

2021.12.12周报

纪新龙

本周计划任务

阅读论文，至少读三篇相关的论文。

具体完成情况

完成。

下周计划任务

阅读论文，至少读三篇相关的论文。

具体完成情况

之前阅读论文时候文中会提到一些state-of-art文章，并会把自己提出的方法和这些先进方法进行对比。

我觉得有必要广泛得了解这些方法。

但是论文发表时间不同，两篇论文发表的时间可能相差10年之久。方法是不断改进的，提出最前沿方法的人肯定通读、理解过在他之前所有的文章和方法。

所以在新的文章中找它提到的state-of-art方法来读，应该还是比较妥当的。我在选文章的时候，觉得最新一篇3_MEKA2021_Real_Time_Global_Illumination_Decomposition_of_Videos确实不错，所以我从他的related works中提到的几个单幅图像本征分解算法入手，去查了发表期刊的等级和引用数量（一定程度反映论文的质量），选读了如下三篇：

- 10_ Barron andMalik2015_Shape_Illumination_and_Reflectance_from_Shading
- 11_LiandBrown2014_Li_Single_Image_Layer_2014_CVPR_paper
- 12_ZHAO2012_A_Closed-Form_Solution_to_Retinex_with_Nonlocal_Texture_Constraints

10_ Barron

andMalik2015_Shape_Illumination_and_Reflectance_from_Sha

ding

提出SIRFS模型，该模型将一张图片分解为shape, surface normals, reflectance, shading, and illumination 各层。所以该算法相当于是shape-from-shading和intrinsic images的组合算法。

算法的核心是利用形状、反射率层、照明三个角度的先验。形状应当满足各向同性和表面光滑；反射率满足线性平滑和低熵，照明满足自然性。

反射率的光滑，通过相邻像素点差值的平方来惩罚；反射率的稀疏，通过不相邻像素点差值的平方来惩罚；反射率的“绝对约束”，在训练集上训练了一个模型 f :

$$\text{minimize}_f f^T n + \log\left(\sum_i \exp(-f_i)\right) + \lambda \sqrt{(f'')^2 + \epsilon^2}$$

其中的 f 是样条函数，用于计算反射率层上的像素值对应的负代价(可能性)。 n 是训练集上图像的灰度值直方图。前两项使得 f 的毕竟训练集上灰度的情况；最后一项是让样条函数更加分段线性平滑的。所以“绝对约束”的惩罚项是

$$g_a(R) = \sum_i f(R_i)$$

图形层的光滑性约束，就是要求图形表明光滑，使用的是平均曲率：

$$H(Z) = \frac{(1 + Z_x^2)Z_{yy} - 2Z_x Z_y Z_{xy} + (1 + Z_y^2)Z_{xx}}{2(1 + Z_x^2 + Z_y^2)^{3/2}}$$

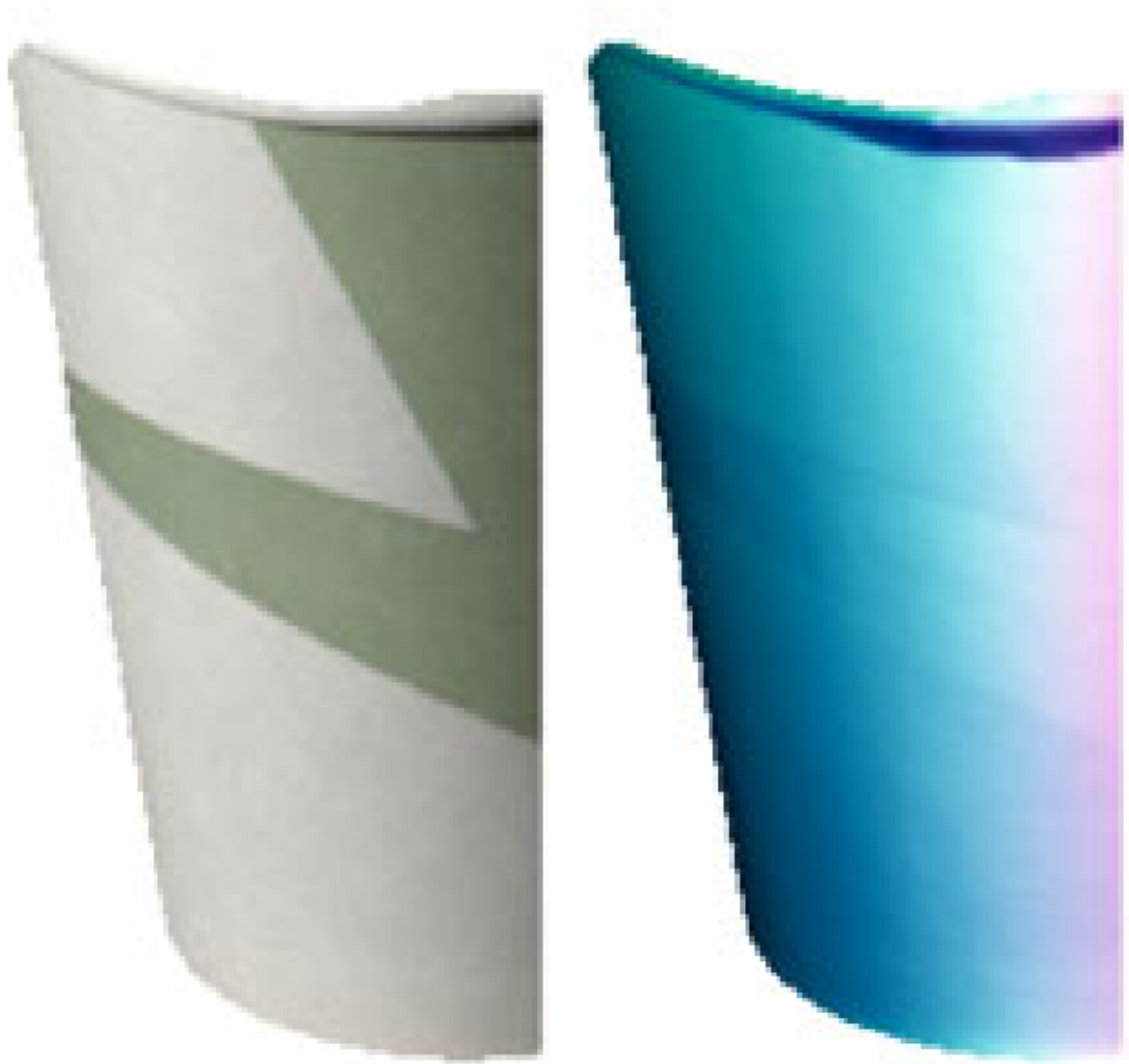
Z_k 是法向层在 k 方向的偏导

要求邻近点的平均曲率之差更小，可以让图形层更平滑。

图形层表面的各向同性约束：自然界中的物体，其表面的某一点处的法向指向各个方向的概率其实相等的。但是还有另外一个现象，就是我们看到的一张图片，里边各个像素处的表面法向指向我们的可能性更大一些。因而用了一个严格各向同性的约束项：让所有点法向量在垂直于图像平面的那个分量（正方向是从图像指向观察者）的和最大。

图形层在“闭塞处”的约束。闭塞处就是一些光照不进去的地方，可以有两种：

.The side of this cup are “limbs”, points where the surface normal faces outward and is perpendicular to the occluding contour,
while the top of the cup are “edges”, sharp discontinuities where the surface is oriented arbitrarily.



(a) A cropped object and its normals

但是作者将所有“闭塞处”都默认为“limbs”进行约束，并且声称这种约束在“edges”处表现良好。按照“limbs”模型，在数据集上训练出一个平凡解，这个通用解按照我的理解就是会把图形的边界都生成上图中水杯左下角边界那样的法向由平行于平面慢慢转向成垂直于平面的圆滑过渡边。效果见下图，输入小恐龙的掩膜，就能自动生成一个边界圆滑的凸出来图形。



(a) occluding contour normals



(b) shape-from-contour output

右边的恐龙就是平凡解。会约束待估计的各点法向量逼近平凡解。

照明层其实很简单，就是一个球谐函数照明模型。如果是灰度图像，就有九个参数，如果是彩色图，就有27个参数。

这篇论文在优化目标函数的时候使用了多尺度（金字塔）。**和标准的standard coarse-to-fine techniques方法不同的是，主要利用金字塔抽象出像素更少的高层，在高层上跑优化，然后利用高层复原出底层，在低层上计算误差，把计算出的误差反传回高层。**这样反复优化，节省了时间。

11_Li and Brown 2014_Li_Single_Image_Layer_2014_CVPR_paper

提出一种图像分解的模型。

亮点：

- 提出的图层分解算法，不仅仅能本征分解稍微改动一下就能用来去除玻璃下的图像的反光影响。
- 用傅立叶变换加速求解；针对L0项用的是half-quadratic方法迭代优化。

下图是本征分解目标表达式：

$$\min_{L_1} \sum_i \left(\sum_{j=1,2} \rho(F_i^j L_1) + \lambda (F_i^3 L_1 - F_i^3 I)^2 \right) \quad (5)$$

s.t. $lb_i \leq (L_1)_i \leq ub_i.$

作者是从梯度属于反射率层和光照层的概率分析出发的，最终推导出的能量项中的 $\rho(F_i^j L_1)$ 其实特别像丁守鸿论文里手动构建的那个L0约束函数。而右边剩下的等于常见的对光照层平滑的约束。 F^j 是j方向上的偏导算子矩阵形式； F^3 是拉普拉斯算子。第二行是一个对光照的绝对约束，要求光照在范围内。这都是常见的约束项了，在本征分解问题里不稀奇。

$$\min_{L_1, g^j} \sum_i \left(\sum_{j=1,2} (\beta(F_i^j L_1 - g_i^j)^2 + \rho(g_i^j)) + \lambda(F_i^3 L_1 - F_i^3 I)^2 \right) \quad (6)$$

为了处理L0项不可微分的问题，如上，引入了辅助变量 g 。**在交替迭代优化过程中 β 逐渐扩大，来约束第一项，这篇文章在介绍这种half-quadratic方法时简洁很多。**如同其他几篇论文，在优化 L_1 的时候用傅立叶变换来加速了。

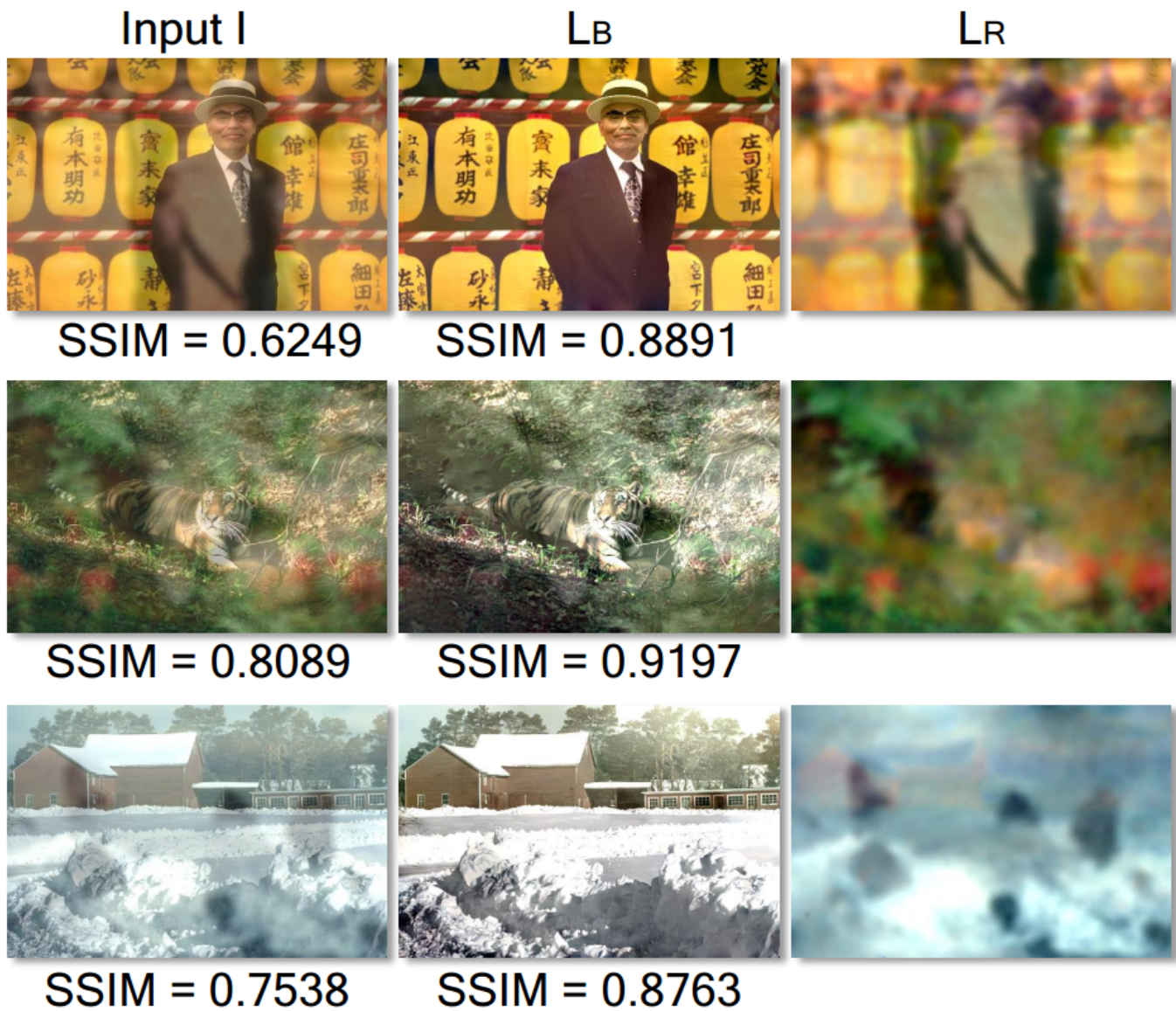
我在思考这篇文章为什么能上CCFA刊时候，不理解它为什么用了些常见方法还能被接收。想来原因可能有两个

- 分析梯度分类的时候用的是两个概率函数，虽然最后得到的最终目标函数没有多大新意
- 这种图像分解模型还能用来解决其他问题，比如把公式改一改：

$$\min_{L_B} \sum_i \left(\sum_{j=1,2} \rho(F_i^j L_B) + \lambda(F_i^3 L_B - F_i^3 I)^2 \right) \quad (11)$$

s.t. $0 \leq (L_B)_i \leq I_i.$

假如一本书是在玻璃橱窗下被拍照的，那么照片里的书就是“背景”，玻璃橱窗上的反射图就是要去除的干扰。移植本征分解中对反射率层的L0稀疏先验和光照平滑先验到这个新问题上，发现都是行的通。不同的是，本征分解中的反射率乘光照等于成像，而 single image reflection interference removal using focus中，背景加上玻璃上的反光等于成像。好在乘法log之后等同于加法。上边公式中的 L_B 表示背景本身，而不是本征分解问题中log之后的各个反射率层。 L_B 没有经过log，所以应该大于0。分解效果如下：



12_ZHAO2012_A_Closed-Form_Solution_to_Retinex_with_Nonlocal_Texture_Constraints

这篇文章主要在retinex方法基础上是引用了“纹理”这种非局部特征作为新的先验。即，如果图像中相隔很远的两个像素点，它们周围的纹理是相同的，那么这两个像素点的反射率也相同。

retinex约束：光照平滑、反射率分段常数、绝对尺度约束。这些跟丁守鸿论文里的第二个算法一模一样。主要的创新点还是“纹理相同”先验。实际上，别的算法在比较同类方法的时候都提到这种方法，但是没有引入“纹理相同”先验的。我觉得可能计算耗时上的问题。

判断两个像素点有相同的纹理：

1. 计算图像各点的色度
2. 在像素点 i 周围取一个大小为 $K \times K$ （初始 $K=3$ ）的窗口。倘若两个像素 p 、 q ，其窗口内区域的色度的SSD（sum of squared differences）小于一个阈值（作者取0.1），就认为这两个像素处在相同的

纹理环境中。为了不遗漏，要将 j 周围窗口内的区域旋转90, 180, 270 度后和不旋转的 i 比较。也可以考虑更多种旋转情况，来更精细的匹配纹理。

3. 所有在初始 $K=3$ 情况下因为一种纹理而匹配的点 $p_1, p_2 \dots p_k$ 形成一个组。计算这种纹理下所有匹配窗口的中间值，形成该纹理下的median window。为了描述不同程度的纹理匹配，当 p 属于某种纹理时，尝试增大 K 的取值为5、7、9...找到能够让 p 和median window匹配的最大窗口边长 $K = m_p$ 。

于是一个描述纹理匹配程度的权重可以构建如下：

$$c_p = m_p \left(1 - \frac{SSD}{9}\right)$$

SSD是在窗口为 3×3 时计算的，SSD越大表示点 p 和该纹理组中的平均水平越不接近， m_p 表示在更大范围内都可以称得上是匹配的。将此权重 c_p 乘在能量约束项中（类似其他论文中经常出现的相似度权重）：

$$f_r(S) = \sum_{G_r^i \in \Gamma_r} \sum_{p, q \in G_r^i} c_p c_q (I_p - I_q - S_p + S_q)^2$$

这样可以在纹理相似度更高的两点之间更加强调其反射率的相似。

文章的另外一个亮点。retinex方法中判断色度梯度是边界还是光照，需要确定一个阈值。本算法用到了retinex算法，所以不可避免的也要确定阈值。我在写代码的时候，觉得阈值的确定不够自动化，甚至可能因为换了一张图片就需要一个新的阈值。retinex本身是效果很好的方法，需要确定一个恰当的阈值可能是它最受诟病的地方了。本文提出一种自动确定阈值的方法。

其实思路很简单，就是在一个范围内随即抽取12数作为阈值，然后用这个12个阈值来跑算法，对比12张分解出的反射率层的“熵”。选图像熵最小时用的那个阈值作为最佳阈值。

这种自动确定阈值的方法，其道理在于反射率图像的混乱程度越低，越接近理想状态。我觉得这种假设是有道理的，从**熵角度来评估图像**，可以积累起来在以后用到。作者用的熵计算方法和我之前学的不一样，可能方法本来就不止一种，此处不赘述。

好处是能够全自动化运行这个本征分解算法了。坏处是运行12边算法，必然耗费算力，或许会有更好的方法。