

数字图像处理课程作业实验报告

阴影检测与去除

14307130167 蒋骐泽

目录

介绍 1

阴影检测算法..... 1

 阴影区域的特征..... 1

 基于自然光模型的一个阴影检测算法..... 1

 对于上文阴影检测算法的改造..... 2

 算法结果..... 2

阴影去除算法..... 3

 阴影去除的难点..... 3

 阴影去除算法..... 3

 算法结果..... 4

实验效果与对比..... 4

 阴影检测..... 4

 阴影去除..... 6

总结 8

参考文献..... 8

介绍

数字图像中阴影是普遍存在的，而且其为数字图像处理的很多任务，如图像特征提取，图像识别，图像分割带来了不利的影响。一个有效的阴影检测与去除方法可以为接下来的图像处理带来很多便利。

阴影检测算法

阴影区域的特征

与同表面非阴影区域相比，图像中阴影区域一般会具有以下特征：其亮度会明显比非阴影区域低；与非阴影区域有分界，界线宽度一般不大，在界线上存在渐变；阴影区域的颜色通道比例和非阴影区域比较接近。我们可以利用这些特征来完成阴影检测的工作。

基于自然光模型的一个阴影检测算法

在白天的户外环境中，主要存在两种光源：太阳光的直射，和天空的光线散射。非阴影区域同时受到了太阳光的直射和天空的光线散射，而阴影区域则没有太阳光的直射，只存在天空的光线散射。显然，太阳光的直射强度要大大高于天空的光线散射强度。对于阴影区域，它和阳光直射区域的差即为太阳直射的颜色分布。如果能够将太阳直射下颜色在三个通道的构成比例计算出来的话就可以很好的辅助进行阴影检测。田建东等[1]通过计算后得出，阳光直射在 RGB 三个通道上的强度比例为[1.31 1.19 1]。利用这个特征，就可以较为准确的识别出阴影区域。同时，他们提出，将图像进行如下公式的变换后，同表面的阴影和非阴影区域的结果会很相近：

$$Y = \log\left(\frac{\max[F_R F_G F_B]}{\min[F_R F_G F_B] + 1}\right)$$

此时利用一些图像区域分割算法即可将同一表面的内容切割到一起，然后即可进行阴影去除。其完整算法如下：

1. 将原图按照上式进行变换得到新图 Y 。
2. 使用分水岭算法在图 Y 中分割区域。
3. 对于每个区域，计算其区域内所有点的颜色均值。颜色均值指的是三个通道分别求其均值。
4. 假设在三个通道上均比均值亮的区域为非阴影区域。求出非阴影区域的颜色均值，同时乘上强度比例[1.31 1.19 1]，这个表面的性质 $L = [S_R \ S_G \ S_B]$ 。
5. 为了突出差异，找出表面性质中差距最大的两个颜色通道。令每个像素 i 的特征 X_i 为这两个颜色通道的值相减。
6. 取 T 为 X_i 的平均值。将满足 $X < T$ 的像素都作为阴影像素
7. 验证阴影的合法性：我们求出阴影区域和非阴影区域的颜色均值并作差。其颜色应该满足这个表面的性质 L 。考虑到在不同地域、不同时间、不同相机拍摄所得到的照片在强度比例上可能有一定偏差，我们认为[0.8L, 1.2L]内均为合法阴影。

8. 如果目前的阴影是合法的，那么再将所有区域内亮度比颜色均值低的点加入阴影集合。

对于上文阴影检测算法的改造

上文提到的阴影检测算法提出了一个很有用的户外环境光模型。在实践中，这个算法没有起到很好的效果，出现了以下问题：

用于分水岭算法的图 Y 效果不佳，难以将阴影和非阴影合在一起

算法的根本是要将属于同个表面的阴影和非阴影区域同时提取出来，也不能包含其他区域，否则就会对表面性质的预测和提取阴影、非阴影区域产生很大的干扰。但是实际上有很多情况下运用这个方法并不能很好的让同一表面的区域融合，而且很多时候也会出现两个不同表面在图中值相近的情况。当两个表面被当成同一区域时，这两个表面上的阴影几乎不能检出，同时很容易出现将其中一个表面完全当做阴影的情况。

选取阴影可能不满足“阴影较暗”的常识

由于选取阴影时使用的判定条件不涉及阴影颜色数值的绝对大小，而是两个颜色通道的差值，因此可能会出现虽然像素亮度较低，但是差值却比高亮度像素还要高的情况，导致高亮度像素反而被当作了阴影。之后在接下来的处理中，甚至会出现整个区域内的点都会被当作阴影的情况。

由于存在上述问题，我们对上文的阴影检测算法进行了如下改造，来减少这些问题造成的影响。改造算法如下：

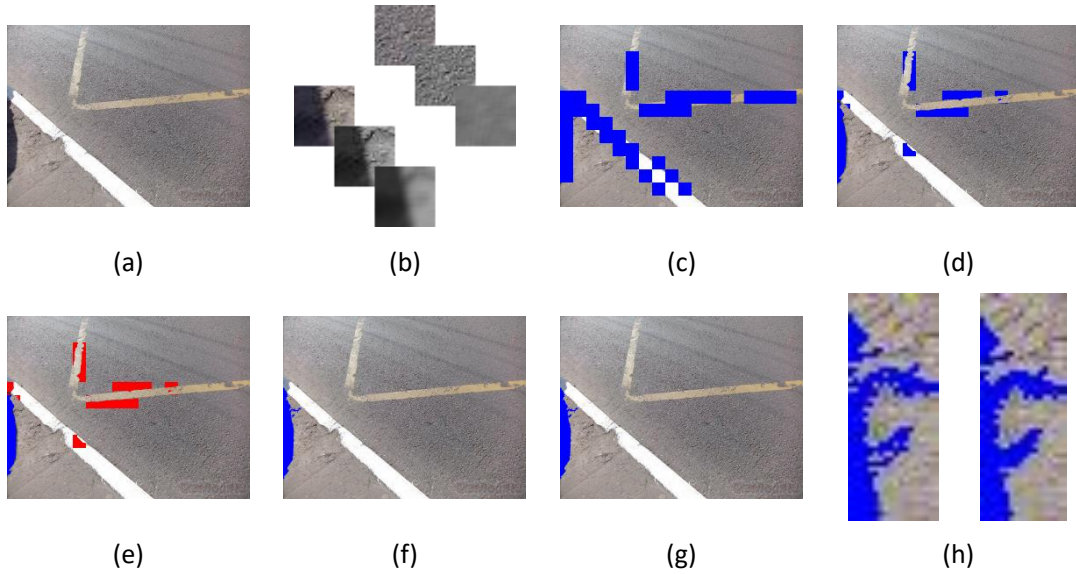
1. 不再使用图 Y 及分水岭算法。考虑到需要去除的阴影在图像中的面积一般较大，而且阴影和非阴影区域会有比较明显的亮度差异，因此使用简单的正方形分块检测办法，将原图像切割成 30×30 的块，取灰度值，并将其与 5×5 的全 1 矩阵作卷积减少噪点影响。由于阴影和非阴影区域亮度差距大，所以需要满足这个块中最亮点和最暗点差距大于 40。同时，该区域应该存在较多点接近最亮点与最暗点，因此认为在归一化后该区域的标准差应该大于 0.25。只有满足以上要求的块才继续进行接下来的检测。
2. 类似上文方法，选出非阴影区域，估计表面性质。在选取阴影像素时，需要满足选出来的像素不能已经被作为非阴影区域使用，避免将过亮的点判断为阴影。如果候选阴影通过了上文的检测，那么就将当前块放入总阴影候选中。由于之后会进行延伸，所以省略上文的步骤 8。
3. 由于阴影较暗，因此取总阴影候选的亮度平均值，并丢弃亮度高于此平均值的候选。剩余在总阴影候选的即认为是阴影，并将从这些阴影开始扩展，查找完整阴影。
4. 对于每一个属于阴影的块，以这个块的点开始做 FloodFill 算法。FloodFill 采用以 SeedPoint 的颜色作为全部的基准的方式，最大差阈值则设为该块在进行上文算法中非阴影区域和阴影区域平均亮度的平均。
5. 在 FloodFill 中，阴影图像可能会存在大量的黑白噪点。因此将图中连通块大小小于 1000 的连通块颜色反转来消除它们。至此完成阴影区域的检测。

组图 1 为算法执行流程示例。

算法结果

进行改造后的阴影检测算法对于大部分阴影均能够准确并完整的找出。但是该算法也存在一些问题：在面对一些颜色特别复杂、丰富的场景时，容易将表面交界处错误识别；在绿

地场景等，即某些通道颜色强度大大高于其他通道时，无法通过自然光模型验证而无法检出阴影等。详细内容会在实验效果与对比一节展示。



组图 1，改造后阴影检测算法的执行步骤。(a)原图。(b)对原图进行切割后两个块的处理样例，左上至右下分别为原图、灰度图、卷积图，左下的方块被接受，右上的方块被拒绝。(c)步骤 1 中所有被接受的方块。(d)通过步骤 2 的检测所得到的阴影候选。(e)筛去平均亮度高于阈值的阴影候选。红色为被筛去的部分。(f)通过阴影候选进行 FloodFill 算法。(g)去除小连通块噪点。(h)图 e，图 f 与图 g 的细节对比

阴影去除算法

阴影去除的难点

在找出阴影区域后，就需要对阴影进行去除。在进行阴影去除时，需要解决这些难点：

1. 不同的阴影所在表面有着不同的性质，不能简单的通过调整亮度、对比度的方式完成。
2. 同一个阴影中，由于光源不为完美的点光源，阴影边界会存在轻微模糊，影响去除。
3. 在一些较高较宽物体的阴影中，由于不但太阳直射光被遮挡，天空的光线散射也被部分遮挡，使得同一个阴影的不同位置阴影的深度不相同。
4. 阴影检测算法可能不够完美，一些阴影的边缘未被当作阴影的一部分

我们的阴影去除算法需要考虑到阴影去除的这些难点，并尽可能的去除这些难点所带来的对结果的影响。

阴影去除算法

基于上面对于阴影去除难点的分析，我们使用如下的算法进行阴影去除：

1. 将一张阴影图像拆成若干张，每张上面为一个连通块的阴影。这样将不同表面的阴影分开处理，避免相互干扰。
2. 对于一张阴影，将阴影扩大 5 像素。由于阴影检测时可能将阴影边缘并不是很暗的点作为非阴影区域，如果忽略这些点会导致阴影去除结果出现一圈黑边，因此扩大阴影范围以期望能包括进阴影边缘，并在之后进行去除。

3. 由于在步骤 2 中我们包括了一些亮度相对较高的阴影边缘，以及阴影自身可能存在的亮度不一的问题，我们不能对这块阴影直接作简单的阴影去除。一般来说，距离阴影边缘距离相等的点集，它们与原来相比的亮度降低比例是相近的，所以我们可以简单的将阴影的像素按照其距离阴影边缘的距离分成若干组，每一组分别进行阴影去除。
4. 对于给定需要进行阴影去除的像素点集，我们首先要知道无阴影下应该是什么样子的，即需要得到一个无阴影的像素点集。简单起见，我们选取了与阴影边缘距离大于 5 像素，又小于 10 像素的点集作为无阴影像素点集。由于三种颜色通道相对独立，所以我们可以分开处理三种颜色，这样阴影去除就是基于灰度图的阴影去除。假设一个像素无阴影时亮度 L 和有阴影时亮度 S 满足以下等式： $L = \alpha + \gamma S$ 。我们认为，阴影会改变一块区域的亮度和对比度，即均值和标准差，而去除阴影后均值和标准差将和无阴影区域一样。因此，我们可以得到 $\gamma = \frac{\sigma(L)}{\sigma(S)}$ ，其中 $\sigma(x)$ 为 x 的标准差，以及 $\alpha = \mu(L) - \mu(S)$ ，其中 $\mu(x)$ 为 x 的均值。计算出 α 和 γ 以后就可以运用变换公式去除阴影。

算法结果

在一些画面和阴影颜色比较单一的图片可以取得很不错的效果。在其他图片会留下一些涂抹的痕迹，或是阴影区域和非阴影区域的边缘比较突兀。对于部分图片无法完全去除阴影，或是阴影和非阴影区有很明显的色差。详细内容会在下一节展示。

实验效果与对比

阴影检测

组图 2 展示了部分阴影检测的结果。

可以看到，在图中表面数量少、阴影面积大的情况下，如前四行，阴影检测算法都能得到比较好的结果，与参考结果十分接近。(a)中手指交叉中间的部分检测时被当做阴影，其主要原因是这块部分的连通块太小，在去除噪点的时候被当做噪点去除了。(c)的右上角和(d)的右边缘的小块阴影没有被检测出来，可能是因为与其相邻的非阴影区面积较小，或是阴影区域面积太小，使得切成正方形后内部对比度不够大导致。同时，在(a)的手腕处，(b)的阴影左上部分，(c)的坐下手指部分均有一些不属于阴影的线条或小块被当做阴影。这是因为在进行 FloodFill 时主要考虑的是点的亮度，这些地方在原图上亮度就较低，因此被当做阴影的一部分。

(e)与(f)是算法检测阴影失败的例子。(e)图像的特点是图像为绿草地，因此图像的 R 和 B 分量和 G 分量相比特别少。因此，在进行自然光模型检验的步骤 7 时，其颜色差无法达到所设定的颜色差标准，导致所有可能阴影块均被舍弃，无法检测出图像中的阴影。(f)中，检测出的阴影大大超过了参考结果，将车辆的大部分都当做了阴影。导致这个问题的原因是车辆相对图像中其他部分亮度偏低，因此在从阴影开始做 FloodFill 时，就将亮度偏低的车辆全部当做阴影包括进来，导致结果严重失真。



组图 2，第一列为原图，第二列为检测结果，第三列为参考结果

阴影去除

组图 3 展示了运用不同的方法进行阴影去除的结果。

可以看到，当阴影覆盖表面颜色比较简单时，阴影去除算法有着不错的效果。图(a)(b)(c)(e)(f)几乎还原了图片没有阴影的样子。对于阴影去除使用的方法做了对比，分别是使用了阴影扩张和按距离分组方法；仅使用按距离分组，不进行阴影扩张方法；按距离分组和阴影扩张方法均不采用。在两种方法均不采用的情况下，可以发现在几乎每一张图中，在阴影边缘都存在因为原本亮度不是很低，在变换后颜色过亮的白线。同时，不进行阴影扩张导致了后两种方法在阴影的边缘处基本都有一圈黑边。

但是简单的阴影扩张也存在问题。以图(e)左下方手的阴影为例，不进行阴影扩张的情况下并没有出现大量像素的高亮，但是进行了阴影扩张后手指间的非阴影区域几乎都变得高亮了。出现这个问题的原因主要是进行了阴影扩张后，本来通过 5 像素扩张进来的手指边缘，因为像素都被填充变成距离边缘几十像素，导致这些像素被严重错估亮度，引起了像素高亮。

同时，也有不少图像的阴影去除不够理想，以图(d)和图(g)为例。图(d)中，虽然边缘比较平滑，但是三种方法处理后依旧显得偏暗，尤其是有半浅色瓷砖被阴影遮挡住的部分。这很明显的体现了当阴影横跨多种表面的时候会导致消除阴影的效果变差。而在图(g)中，颜色偏差则更加严重，除了阴影横跨两个表面以外，还有简单的取阴影边缘所有像素点作为非阴影区域，把两个不同表面的像素点混合到了一起当作特征的原因。



组图 3，第一列为原图，第二列为包含阴影扩张和按距离分组，第三列仅按距离分组不进行阴影扩张，第四列按距离分组和阴影扩张都不进行

总结

通过提出的阴影检测和去除算法，在一些数据上能够取得较好的效果。同时，在阴影检测和阴影去除算法两个方面都有一些不够好的方面可以改进。

在阴影检测方面，将图像切块时切成了固定的 30×30 的大小，这样实现比较简单，但是对于一些地形复杂，或是阴影面积较小的时候就无能为力。改进方法可以是使用更加精准的区域划分将阴影区域和非阴影区域划分到一起。候选阴影判定使用了自然光模型，对于阴影检测起到了不错的效果。但是由于其推导的颜色相关性的限制，在一些颜色通道单一，如绿地上的阴影检测效果很不好，基本无法成功判断出阴影。改进方法可以考虑更换模型，或是使用多种模型来减少这类特殊情况下的误判。在确定了候选阴影后的 FloodFill 算法大部分情况下能够找出阴影，但是也会出现将一些场景中原本亮度较低的部分当作阴影的情况。这个问题的解决办法可以是不仅仅采用简单的 FloodFill，而是利用颜色，位置，附近点的信息等来辅助 FloodFill，减少误判的情况。最后一步去除噪点使用了简单的消去较小的连通块的方法，这会导致原本面积较小的阴影及非阴影区域被删去。因此可以根据连通块的亮度及与附近的相似度等判定是否将其删去。

在阴影去除方面，虽然将阴影按照连通块分别处理可以减小影响，但是即使是同一个阴影也有可能横跨多个表面，因此可以通过区域点的均值、方差等要素更好的切割阴影。为了解决阴影边界和阴影内部亮度差异的问题，使用了将阴影按照到边界的距离切成一条条分别做去除的方法，但是阴影亮度变化并不一定和它到边界的距离成比例，可能一些亮度相同的点到边界的距离差距很大。因此可以使用亮度等高线等方法将阴影切割并做阴影去除，减小影响。在进行阴影去除选取非阴影点集时，简单的取了阴影外缘一圈的点，这样可能会取到不同平面上的点，影响去除效果。在将阴影更合理的切割后，我们可以在其外缘选取颜色特征与其相近的点作为非阴影点集。目前算法使用的阴影变换仅为简单的使阴影内外点的标准差和均值相同，可以提出更加优秀的变换模型，来改进变换结果。

参考文献

- [1]Tian J, Sun J, Tang Y. Tricolor Attenuation Model for Shadow Detection[J]. IEEE Transactions on Image Processing, 2009, 18(10):2355-2363.
- [2]Sun J, Tian J, Du Y, et al. Retinex theory-based shadow detection and removal in single outdoor image[J]. Industrial Robot, 2009, 36(3):263-269.
- [3]Tian J, Zhu L, Tang Y. Outdoor shadow detection by combining tricolor attenuation and intensity[J]. EURASIP Journal on Advances in Signal Processing, 2012, 2012(1):116.
- [4]Shor Y, Lischinski D. The Shadow Meets the Mask: Pyramid-Based Shadow Removal[C]// Computer Graphics Forum. 2008:577-586.
- [5]田建东, 王占鹏, 唐延东. 静态阴影检测的研究进展[J]. 信息与控制, 2015, 44(2):215-222.