

自适应去雾算法归档

概述

本文主要介绍一种基于限制对比度自适应直方图均衡和暗通道优先的图像去雾算法的新式图像去雾算法，新的算法主要针对在部分场景下 CLAHE 算法容易过饱和，还有暗通道法在部分场景效果不明显的问题，做出了一定改进。

算法介绍

算法主要是为了实现在多种实际场景下实现较为鲁棒的去雾效果，往期算法虽然能在标准测试图下取得不错的效果，但对于复杂的实际场景下应用表现却不尽如人意，而且一些算法计算复杂度较高，比如使用软抠图或者导向滤波的暗通道法，还有使用 Maccan 迭代的 Retinex 法都无法实时完成计算。本报告中提出的算法，在复杂场景下鲁棒性较高，并且拥有较高的计算速度，在 SurfacePro3 i5 4300u(1.9Ghz) 8G RAM 条件使用 Matlab r2016 下处理 720p 图像，不进行优化可以做到 5FPS，做并行优化后可以更快。

算法主要想法是来源于在优化暗通道优先的去雾算法后发现虽然算法的速度极大提高，但是在过曝场景下效果欠佳，而且对比度也不是很强。然后又实验了一些别的优化算法，比如 Color-Prior 方法，也有类似的问题。Retinex 方法，通过 SCR、MSCR、McCann 之类方法估计大气光然后去雾，迭代法可以获得较为精确的估计，但是实际使用中发现也容易过饱和，而且计算量并不低。自适应直方图均衡，发现对小动态场景较好，大动态场景效果较差，使用 CLAHE，分块自适应直方图均衡化后发现有所改观，但是默认算法使用 Lab 空间进行处理，效果不是很好，如果使用 Hsv 的亮度和饱和度处理容易过饱和，使用 RGB 分开处理容易出现杂色，于是重写了 RGB 整合处理的算法，改良的色彩观感，但是均衡过度，容易让亮度不均匀，最后改用均衡 RGB 的算法做，色彩观感和亮度均匀性得到了一定保证。根据前面实验得到的经验，在高亮度场景下倾向于使用 CLAHE，在大对比的条件下则增加暗通道法的权重。具体使用一个非线性的权重计算公式来更好的适应。算法在 Matlab 下验证，并使用 C 语言重写，在 ADI DSP 平台测试通过。

算法原理

首先，从采集设备，或者已经采集的图像处取得原始 RGB 图像，读取 8 位 RGB 图像 I_{source} 后，进行暗通道去雾图像获取。通过比较每个像素 RGB 三个通道亮度最小值得到图像的暗通道图像 I_{dark} ，同时计算图像暗通道图像 I_{dark} 各个像素的亮度平均值 L_{mean} 和暗通道平均亮度 $L_{darkmax}$ ，和图像最大亮度 L_{max} 。可通过将图像分为多个块并行计算，大大提高速度。

然后，通过暗通道图像和暗通道图像的平均亮度比较来获得传输函数。 $L_{trans} = \text{Min}(I_{dark}, \text{Min}(0.9, 1.3 \times L_{mean} \div 255) \times I_{darkaverage})$ 。参数 其中 Min 是逐点进行比较的函数， $I_{darkaverage}$ 是横向相邻 8 个像素暗通道亮度的平均，比较加权后的暗通道平均亮度和加权后暗通道图像，取得规则化后的暗通道图像，生成一个和原始图像大小相同的传输函数度量图像。可通过将图像分多块来加速算法。

通过 $L_A = 0.5 \times L_{darkmax} + 0.5 \times L_{max}$ ，对大气光参数进行估计。

最后, 通过 $I_{HazeFree} = \frac{I_{source} - I_{trans}}{1 - \frac{I_{trans}}{L_A}}$, 对图像进行去雾, 获得去雾后图像 $I_{HazeFree}$, 这里如

果对计算速度有要求, 可以简化计算为 $I_{HazeFree} = I_{source} - I_{trans}$, 根据亮度进行粗糙的拉伸, 并不会严重影响去雾质量。同时算法可以通过图像分块处理进行加速。这一步同时计算出 $I_{HazeFree}$ 的灰度图的平均亮度 $L_{hazefreemean}$ 。

之后计算改进的对比度限制自适应直方图均衡去雾, 首先对输入图像 I_{source} 分块, 这里我们分 10×10 块, 需要注意分块后必须每块的大小一样, 如果不同则需要在边缘做镜像拓展, 处理后裁剪出来。

设置削减系数为 $ClipLimit = 0.01$, 每个图块中像素数量为 $numPixInTile$, 逐个计算每个图像块直方图均衡后的查找表。我们首先计算得到 RGB 三个通道的直方图并平均得到向量 $imgHist$, 这里与之前通过 Lab 图像的亮度通道处理不同。计算削减量 $numClipLimit = round(ClipLimit \times numPixInTile + \frac{numPixInTile}{256})$ 。对求和 $totalExcess = sum(max(imgHist - numClipLimit, 0))$ 。Max 操作为求两个数的较大值, $round$ 为取整函数。计算平均量 $avgBinIncr = round(totalExcess/numBins)$ 。

从直方图的 0 到 255 逐个比较, 大于 $numClipLimit$ 则削减为 $numClipLimit$ 。如果小于 $numClipLimit$, 大于 $upperLimit$, 则定为 $numClipLimit$, 同时 $totalExcess$ 减去这里补充的量。如果更小则加上 $avgBinIncr$, 同时 $totalExcess$ 减去 $avgBinIncr$ 。如果这里 $totalExcess$ 没有归零则对图像处理, 从直方图 $k=0$ 开始, 设置步长为 $stepSize = max(round(numBins/totalExcess), 1)$, 从 0 到 255 如果图像直方图小于 $numClipLimit$ 则加 1, 同时 $totalExcess$ 减 1, 如果 $totalExcess$ 归零则停止, 到头则 $k=k+1$, 如果 $k=255$ 则归零, 循环继续。对处理后的 $imgHist$ 累加得到 $HistSum$, 设置指数分布化系数 $alpha = 0.4$, $vmax = 1 - exp(-alpha)$, $val = (vmax * histSum/numPixInTile)$, val 大于等于 1 的数设为 $1-eps$, eps 为一极小数用于防止 $\log 0$ 错误。 $temp = -1/alpha * \log(1 - val)$, 最后根据 $mapping = min(temp * 255, 255)$ 得到映射查找表。这里也可以直接按照一般直方图均衡算法线性拉伸, 计算压力可以降低很多, 但是对于大动态情况下效果略差。

对每个像素进行插值, 每个像素根据它当前图块, 和当前图块左边上边左上的三个图块的查找表进行插值, 根据当前点距离当前图块, 左边, 上边, 左上的 4 个图块中心的距离, 进行双线性插值。得到对比度均衡后的图像 $I_{HazeFree}$, 和图像灰度图的平均亮度 $L_{hazefreemean}$ 。

计算原图像灰度图的平均亮度 $L_{sourcemean}$ 和原图各个分块灰度图的亮度的标准差 std , 计算因

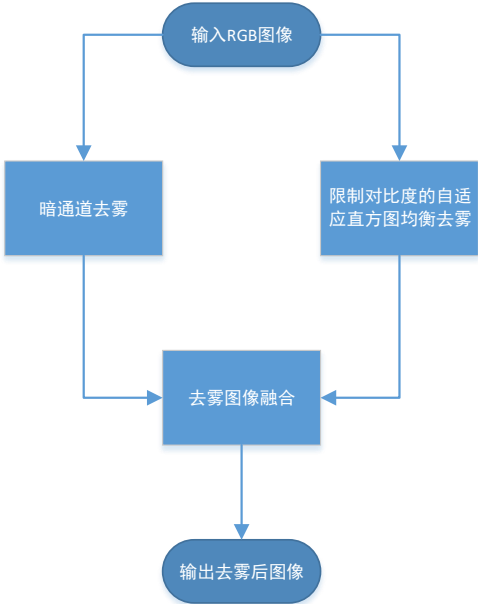
子 $Q = 9.5 \times \left(\frac{L_{sourcemean}}{80}\right)^2 \times std$, Q 设置为 5 到 70 之间。

最终合成的去雾图像为:

$$I_{HazeFree} = \frac{\left\{ I_{HazeFree} \times \left(6.0 - 4.0 \times \left(\frac{Q}{70} \right)^2 \right) \times \frac{L_{hazefreemean}}{L_{hazefreemean}} + I_{HazeFree} \times \left(2.0 + 4.0 \times \left(\frac{Q}{70} \right)^2 \right) \right\}}{8.0}。$$

算法解析

算法整体框架如流程图 1，输入 8 位 RGB 图像后，分别进行暗通道优先算法去雾和限制对比度的自适应直方图均衡化去雾。然后根据图像本身的亮度和亮度分布特点进行权重分配。最后合成去雾后图像。具体效果如图 1。



流程图 1 整体流程图



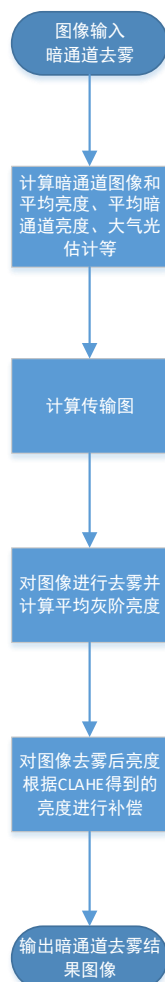
图 1 整体效果图

下面具体分析算法流程，如流程图 2 是暗通道法的算法流程，对原图逐点取三个通道中亮度最小的通道的亮度作为暗通道亮度。这里考虑了一个先验知识，在绝大多数自然图像中，都至少有一个通道的亮度接近 0，为暗通道，而对于有雾气图像由于雾气的存在，实际图像 $I = J \times t + (1 - t) \times A$ ，看到的图像是原始图像衰减后加入大气光形成的，我们在计算的时候，根据暗通道这一先验知识可以估计出传输函数和大气光，这样 $J = \frac{I - (1 - t) \times A}{t}$ ，就是

很容易能算出来的了。具体到最后就是我们估计传输图就是 $L = (1 - t) \times A$, $t = 1 - \frac{A}{L}$, 然

后 $J = \frac{1 - (1 - t) \times A}{t} = \frac{1 - L}{1 - \frac{A}{L}}$, 在这其中由于权重会被不断削减, 实际过程中 $J = \frac{1 - L}{1 - \frac{A}{L}} \approx (1 - L) * r$,

其中 r 是暗通道去雾图像和 CLAHE 亮度比值, 这种优化对最终结果也是可以接受的, 而且计算量节省可观。如图 2 左面为精细处理图像, 右为精简计算图像, 过程中图像精细计算可能效果更好, 但是对于结果而言, 精简计算不差, 而且观感可能更鲜亮。所以可以直接用后者, 对效果是没啥问题的。算法中关于传输图和大气光计算都是极度粗糙的, 大气光不论用 120 加暗通道最大值一半或者最高亮度+暗通道最大值一半, 并应用到三个通道, 都是不准确的, 但是生成较大值可以满足计算需求了, 特别是对于白平衡正确的环境下还是效果不错的。不考虑计算量的条件下有很多复杂方法了, 例如使用图中靠上方区域 1/20-3/20 区域, 中间 1/10-9/10 中间, 三个通道两两插值小于 50 的像素 RGB 三通道的平均值作为大气光参数, 实际在类似有天空的场景中效果不错, 还有 He 在论文中讲的用暗通道前 1% 像素平均作为估计, 他认为三个通道 A 一样, 其实分开做更好。传输图 He 原文是用的暗通道膨胀后软抠图或导向滤波精细化, 效果不错, 但是计算量很大, 实际算法中均衡部分保留了细节, 这里就用了简单的均值滤波代替。至于传输图三个通道其实也是可以分开做的, 因为 RGB 光在大气中传输并不是一样的衰减。对于暗通道处理后的图像, 亮度可能出现问题, 所以这里用了 CLAHE 算法的亮度作为标准进行补偿。



流程图 2 暗通道去雾流程图

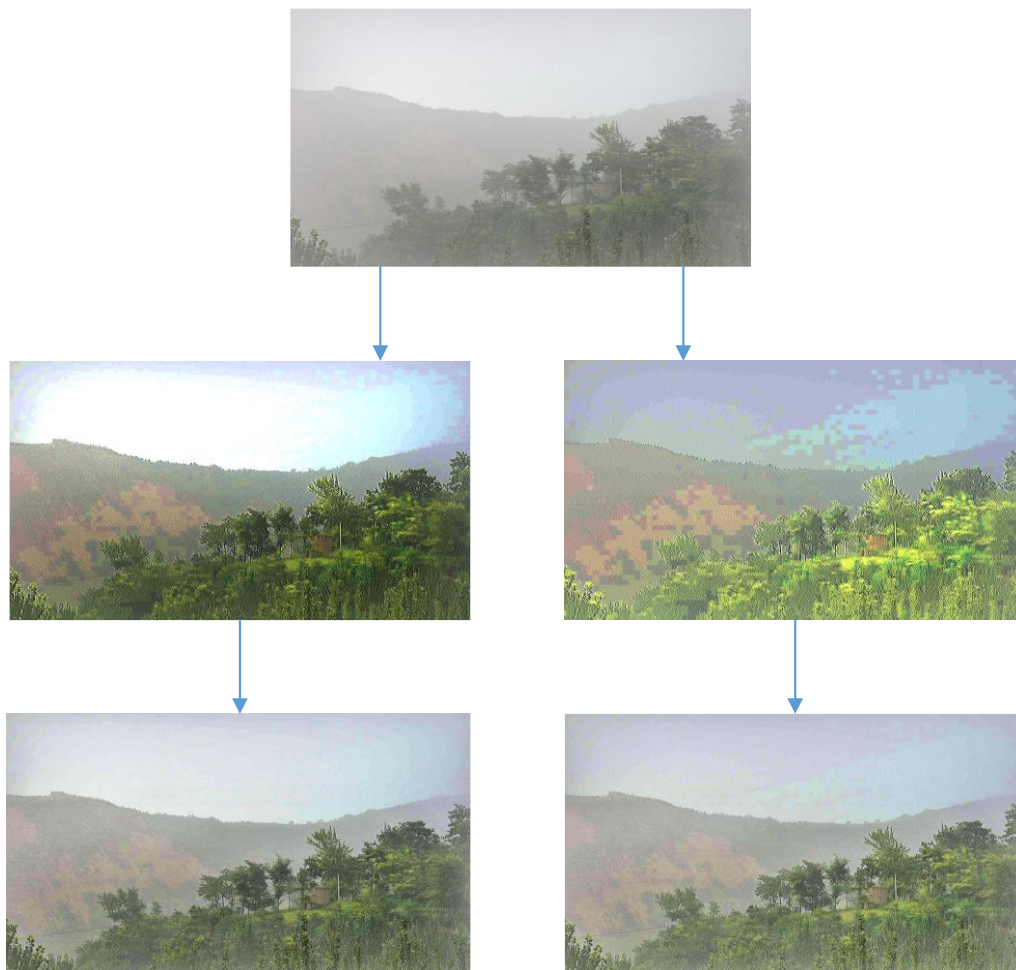


图 2 暗通道去雾对比

限制对比度的直方图均衡算法，可以归为局部对比度拉伸，将图像分为多个块然后拉伸对比度然后通过合适的方式整合在一起，让图片不要太奇怪。具体如流程图 3。图像进入系统先分块，如果分块后每个块不是一样大小，而且要都是偶数，有的块小的话，那就需要 padding 了，一半使用镜像 padding 即可，或者直接分块时候不要分成那样就好了。分块后计算每个分块的直方图，然后做限制对比度直方图均衡，所谓限制对比度就是设定一个 Clip 阈值，比如我们设为 2% 区块总像素左右，直方图中高于这个阈值的直接砍到 Cliplimit，然后砍下来的部分平均 Avg 下，小于 Cliplimit 大于 Upperlimit 即 $Upperlimit = Cliplimit - Avg$ 部分的补齐为 Cliplimit，其他部分则加 Avg。如果还有不够的部分，实际还有，处分对比度拉开了，再分配个各处，分配的策略就是可变步长加 1。加完了根据新的直方图形成映射表，这里我用了一个指数均衡来减少亮度不均衡情况下的观感问题。如图 3。

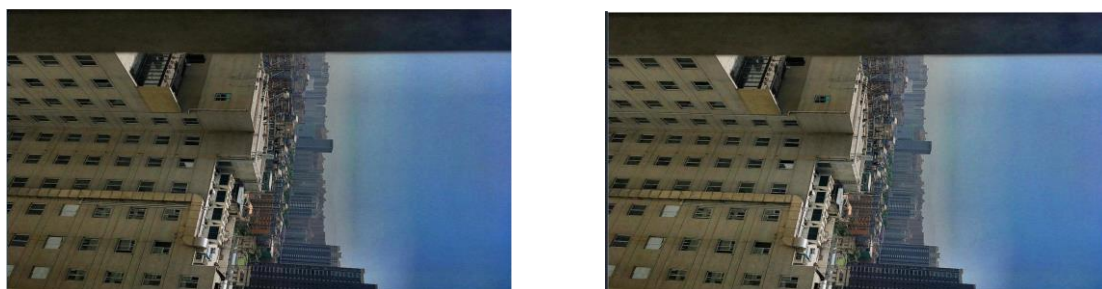


图 3 线性指数拉伸

左为指数右为线性，观感上可能左边更饱和一些，观感差距不明显。对于三个通道我直

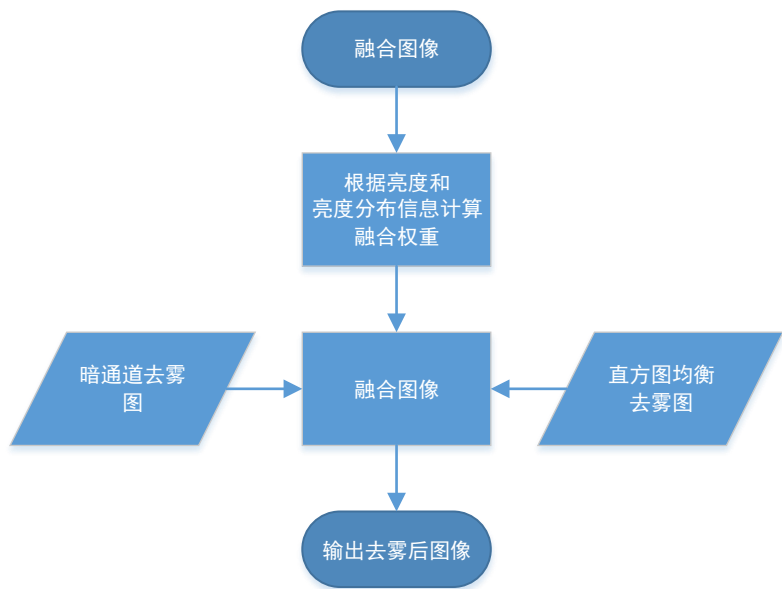
接使用三个通道一起加起来处理，发现效果不错，原来算法使用 Lab 的 L 亮度做，发现饱和度不够，HSV 的亮度和饱和度做又容易过饱和，分别作 RGB 容易出伪色，使用两端剪切拉伸容易出现亮度不均匀。最后选择这种方法效果较好。至于 Retinex 方法则是认为看到的東西是物体本身反射照射光 $S=R*L$ ，换到对数域，则变成了 $\log(S)=\log(R)+\log(L)$ ，我们可以试试用 $s=r+l$ 简化表示。对于去雾则可以直覺认为是原有 R 上加了大气光 A，有很多算法可以估计光，比如经典的 SCR MSSCR，这几种不算太准确，迭代算法那比如 Macnn 挺准确，但是也挺慢，而且一样容易过饱和，场景适应性不是很好。



流程图 3 限制对比度自适应直方图均衡去雾流程图

图像融合部分流程如流程图 4，这里主要是考虑到暗通道法在过曝场景下效果不好，在

亮度分布不均匀的场景下自适应直方图均衡可能造成边缘明暗变化不自然这两个问题, 通过计算图像本身的亮度和分块后亮度的标准差数据形成两者系数, 同时两个图像都有最少 25% 的权重, 可以防止图像权重过极端。如图 4, 就是过曝场景下的测试, 可以看到在这种情况下暗通道法的饱和度过高, 很不真实, 这与算法本身有关系, 如图 5 是 He 原文方法的效果, 本报告提出的方法效果明显好于 He 的方法。



流程图 4 去雾图像融合流程图



图 4 过曝场景

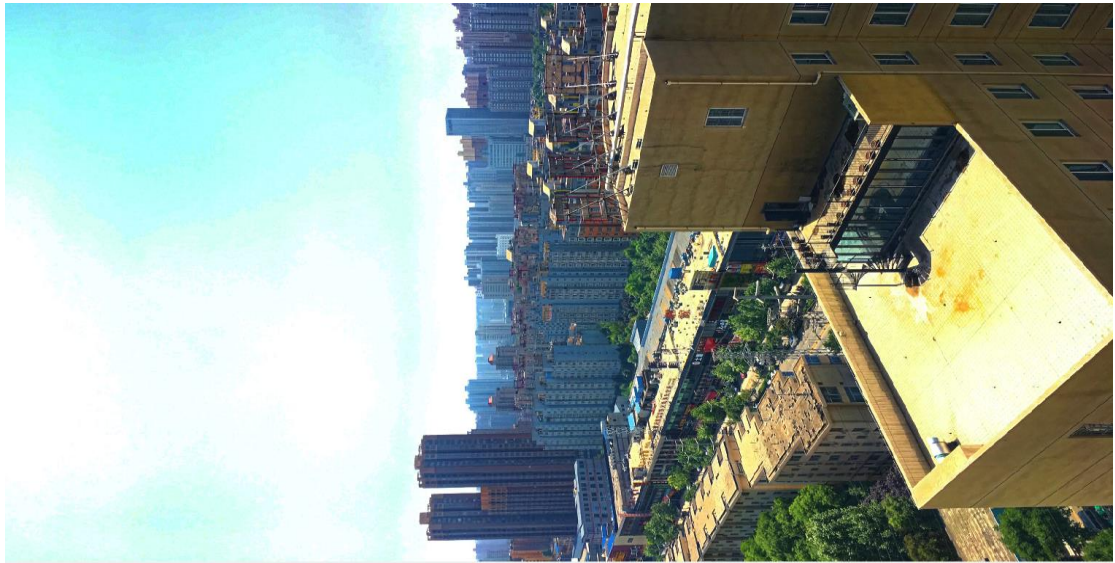


图 5 He 文中方法

对于 CLAHE 算法，则在一些亮度有反差，特别是有较大反差时候过度不自然，使用暗通道均衡，如图 6。这种问题可以使用使用其他的更精密的插值算法解决，这里使用融合去抵消一部分不自然感觉，同时更重要的是对过饱和颜色协调。如图 7，部分树叶由于 CLAHE 过饱和变的发黑，通过暗通道均衡来保证叶子鲜亮观感。



图 6 欠曝大反差场景



图 7 颜色协调

效果视频

见效果视频文件夹

待解决问题

在白平衡不正确的情况下，出现偏色问题，由于两种去雾都带有拉伸对比度的倾向，故必然会将偏色放大。

总结

本报告提出一种快速鲁棒的去雾算法，在极端条件下仍具有较好的鲁棒性，并且计算复杂度 $O(n)$ ，在实际使用中还能够按照文中提出的方案进行并行加速优化。