

此文档为教案，统筹教学安排，包括讲稿预演和课程说明，点击以下目录可跳转各部分安排，教案与PPT对应

- I.引入
 - 注明
 - 介绍
 - 课程Outline
- II.HDR介绍
 - 动态范围Dynamic Range
 - HDR存储
 - HDR合成
 - HDR显示
- III.HDR合成
 - 获得HDR图像的方式
 - 摄像学-了解
 - 数字图像处理学-多重曝光融合
 - 多重曝光融合技术讲解
 - Step1 多重曝光
 - 介绍-曝光控制
 - 介绍-拍摄技巧
 - Step2 曝光融合
 - 辐照度和响应曲线
 - 相机内部处理流程
 - 线性图像融合(raw)
 - 非线性图像融合
 - 具体实现-Debevec算法
 - Step1 相机响应曲线(CRF)标定
 - Step2 响应曲线反映射回线性空间
 - 官方函数
 - MATLAB
 - Python
 - 了解-深度学习实现HDR合成
- IV.HDR显示
 - 重点-Tone Mapping
 - 介绍与分类
 - 全局映射
 - 线性映射
 - Reinhard算法(初步)
 - 对数映射

- CryEngine2算法
- Flimc算法
- ACES算法
- Reinhard算法
 - 从色彩到亮度
 - 步骤
- 局部映射
 - 基于双边滤波Bilateral Filtering
 - 基于梯度
- 官方函数
 - MATLAB函数
 - Python函数
 - Reinhard
 - Dragon-双边滤波
 - Mantiuk
- 了解-Multi-Exposure Fusion
 - 介绍
 - 官方函数
 - MATLAB
 - Python
- V.课堂总结
 - 内容总结重述
 - Homework提供
 - 资料提供

I.引入

注明

- 我的教学实践以HDR为主题，把我所要讲的知识分为需要听众理解、掌握、了解的三部分：
 - **理解**：重要概念、必备知识，达到听众理解技术本源的目的
 - **掌握**：经典合成与映射算法(会着重给出代码和"作业")，达到听众可以实现MATLAB/Python复刻的效果
 - **了解**：拓展延伸方面，如exposure fusion技术、深度学习实现HDR合成等，仅作为介绍
- 注意：每一部分都要凝聚在**问题/概念本质**，**需要让听者思路清晰**。
- **可行拓展/教学建议**：如果作为真实课程，对于ToneMapping和Expsoure Fusion的各种算法，可以布置补充代码的任务。

- 比如，给出未完成的Python代码(教辅)，请同学来补足其中的ACES,Reinhard等函数，作为技术实现体验。
- 为保证教学实践的完整性，我也设计了一份**HDR相关的完整作业**。

介绍

- **前设**：这个教学实践，是在假设听众并未了解过相关知识的基础上制作的。
- **背景**：HDR的处理在HW5中也有布置，叶高翔和汪宇翔同学做的相关分享都内容丰富、引人入胜，各有所长。
- **原因**：因为呢，我在选这门课之前是真正的“零基础”，真正自己实现过的图像处理相关技术也仅仅局限于几次作业。那限于能力确实有限，我决定在原有的HW5的HDR技术实现上进行扩充。
- **目标**：如果能够达到把HDR向一位从未了解过的人清晰地传输的效果，对我而言，我的这份教学就算成功，这门课也就有所收获了。
- **作业**：保证课程完整性，设计**HDR相关作业**，作业分为HDR合成和色调映射两部分：**一方面**要求深入了解尝试OpenCV和MATLAB的相关函数；**另一方面**要求实现简单传统算法。

课程Outline

1. **HDR介绍**：概念了解，动态范围、HDR由来与特性、HDR存储编码等
2. **HDR合成**：HDR合成(raw与LDR)，Debevec算法,官方函数，拓展
3. **HDR显示**：从HDR到LDR的色调映射：整体/局部映射方法，其他方法，官方函数
4. (仅了解)**HDR成像**：合成+显示过程的融合，Exposure Fusion
5. **总结**：
 - 实操：MATLAB官方函数总结
 - 课堂脉络回顾
 - 布置作业，提供教辅资料作为可行答案(可运行代码)
 - **理解、掌握、了解**要求重述，提醒尝试复刻ToneMapping代码，**再次强调**重点算法实现。

II.HDR介绍

今天我主要为大家介绍HDR的概念以及相关的简单技术-合成与显示。那首先，来到我们的第一个问题——HDR是什么？

动态范围Dynamic Range

- **Dyanmic Range**= $\log_{10}(\text{Max Intensity} / \text{Min Intensity})$

- 人类视觉系统可以适应非常宽广的亮度变化范围(5000:1)
- 真实世界动态范围极高(100000:1)
- raw图像、LDR(**Low Dynamic Range**)图像:根据曝光设定不同, 图像覆盖的动态范围会有所变化 (300:1-1000:1)
 - 根据0-255纯黑与纯白亮度计算, LDR动态范围最高也只有**50**左右
 - 无论是在高亮度情况下还是在低亮度情况下, 数码相机成像的动态范围都只是典型自然场景的动态范围的很小一部分
- 引入**HDR**: High Dynamic Range高动态范围图像,拥有更多细节信息

HDR存储

- 一般的低动态范围图像为8bit存储, 对于HDR, 高动态范围图像, 我们要有新的存储格式。
- **HDRI**(High-Dynamic Range Image)就是记录采用了HDR技术的图象数据文件
- 常用的HDRI文件有以下三种格式:
 - **OpenEXR**:.exr,R,G,B,A四通道, 每通道FP16(16bit)
 - **RadianceRGBE**:.hdr,R,G,B,R四通道, 每通道8bit
 - **FloatTIFF**:.tif,3通道, 每通道FP32
- 浮点格式存储:
 - FP8: 8位浮点格式
 - FP32:32位浮点格式
 - FP16:16位浮点格式
- RGBE与FP32(R,G,B通道)的转换(介绍)

HDR合成

对于HDR, 传统技术主要围绕合成与显示。HDR合成, 指的是利用每张不同曝光时间LDR (Low Dynamic Range 图像覆盖的动态范围会有所变化, 低动态范围) 图像中的最佳细节, 可以合成HDR图像; 相比普通图像, HDR图像提供了更宽广的动态范围和更丰富的图像细节。

HDR显示

由于显示器的动态范围有所限制, 可能不能完全涵盖HDR的动态范围。HDR显示, 指的是将HDR图像呈现在显示器的过程。

- 如果显示器的动态范围大于HDR图像, 那么我们可以直接显示
- 但如果显示器的动态范围小于HDR图像, 我们就需要通过色调映射(也就是Tone Mapping)对HDR图像进行压缩, 将HDR图像的动态范围映射变换到普通显示器的动态范围之内。

HDR的成像过程, 指的就是我们先从LDR图像中获得HDR图像, 再映射到显示器成像, 显示出较好的图像品质的过程。今天我们主要介绍HDR合成与显示的基本知识。

III.HDR合成

先来看HDR的合成。

获得HDR图像的方式

摄像学-了解

从前面我们可以知道，HDR实际是从摄像学中诞生的概念。当前的摄像已经有许多获得HDR图像的方式，如：

- raw:对于现代传感器来说
 - 单张RAW格式的照片所存储的不同明暗的信息已经足够多，他们远高于普通显示器所能显示的亮度范围，完全可以应付一般场合下的亮暗差别
- 当前有些手机也已经提供了HDR模式图像的拍摄
 - e.g.: SDR+ Gain Map(增益图) 扩展为HDR
- 市面上也有一些常见的HDR合成工具：
 - Lightroom 和 Photoshop 的 HDR Pro 等

数字图像处理学-多重曝光融合

从数字图像处理技术角度而言，我们主要讲解多重曝光融合技术，编写算法来实现HDR合成。

多重曝光融合技术讲解

多重曝光融合技术，分为两步——多重曝光和曝光融合。（分别进行什么操作）

Step1 多重曝光

首先来看多重曝光，我们需要拍摄多张不同曝光的LDR图像。

介绍-曝光控制

相机的曝光由快门、光圈、感光度(ISO)三要素控制

介绍-拍摄技巧

我们的多重曝光图像的拍摄，一般要控制：

- **光圈固定**（否则景深会有差异）
- **对焦点固定**（否则对焦点的前后移动，会导致画面有轻微的呼吸效应，此时建议手动对焦）
- **焦段固定**（要不然画面内容都不一样了）

- **ISO固定**（曝光合成很大程度上是为了画质，如果ISO随意浮动，那可能就让你的辛苦打水漂了）
- **只调整快门**
 - 不同曝光时间的差异越大，能覆盖的动态范围就越大
 - **曝光次数和曝光时长**的选取取决于场景的动态范围
 - 一般选取5个不同的曝光时间、相邻曝光间隔两档
- 考虑**运动**物体对成像的影响，特别是在有风的时候，**水、灌木和云**，因为较为柔和缓慢经常会被忽略；
- 不一定需要很多张，适度几张就好。**确保最高曝光的，能够把场景中的阴暗部分细节拍到；而最低曝光的能够保留场景中亮部的细节**
- 建议用**三脚架**拍摄，且**短时间内**完成。
 - 对于相机的移动，现代合成软件大部分都提供了**对齐**的功能
 - 不过也有很多时候，对齐并不能达到我们预期的效果
 - 如果对齐出现问题，且画面中有一些中等大小、且边缘非常不工整的物体，比如树丛、栅栏，后期就会出现大量的重影

Step2 曝光融合

在拍摄完一组LDR图像后，接下来我们需要将它们进行融合。

辐照度和响应曲线

在介绍融合原理前，我们要先明确两个概念。辐照度和相机响应函数(Camera Response Function)介绍

- 响应曲线的本质可以认为是**现实亮度到图像亮度的映射曲线**
- 而我们进行的**HDR合成**，实际上就是根据图像的像素值，通过相应曲线获得辐照度，将现实的辐照度作为HDR图像的像素值的过程。只不过响应曲线的映射分为线性和非线性。

相机内部处理流程

ISP(Image Signal Processor),相机的图像处理器会执行一系列将原始图像转换为“正常”图像的一系列图像处理操作。对于原始的raw图像，辐照度和亮度之间的相机响应函数为线性映射；而经过ISP处理后得到的图像，其相应函数是非线性的。我们从满足线性映射的raw图像开始说起。

线性图像融合(raw)

- 对于raw图像，辐射度 L 与图像上的像素值满足线性映射，这个线性关系会受到噪声影响。我们可以用这个公式来表示 $I(x,y)=\text{clip}(t_i*L(x,y)+\text{noise})$.
 - 这里假设是通过改变曝光时间从而改变了图像的像素值，其中 L 是场景的辐照度， t_i 是第 i 帧的曝光时间。
- 我们图像融合的目的就是要通过已知的 $I(x,y)$ 来恢复 $L(x,y)$

- 对同一场景，有点地方亮度过高曝光过度，变成了纯白色，而有的地方则又亮度过低曝光不足，出现了很多噪声；也就是说，对于过曝/欠曝的部分，我们认为其不符合线性原则，即不符合 $I=t*L$ 的原则。因此我们要把合成时要用的像素值 I 限制在一定范围内，保证有效性。
- 因此，典型的步骤分以下三步：
 - 找到该像素**正常曝光**的图像(比如我们限制在0.05-0.95之间，控制噪声和饱和的影响)
 - 选择合适的权值为不同图像加权，在这里我们使用最简单的 I/t 来作为权重
 - 加权求和得到最终的图像

非线性图像融合

以上是对于raw图像，辐照度与像素值满足线性关系的情形下，**想要通过 I 来恢复 L** ，只需要加权求和就可以；但是更多情况下，我们得到的是相机ISP处理后的呈现非线性相机响应曲线的图像，对于这些非线性图像，想要合成HDR，**依然从像素值 I 恢复辐照度 L** ，就需要先进行线性化的过程。

- **线性化操作**：也就是说，我们知道，对于这些图像，辐照度 L 和像素值 I 的相机响应曲线是如图非线性的，如果想要从图像的像素值 I 恢复现实的辐照度 L ，我们就需要获得逆响应曲线，从 I 恢复 L ，这就是对图像**线性化处理**的过程
- **融合步骤**：非线性图像的融合相比ra图像，实际上就是多了前两步：**我们需要首先标定图像的相机响应曲线，从而能够从图像像素值恢复现实的辐照值 L ，实现线性化过程**；之后再按照raw图像相同的处理步骤，进行加权平均。

具体实现-Debevec算法

已经确认了理论，接下来我们进行实操。Debevec在其论文中给出了一个完整的方案。我们的目标就是，**恢复相机响应曲线，然后恢复现实的辐照值**。

Step1 相机响应曲线(CRF)标定

- 首先，第一步，我们要进行响应曲线标定，获取从辐照度到像素值的响应曲线 f 或者对应的逆响应曲线
 - 我们先来进行**变量约定**(见PPT，叙述)， $E=Xt$ 是我们推导的根本基础
 - $g=\ln f^{-1}$ 的推导(PPT，叙述)，我们现在的目标就是求得 g
- 再看重新看一下这个问题，现在我们有一个方程组 $g(Z_{ij})=\ln E_i+\ln \Delta t_i$ ，未知量为 g 函数和 E_i ，我们这一部分的**目标是求得 g**
 - **方程个数**： $NP(1 \leq i \leq N; 1 \leq j \leq P)$
 - P 为图像数
 - N 为每张图片所取像素点个数
 - **未知数个数**： $(Z_{\max}-Z_{\min}+1)$ 个 $g(Z)$ 的值和 N 个 $\ln E_i$ 的值
 - 由于 g 是离散的，所以我们求得 $(Z_{\max}-Z_{\min}+1)$ 个 $g(Z)$ 的值后即相当于确定 g
 - 我们假设 Z_{\min} 和 Z_{\max} 是最小和最大像素值(整数),则需要 $(Z_{\max}-Z_{\min}+1)=256$ 个 $g(Z)$ 的值

- 现在我们令 $NP \geq Z_{\max} - Z_{\min} + 1 + N = 256 + N$ ，从而就构成一个**超定方程组**，我们对这个方程组直接用最小二乘法求解：**我们希望求一组解 $g(z), \ln E_i$ ，使得这个方程组左右两端相差最小，拟合程度最高**。也就是：将使得以下目标函数 O 取得最小值的未知数作为最小二乘解(这是最小二乘法的内容，此处不予讲解)
 - 加入 $g''(z)$ 为了平滑性；
 - 加入 $w(z)$: 设置权重
 - 我们来考虑一下，是否照片中的所有像素都同等重要？什么意思呢，考虑照片中有一部分是非常黑或者非常白，它能给予我们的信息其实是非常有限的，当我们观赏一张照片时，我们往往是盯着那些既不太亮又不太暗的部分，那些区域能给予我们更多的信息。从另一角度来说，因为像素值范围是有限的，当它取边界值时，往往不能反映真实世界中的光照，于是我们可以认为这些太黑或太白的像素是不那么重要的。这里需要区分绘画和照相，绘画中，使用纯黑或纯白来增强表现力是常用技巧，而不同于照相。那么就有一个很简单的想法，我们给每个像素加权，让它体现不同的重要性即可。
 - 所以，我们对 **0~255 这 256 个像素值赋以不同的权重**，越靠中间（127）的权重越高，越靠两边（0 或 255）的权重越低。
 - 一个可行的 $w(z)$ 方案如下
- 好，那我们再次重申一下我们的问题：
 - **未知数**: N 个 $\ln E_i$ 和 $g(0), g(1), \dots, g(255)$ (假设 Z_{\min} 和 Z_{\max} 是最小和最大像素值)
 - **求解**: 求出使 O 值最小的 $N+256$ 个未知数
 - 求得解的 $g(0), g(1), g(2), \dots, g(255)$ 即相当于确定 $g(g = \ln f - 1)$
 - **解决方法**: 构造线性方程组 $Ax = b$ ，求使得 $|Ax - b|$ 取最小值的 x (最小二乘法)
- 现在问题的关键在于，构造线性方程组 $Ax = b$ ，使得该线性方程组与 O 对应，即使得 $|Ax - b|^2 = O$ 。我们这样构造
 - 显然， x 作为要求的解，即为我们的未知数 $g(0), g(1), \dots, g(255), \ln E_1, \dots, \ln E_n$ 。
 - O 前半项... 分别对应方程...
 - O 后半项...
 - 也就是说，构造出一个这样的方程 $Ax = b$ (重新与前一页 PPT 的 O 前后半项对应的方程进行讲解对照)，现在用最小二乘法求我们的解 x
- 最小二乘法可以通过正规方程法、奇异值分解法(SVD)求解，推导过程不是我们的重点，直接按照公式求得 x ，从而获得 $g(0)$ 到 $g(255)$ 即可

Step2 响应曲线反映射回线性空间

- 接下来我们进行第二步：响应曲线反映射回线性空间，我们从像素值根据 g 恢复辐照度 E ，对其进行加权平均，最终结果作为 hdr 图像像素
 1. 依然回到我们的方程组： $g(Z_{ij}) = \ln E_i + \ln \Delta t_i$ ，取对数，取指数，求得 E_i 。
 2. 而对于 P 张图像分别求得的 E_i ，我们根据分配的权值 $w(z)$ 进行加权求和。在这里，由于对数加权更符合人眼特性，会更优一些，我们是直接对 E_i 对数进行求和最后再指数化得到 E 的。

- 或许有同学会问为什么前面已经求过 E_i 了，这里还要重新求。是这样的：
 - 我们往往不会将所有的像素点用于构造这个线性方程组，毕竟一般情况下我们只要求**矩阵A的行数大于列数,也就是 $NP > 256 + N$** ,这样就保证存在最小二乘解，而求解的时间复杂度是很高的
 - 如今随随便便一张照片都不止 100 万像素，几张图片的像素总量可能上千万，假如所有像素都用于构造，解这个方程组可能得花上一年半载
 - 因此，实际操作时，我们往往是只选其中**若干个位置的像素**来构造矩阵，这样就减少了的维度，同时也降低了矩阵 A 的行数和列数，降到能在短时间内求出解的程度。
 - 这些选取的像素点可能是随机选的，也可能是有规律的（比如每间隔多少就选一个），甚至可能是人为挑选的，算法的原论文就是手动选的点。事实上，**选点的好坏也会影响相机响应曲线恢复的好坏**，从而影响最终图片的效果。这里的话可能需要具体情况具体设置了。
- **总之，最终，我们得到E，即得到了HDR图像的像素值**，HDR合成算法结束。我后边设计了一份Debvec算法实现的作业，帮助理解。如果大家有不理解的地方，可以再参考PPT这个网站。

官方函数

关于HDR的合成，MATLAB和Python都提供了官方的函数，我再做一个简单的介绍。

MATLAB

`HDR=makehdr(files,Name,Value)` **Name-Value:**

- 'BaseFlie':曝光基底的文件名
- 'ExposureValues' 输入集中每个图像的曝光值，类型为正数向量，向量中第k个值对应输入集中第k个LDR图像。曝光值的增加意味着曝光值加倍，曝光值的减少意味着曝光值减半。
- 'RelativeExposure' 输入集中每个图像的相对曝光值，类型为正数向量，向量中第k个值对应输入集中第k个LDR图像。例如，相对曝光值为0.5的图像的曝光量是曝光值为1的图像的一半。类似地，相对曝光值为3的图像的曝光量是曝光值为1的图像的三倍。
- 'MinimumLimit' 最小曝光值，指定为正整数。对于每个LDR图像，值小于此最小值的像素被认为曝光不足，不会对最终的HDR图像产生影响。默认情况下，此最小值设置为图像数据类型允许的最大强度的2%。
- 'MaximumLimit' 最大曝光值，指定为正整数。对于每个LDR图像，值大于此最小值的像素被认为过曝，不会对最终的HDR图像产生影响。默认情况下，此最小值设置为图像数据类型允许的最大强度的98%。
- 'CameraResponse' 相机响应函数，对于灰度图像，此参数指定为 $n \times 1$ 向量，对于彩色图像，此参数指定为 $n \times 3$ 矩阵，相机相应函数将曝光值的对数值映射到输入图像的强度级别。n的值是输入图像位深的2次幂，例如，如果输入图像集的位深度为8，则n为256。

Python

- Step1:读取不同曝光度的多张图像
- Step2:对齐图像
 - 合成 HDR 图像时使用的图像如果未对齐可能会导致严重的伪影。在下图中，左侧的图像是使用未对齐的图像组成的 HDR 图像，右侧的图像是使用对齐的图像的图像。通过放大图像的一部分（使用红色圆圈显示的）我们会在左侧图像中看到严重的鬼影
 - 即使用三脚架、短时间拍摄，依然有可能有没有对齐的现象，Python提供了一个帮助对齐的函数
- Step3:提取相机响应函数Camera Response Function(CRF)
- Step4:合并图像

了解-深度学习实现HDR合成

那关于HDR的合成，我们今天需要大家掌握的呢，是最简单的传统HDR合成算法的实现；当然，当前的深度学习算法已经实现了多种从单个/多个LDR图像合成HDR的算法，这里提供一些论文，有兴趣的同学可以再深入了解。

IV.HDR显示

完成了HDR合成，接下来要考虑的就是如何将HDR图像在较小动态范围的显示器上成像。前面我们已经介绍，主要依靠色调映射，也就是ToneMapping来实现：**将HDR图像的高动态范围的色调映射到 $[0,1]$ 范围内**

重点-Tone Mapping

介绍与分类

- Tone mapping算法可以分为全局处理方法(Global Tone Mapping)和局部处理方法(Local Tone Mapping)
 - GTM算子对全图所有像素值应用一个统一的映射规律
 - 而LTM算子会在局部根据一定的规律进行映射。
- **全局映射**算法的设计，主要在于设计像素值 L 的映射函数，对所有像素值进行统一映射；
- **局部映射**算法则会通过分段映射、双边滤波等方式进行局部的映射。

全局映射

事实上，作业5报告的两位同学介绍的方法都是全局映射算法——寻找一个函数，把像素值映射过去。我们现在也先来看一些经典的全局映射算法。问题的本质还是：寻找一个函数，进行高动态到 $[0,1]$ 的映射

线性映射

我们首先想到的映射方式肯定是线性映射：直接进行线性放缩。把HDR图像的最大值映射为1。这将得到非常糟糕的图像，此时图像显得非常暗，这是因为人眼对暗光更加敏感。因此线性映射几乎不予考虑使用

Reinhard算法(初步)

所以我们想要找一个非线性映射，比较简单的映射呢，就是Reinhard提出的这个， $l=l/(1+l)$ 。效果会比线性映射好一些，得到的映射曲线也比较贴合人眼的曲线。这个简单的映射，给了我们启发：去寻找映射函数，使得映射函数贴合人眼的这种S形曲线。那这种S形曲线启发了很多新的思路。

对数映射

比如说这个对数映射，这个映射相比前面最简单的Reinhard算法的优势在于，可以提高暗部亮度。

CryEngine2算法

比如说孤岛危机（Crysis）的CryEngine2。反正就是想拟合S形，我们直接用指数去做，`adapted_lum`是一个比较随意的值，可能需要不断地调试来去拟合。说白了，就是凑函数，改进的话就比较玄学。

Flimc算法

到了2010年，Uncharted 2公开了它的tone mapping方法，称为Filmic tone mapping，用一个复杂的多项式来去拟合形曲线。ABCDEF都是多项式的系数，而WHITE是个magic number，表示白色的位置。这个方法开启了tone mapping的新路径，让人们知道了曲线拟合的好处。它的表现是非常优异的。

ACES算法

后来，电影行业又衍生出一种ACES算法，也是多项式进行拟合，但是实现起来比Filmic更简单，效果也是出奇的好。

Reinhard算法

那以上的这些算法呢，都是通过全局映射进行S形曲线拟合，且都是在对RGB通道进行相同的映射。但是由于人眼对不同颜色的敏感程度其实是不同的，这样R,G,B直接色彩映射得到的结果会**更暗**一些。

从色彩到亮度

事实上人眼感知(0,0.7,0)比(0,0,0.7)更亮，因此Reinhard提出，在映射中加入亮度信息。

步骤

- Step1:计算平均亮度

- Step2:计算相对亮度
- Step3:亮度映射到区间[0,1]
- Step4:根据亮度比例, 对颜色空间color进行映射
- 这就是我们希望做的
 - 将图像转换到新的颜色空间, 从而仅仅对亮度通道做色调映射, 颜色通道通过亮度通道的变化比例来做简单压缩。
 - 对于Intensity, 将其从input映射变换为output, 从而color通道(R,G,B)根据input与output的比例进行变换。
- 这样相比R,G,B直接色彩映射, 弥补了一部分变暗的缺陷。

局部映射

这就是我们Reinhard对色彩进行映射的结果 可以看到, 图像变亮了。但是, 对比度较低, 丢失了很多细节。对于**对比度**等细节的恢复, 我们就需要局部映射来实现。

基于双边滤波Bilateral Filtering

- 于是为了更多细节的恢复, 我们可以只对图像亮度通道的低频信息做色调映射, 而对高频信息和颜色通道, 按照低频信息变换的比例来做调整
 - 注意看图像下面的建筑物, 确实更多的细节凸显出来了。但树顶却出现了光晕现象, 这是由于在拆分图像的高频区域和低配区域时所用的低通滤波器 (一般是高斯滤波) 将图像的不同亮度的区域加权混合在一起的原因
- 于是科学家提出了基于双边滤波的局部映射方式。依然分为亮度映射和色彩变换两部分
 - **亮度映射**
 - 对图片进行分层: 主要为亮度变化信息的基础层 (base), 和包含高频纹理的细节层 (detail)
 - 在压缩的时候, 只需要对基础层做对比度压缩, 压缩之后的基础层再和原来的细节层相加, 得到保留了细节信息的低动态范围图像
 - **像素映射**: (Intensity)亮度/强度变化后, 根据亮度的input与output,对颜色通道(R, G, B)进行比例映射
- 具体的**压缩方法**:
 - 原来的 HDR 图像 = $1.0 * \text{基础层} + 1.0 * \text{细节层}$
 - 压缩的 HDR 图像 = $\text{factor} * \text{基础层} + 1.0 * \text{细节层}$
- 步骤如下:
 - 用双边滤波算子与原图做卷积 (调用 Bilateral 函数), 卷积后的得到图像作为**基础层Base**, 原图减去基础层作为**细节层Detail**。
 - 对基础层(Base)做**对比度压缩**. 设定压缩因子 $\text{factor} = \text{contrast} / (\max(\log \text{Base}) - \max(\log \text{Detail}))$ 。

- 压缩之后的**基础层再和原来的细节层相加**, $\log(\text{Output_Intensity}) = \text{factor} * \log(\text{Base}) + \log(\text{Detail})$. 【注意我们这里是对数操作, 效果更好】
- 获得Output_Intensity后, **color=color*Output_Intensity/Input_Intensity**
- 最终的这些结果对比见图。
- 我后边设计了作业来帮助大家实现这些算法。

基于梯度

对于局部映射, 还有一种较为经典的算法, 是基于梯度进行的, 步骤为... 今天不是我们这门课的重点。而两种局部映射的效果, 或许也是见仁见智。

官方函数

下面我们还是来为大家介绍一些MATLAB和OpenCV的官方函数。

MATLAB函数

- **tonemap**,Name-Value:
 - 'AdjustLightness' 渲染图像的整体亮度,指定为双元素向量。向量形式为[low high], 其中低和高是低动态范围图像的亮度值, 在范围 (0,1) 内
 - 'AdjustSaturation' 渲染图像中颜色的饱和度, 指定为正标量。默认为1
- **tonefarbman**:使用称为多尺度色调和细节处理的边缘保持分解的过程,Name-Value:
 - 'RangeCompression' 应用于HDR图像的动态范围压缩量, 取值在[0 1]之间, 默认为0.3。0代表最小压缩, 1代表最大压缩。
 - 'Saturation' 饱和度, 建议取值在[0 5]之间, 默认为1.6
 - 'Exposure' 曝光度, 建议取值在(0 5]之间, 默认为3
- **localtonemap**:在使用色调映射的同时保留局部对比度,Name-Value:
 - 'RangeCompression' 含义同tonemapfarbman函数的'RangeCommpression', 默认为1。
 - 'EnhanceContrast' 局部对比度增强量, 取值在[0 1]之间。默认为0。0代表局部对比度不变, 1代表最大的局部增强对比度
- **介绍结果对比**, 曝光、对比度有所差异

Python函数

参数是通过反复试验获得的。**最后的结果乘以3只是因为它给出了最令人满意的结果**

Reinhard

- **intensity** 参数应在 [-8, 8] 范围内.更高的亮度值会产生更明亮的结果。
- **light_adapt** 控制灯光, 范围为 [0, 1]。
 - 值 1 表示仅基于像素值的自适应
 - 而值 0 表示全局自适应
 - 中间值可以用于两者的加权组合。

- **color_adapt** 控制色彩，范围为 $[0, 1]$
 - 如果值被设置为 1，则通道被独立处理
 - 如果该值被设置为 0，则每个通道的适应级别相同
 - 中间值可以用于两者的加权组合

Dragon-双边滤波

实际就是双边滤波算法 这里， bias 是 $[0, 1]$ 范围内偏差函数的值。从 0.7 到 0.9 的值通常效果较好。默认值是 0.85。

- 该算法基于将图像分解为基础层和细节层。使用称为双边滤波器的边缘保留滤波器来获得基本层。
- sigma_space 和 sigma_color 是双边滤波器的参数，分别控制空间域和彩色域中的平滑量。

Mantiuk

参数 scale 是对比度比例因子。从 0.7 到 0.9 的值通常效果较好

了解-Multi-Exposure Fusion

介绍

- **Muti-Exposure Fusion曝光融合**:对于拍摄得到的LDR图像集，跳过HDR的合成和色调映射(不需要曝光时间提供和相机标定曲线)，直接将多张LDR图像融合为一张可显示的、效果更好的图像
- **思想**:
 - 对于多曝光图像序列，取每一张图像中最有价值的部分用于合成。
 - 需要一个指标来衡量每张图像中哪些像素有价值，然后通过计算每张图每个像素的价值指标当作对应的权重。
 - 最终通过加权融合的方式得到HDR。
- **价值评估(3维)**:
 - **对比度** $C(I_k)$:对图像的灰度图执行**拉普拉斯滤波**，结果取绝对值作为对比度指标
 - **饱和度** $S_{k,ij}$:将RGB三个通道之间的标准差作为饱和度指标
 - **亮度** $E_{k,ij}$:对于归一化至0~1范围的图像，将取值在0.5左右的像素视为曝光良好，应该分配很大的权重；接近0和1的分别为欠曝和过曝应该分配很小的权重。像素值与其对应权重的关系符合均值为0.5的**高斯分布**
- **权重计算**:获取以上3个指标后，就能计算每个像素对应的权重 $W_{i,j,k}$.默认 $w_c=w_e=w_s=1$. 为了保证多张图像在同一位置的权重和为1，需要在图像数量维度上对权重进行归一化
- **融合**:根据计算的权重对原始图像进行加权求和，即可得到融合后的图像

官方函数

MATLAB

- **blendexposure函数**

- blendexposure函数将具有不同曝光值的灰度图或RGB图像根据图像的对比度、饱和度、曝光度进行合理混合，最终返回曝光良好的图像J

- **参数:**

- **I1,I2,...In**: 灰度图像或RGB图像。所有的输入图像都必须具有相同的大小和数据类型
- **J**: 输出的混合图像，图像数据大小和数据类型和输入图像一致
- **Name-Value**:
 - 'Contrast': 对比度的权重，默认为1，可以指定为[0 1]内的数
 - 'Saturation': 饱和度的权重，默认为1，可以指定为[0 1]内的数
 - 'WellExposedness': 曝光量的权重，默认为1，可以指定为[0 1]内的数
 - 'ReduceStrongLight': 可以指定为'true'(default)和'false'，如果"ReduceStrongLight"为真，则blendexposure将尝试抑制图像中强光源的高光

- **结果:**4张图像融合-ReduceStrongLight设为true与false的区别

Python

- Mertens
- OpenCV: Here we show an alternative algorithm to merge the exposure images, where we do not need the exposure times. We also do not need to use any tonemap algorithm because the Mertens algorithm already gives us the result in the range of [0..1].

V.课堂总结

内容总结重述

- HDR介绍、合成、显示，分别讲了什么内容
- 哪一部分是要求掌握的，哪一部分是要求理解的，哪一部分是了解即可
- HDR成像，实际过程就是(见图)

Homework提供

- 布置Homework，主要是为同学们提供准备好的算法资料
 - 分为**同学实践部分**(Homework)和**可运行代码实现部分**(Homework_Solution,相当于一份答案)
 - 可以为大家提供完整的HDR相关知识笔记总结整理

资料提供

- **资料获取:** <https://github.com/cryingnow/Digital-Image-Processing-HDR.git>
 - PPT
 - 教案(知识点总结)
 - Homework与答案
 - 一些HDR算法相关的论文