估算加性噪声更为可靠。
- 在执行噪声校准时，应确保你的场景中没有任何荧光灯。这些灯的光输出随时间变化，尽管频率非常高，我们无法感知。这种时间变化可能会使你的噪声校准结果无效。
- 在使用噪声校准结果进行最佳权重合并时，应确保调整增益和方差值，以适当考虑到在作业的不同部分中如何规范化图像数据的差异。此外，由于白平衡，每个颜色通道的噪声校准结果将有所不同，因此你应确保为每个通道使用正确的值。

致谢

该作业改编自CMU 2021年计算摄影学公开课程15-463的作业2，感谢CMU，以及授课老师Ioannis (Yannis) Gkioulekas的开放精神。

另外，源作业说明中提到下面的信息，在此也一并感谢：

本次作业和撰写报告的一些灵感来自于布朗大学的James Hays和西北大学的Oliver Cossairt的计算摄影课程。读取真值颜色检查器RGB值的代码来自于Ivo Ihrke的颜色校准工具箱。

参考资料

- [1] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369-378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [2] S. W. Hasinoff, F. Durand, and W. T. Freeman. Noise-optimal capture for high dynamic range photography. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 553-560. IEEE, 2010.
- [3] K. Kirk and H. J. Andersen. Noise characterization of weighting schemes for combination of multiple exposures. In *BMVC*, volume 3, pages 1129-1138, 2006.
- [4] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 267-276, New York, NY, USA, 2002. ACM.