

1 摘要

静态图片阴影去除是图像处理领域的一个问题。相比动态视频阴影去除，静态图片有着更少的信息量。对人类而言，可以轻松认出静态图像上不同物体及他们在光照下产生的阴影。但是对计算机而言却较为困难。不同的光照条件，纹理都会成为计算机识别的阻碍。同时，阴影识别和去除在工业领域也有着巨大的意义，研究阴影去除和识别正成为一个新兴的热点。

2 介绍

本项目采用 `Python+Opencv+Skimage` 完成，测试数据来自 `SBU` 阴影数据集。

再图像处理方面，首先采用了色域变换的算法，将 `RGB` 静态图像变换到 `HSV` 空间。因为 `HSV` 空间的颜色特征不变性，获得阴影区域的轮廓并进行形态学处理。随后计算阴影区域的光照补偿，计算得到阴影区域的光照状态下颜色，再处理阴影区边缘，获得去除阴影后的静态图像。

本项目创新点在于，去除阴影算法中利用了超像素块计算相同纹理材质部分的平均颜色。对于不同的阴影片区，使用了不同光照补偿强度，一定程度上解决了多纹理材质图像阴影去除的困难。在阴影边缘处理部分中，参考了 `PHOTOSHOP` 中仿制图章和减淡工具的思路，有效地将本影、半影区因光照强度不同而产生的边缘异常减到最小，并收获了良好的效果。

根据对不同参考论文中给出的不同算法结果的对比，可以发现本项目采用的算法有良好效果。具体对比参见实验部分。

3 算法模型

3.1 阴影检测

根据 `Salvador E` 提出的颜色特征不变理论，将图像转换到具有对阴影颜色特征不变性质的色域空间中，物体阴影本身的特征将被保留，且放大和周围环境的差异，可以更方便地提取阴影轮廓。随后，将变换后的图像转变为灰度图。以不同灰度值为阈值，计算对应灰度阈值前提下，统计原图前景和后景的标准差。前景和后景分别为小于和大于阈值的像素点。利用 `OSTU` 大律法计算出对灰度图二值化后的结果，可以获得比较可靠的阴影片区。但是，直接获取的阴影区有较多的噪声。我采用了基于马尔可夫随机场的降噪模型对得到的阴影区二值图降噪。这时候二值图已经初具可以利用的价值，但是检测出来的

阴影依旧有不少瑕疵。进行形态学处理后，使得原本生硬的阴影边缘变得更加圆滑。这里采用了先闭运算再开运算的算法处理阴影结果。对比 **ground-truth**，可以发现阴影检测的效果非常接近。

对大津法可作如下理解：该式实际上就是类间方差值，阈值 t 分割出的前景和背景两部分构成了整幅图像，而前景取值 u_0 ，概率为 w_0 ，背景取值 u_1 ，概率为 w_1 ，总均值为 u ，根据方差的定义即得该式。因方差是灰度分布均匀性的一种度量，方差值越大，说明构成图像的两部分差别越大，当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小，因此使类间方差最大的分割意味着错分概率最小。

在实验过程中我发现大津法其实效率较低，因为他每次运算都需要对全图进行一次时间复杂度为 $O(nm)$ 的计算。通常来说需要计算 200 次以上才可得出最优解。用 **Matplot** 直观显示多次运算代价，可以发现在大多数时候（我尚未发现任何反例），代价曲线有且仅有一个峰值。为此我们可以在此优化。将 $[0, 255]$ 平均分为 n 个区间，只计算以每个区间第一个为灰度阈值时的代价，得到大致曲线后，找到第一个代价下降的点。计算该点所属区间及其上一个区间中总计 $2 \times \frac{255}{n}$ 个数字为阈值的代价，从而找到最优解。我们可以得到如下方程

$$2 \times \frac{255}{n} + n \rightarrow MIN \quad (1)$$

n 取 23 即可。优化后的算法速度理论上提高了 5 倍

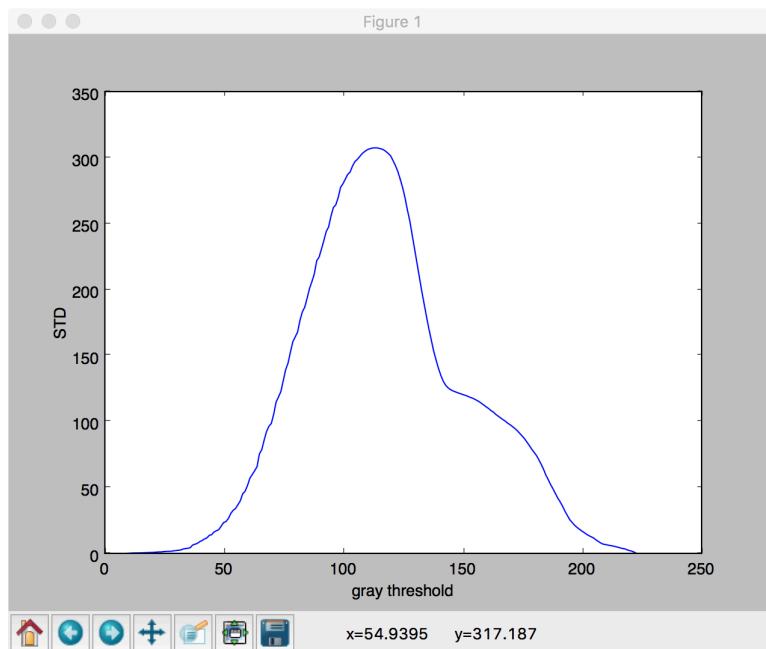


图 1：对一张图片采用不同阈值的代价曲线

3.2 阴影去除

根据 BARROW H. G., TENENBAUM J. M 等人提出的光照模型，我们可以知道

$$I(x, \lambda) = L(x, \lambda)R(x, \lambda) \quad (2)$$

其中， x 为点， λ 为波长， I 为光照强度， L 为亮度， R 为反射度。根据数学推算，可以得出一下一系列等式。

$$L(x, \lambda) = L^d(x, \lambda) + L^a(x, \lambda) \quad (3)$$

$$I^{lit}(x, \lambda) = L^d(x, \lambda)R(x, \lambda) + L^a(x, \lambda)R(x, \lambda) \quad (4)$$

$$I^{shadow}(x, \lambda) = a(x)L^a(x, \lambda)R(x, \lambda) \quad (5)$$

其中， a 是针对阴影内部区域空间亮度系数。我们假设环境光谱分布均匀，各个角度的入射光均匀，结合上述最后两个等式，可以获得如下结论：

$$I^{lit}(x, \lambda) = L^d(x, \lambda)R(x, \lambda) + \frac{1}{a(x)}I^{shadow}(x, \lambda) \quad (6)$$

记 $\frac{1}{a(x)}$ 为 γ ， $L^d(x, \lambda)R(x, \lambda)$ 可以视作不同阴影区的照亮参数，简化后为：

$$I_k^{lit}(p) = \alpha_k p + \gamma(p)I_k^{shadow}(p), k \in \{R, G, B\} \quad (7)$$

Yael Shor 归纳总结前人经验，得出了光照补偿公式。也就是说，只要计算得到上述方程式中四个参数，就可以计算出被照亮后的阴影区状态。

3.3 参数估计

需要注意的是，原图像可以检测出多块阴影区域。记检测出的阴影模块为 $Musk_i (i \in [1..n])$ n 为不同的阴影区块个数。 n 可以用联通域分析 (FLOOD FILL) 算法确定，并对不同的联通域打上对应的标记，以方便随后对每个联通域分别处理。

为了计算得到照亮部分的平均颜色，我采用了利用超像素分割的方法，每个分割块都有类似的纹理和色彩。超像素风格算法我采用了 OPENCV3.0 自带的 SEEDS 算法。随后计算阴影区域的边缘，对阴影边缘上每一个点，标记其周围的非阴影区的超像素块为照

亮区域。统计这些超像素块的平均颜色从作为对应阴影区的补偿基色，统计照亮部分灰度的标准差。随后统计相应的阴影区域的平均颜色和灰度标准差，带入上述公式中得到 γ, μ 和 α 等参数。我们记照亮纹理区域为 L ，阴影纹理区域为 S 。利用公式

$$\gamma = \frac{\sigma(L)}{\sigma(S)} \quad (8)$$

$$\alpha_k = \mu_k(L) - \gamma \mu_k(S), k \in \{R, G, B\} \quad (9)$$

其中， $\mu(S)$ 和 $\mu(L)$ 分别代表了阴影和照亮部分在 RGB 分量上的均值。 $\sigma(L)$ 和 $\sigma(S)$ 表示照亮和阴影片区灰度的标准差。

代回上述照亮补偿公式中，可以计算出对应光照区域的颜色。

3.4 边缘处理

众所周知，阴影有本影和半影的区别，但是检测出的阴影区域并没有区分这二者。通常来说，阴影的边缘区域是半影，但是边缘区域拥有较为复杂的几何形状，所以对阴影光照补偿难以计算。通常用上文方法处理后，会在背景中留下一个阴影轮廓。参考平日 PHOTOSHOP 的经验，仿制图章和减淡工具（模糊工具）可以比较好地处理边缘异常问题。

本项目算法中，先计算出阴影部分边缘。随后将边缘附近的像素窗口覆盖到边缘上，从而抵消边缘带来的异常。要注意的是，从待覆盖像素附近取得的像素窗口不应该包含另一块待覆盖像素，同时，像素窗口的纹理材质应该尽可能同待覆盖区域相似。因为阴影区域是一个闭合的不规则几何图形，首先计算阴影区域的几何中心。记几何中心为 G ，待覆盖的像素点为 $P_i (i \in \{EDGE\})$ 。做向量 \vec{GP}_i ，并沿着向量方向向外延长 10 个像素点的位置，以这个位置为截取像素窗口的中心取 5×5 的像素窗口并覆盖在 P_i 上。可以看到我们成功去除了阴影的边缘异常。

但是，因为覆盖区域纹理的原因，再处理规则纹理背景的阴影边缘时，会出现截取后的阴影和本来应有的纹理不重合等问题，造成了一定的形变。这个问题需要计算覆盖区域像素最小割问题，因时间有限我没有实现这个功能。

至此，我实现了 PHOTOSHOP 中的仿制图章功能。

3.5 减淡工具（混合图层算法）

经过实验论证，当去除阴影部分亮度和环境亮度（灰度）有着巨大差异的时候，用了上述算法后，依然会发现阴影部分灰度较低，看起来有一个浅浅的影子。这是因为亮度

补偿算法的缺陷导致的。但是如果将输出结果在此迭代一次，因为较浅的阴影灰度和周围部分非常相似，阴影检测算法难以检测出阴影部分。同时，亮度补偿算法的 γ 值非常小，使得阴影去除效果不明显。

再次参考了 PHOTOSHOP 的原理，分析图层混合功能中的图层减淡模块。查询相关资料后，可以发现这样的公式。

$$P + (P \times Q)/(255 - Q) = ANS \quad (10)$$

其中， P 为基色， Q 为混合色， ANS 为结果色。当混合色为 0 （黑色），（基色 \times 混合色）为 0 ，结果色等于基色，反映了条纹图最上方的情形。黑色是“颜色减淡”模式的中性点（中性色），即任何颜色与黑色复合保持不变。选取混合色 $Q > 128$ ，即可实现颜色的减淡。对原图 RGB 三通道分别进行这一操作，可以减淡去阴影后依旧残留的淡淡的深色部分。

Q 取值为阴影片区内平均灰度值和阴影边缘附近超像素块平均灰度值的比值乘以 128

$$Q = \frac{\bar{G}_S}{\bar{G}_L} \times 128 \quad (11)$$

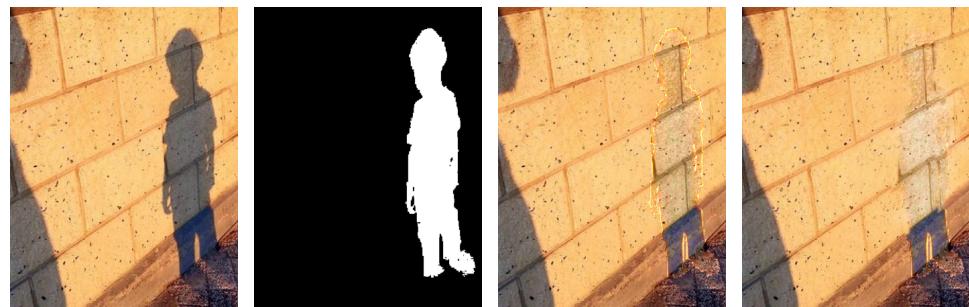
4 实验

以下列举了不同输入图片，在分别使用了上述各个算法模块之后的输出。可以看见上述算法综合之后效果不错。

分别为原图、阴影 musk、去阴影、去边缘







5 结论

可以看到，对于室外自然光，无多个光源的情况下，阴影检测效果良好。色域变换对于这种情况是非常有效的。但是到了室内，有多个光源的时候，会产生各种半影区域，从而导致检测效果糟糕。对于光照补偿模型，在上述检测条件良好，且影子没有跨过多个复杂纹理的时候，可以效果良好地消除阴影。但是当跨越多个纹理时候会产生错误。边界去除算法同理，在单一纹理的环境下可以有比较好的效果，能够去掉边缘异常。到那时到了边缘纹理复杂而且有规律的时候，因为没有检测小范围内纹理的匹配程度而导致纹理容易发生偏移异常。

对于一些效果较差的结果，因为阴影部分周围有多种纹理，所以在计算照亮纹理区域时，将不同纹理（比如蓝色的大桶）计算了进来。从而使得亮度补偿时颜色偏蓝。左下角阴影去除效果相比中部好些，因为周围地面纹理材质相似，没有混入其他纹理材质。或是阴影检测算法不够成熟，部分阴影检测出现误差，从而导致阴影去除效果较差。

总而言之，在纹理不复杂，没有多光源的情况下，阴影检测与去除可以取得良好的效果。

6 参考文献

1、[置顶] 基于 HSV 色彩空间变换的阴影检测算法——MATLAB 实现；

<http://blog.csdn.net/wuxianglong/article/details/6156919>

2、SLIC 超像素分割详解：关键代码分析；

<http://m.blog.csdn.net/article/details?id=45535203>

3、马尔科夫随机场 (MRF) 及其在图像降噪；

<http://www.cnblogs.com/simayuhe/p/5444091.html>

4、用 Python 通过马尔可夫随机场 (MRF) 与 Ising Model 进行二值图降噪；

<http://www.cnblogs.com/joyeecheung/p/4264990.html>

5、《图像阴影的检测与去除算法研究》李瑞波；山东科技大学二零一零年五月

6. The Shadow Meets the Mask: Pyramid-Based Shadow Removal ; Yael Shor Dani Lischinski School of Computer Science and Engineering The Hebrew University of Jerusalem

7. BARROW H. G., TENENBAUM J. M.: Recovering intrinsic scene characteristics from images. In Computer Vision Systems (1978), pp. 3–26.

8. KWATRA V., SCHÖDL A., ESSA I., TURK G., BO- BICK A.: Graphcut textures: Image and video synthesis using graph cuts. ACM Trans. Graph. 22, 3 (July 2003), 277–286.

9. KWATRA V., SCHÖDL A., ESSA I., TURK G., BO- BICK A.: Graphcut textures: Image and video synthesis using graph cuts. ACM Trans. Graph. 22, 3 (July 2003), 277–286.

10. Removing Shadows from Images Graham D ,Finlayson Steven D Hordley and Mark S Drew,School of Information Systems,The University of East Anglia,School of Computing Science,Simon Fraser University,Vancouver British Columbia

11、《静态阴影检测的研究进展》田建东，王占鹏，唐延东，中国科学院沈阳自动化研究所机器人学国家重点实验室，辽宁沈阳

12、减淡工具-百度百科，<http://baike.baidu.com/link?url=1dV-FyNYq-8sKlqH1KRNmSBxKsirLc-TVrWvorQz7uLWjY6dpfZ5s8F0P9E9z3V0Xlk0I29ieepQdTXboVGB5ju4>