



Vision Cpp

高性能工业图像算法

作者：梁富楚

组织：

时间：April 14, 2023

版本：1.0

项目源码地址：https://github.com/liangfu669/vision_cpp



注意：本项目为开源项目，部分代码可能会参考其他开源项目，如有侵权，联删

目录

第一章 绪论	1
第二章 图像增强算法	2
2.1 图像光照修复算法: Retinex	2
2.1.1 SSR(单尺度 Retinex)	2
2.1.2 MSR(多尺度 Retinex)	2
2.1.3 MSRCR(带色彩恢复的多尺度视网膜增强算法)	3
2.2 光照修复: gamma 矫正	3

第一章 绪论

本文档用于描述如何 vision_cpp 算法库的部署，vision_cpp 的代码由 cuda, c++, tensorrt 开发。不同类型的算法存储在不同文件夹下，各文件夹下包含内容：

- utils_algorithm：包含图像增强算法与关键点矫正，颜色提取等算法。
- detect_algorithm：包含目标检测算法。
- classification_algorithm：包含分类算法。

第二章 图像增强算法

2.1 图像光照修复算法：Retinex

Retinex 是一种常用的建立在科学实验和科学分析基础上的图像增强方法，它是 Edwin.H.Land 于 1963 年提出的。就跟 Matlab 是由 Matrix 和 Laboratory 合成的一样，Retinex 也是由两个单词合成的一个词语，他们分别是 retina 和 cortex，即：视网膜和皮层。Land 的 retinex 模式是建立在以下三个假设之上的：

- 真实世界是无颜色的，我们所感知的颜色是光与物质的相互作用的结果。我们见到的水是无色的，但是水膜—肥皂膜却是显现五彩缤纷，那是薄膜表面光干涉的结果。
- 每一颜色区域由给定波长的红、绿、蓝三原色构成的。
- 三原色决定了每个单位区域的颜色。

基于三个假设，得到 Retinex 的基础理论：

1. 物体的颜色是由物体对长波（红色），中波（绿色），短波（蓝色）光线的反射能力来决定的，而不是由反射光强度的绝对值来决定的。
2. 物体的颜色不受光照非均匀性的影响，具有一致性，即 Retinex 算法是基于物体的颜色恒常性来实现的。
3. Retinex 算法可以将图像的动态颜色范围压缩，边缘增强和颜色恒常三个方面达到平衡，在图像除雾方面有着较好的效果（对正常图像的增强效果不明显）。

2.1.1 SSR(单尺度 Retinex)

根据 Retinex 算法的基础理论，我们可以得到以下数学表达式：

$$S(x, y) = R(x, y) \cdot L(x, y) \quad (2.1)$$

其中 $R(x, y)$ 表示物体的反射性质，即图像的内在属性，应该最大程度的保留； $L(x, y)$ 表示入射光，决定了图像像素能够达到的动态范围，应该尽量去除； $S(x, y)$ 为人眼观察到或者相机接收到的图像。对表达式 2.1 两边取以 10 为底的对数得：

$$r(x, y) = \lg[R(x, y)] = \lg[S(x, y)] - \lg[L(x, y)] \quad (2.2)$$

利用高斯滤波估计 L ：

$$L(x, y) = F(x, y) * S(x, y); \quad F(x, y) = \lambda e^{\frac{-(x^2+y^2)}{\sigma^2}} \quad (2.3)$$

算法的目的在于得到增强的图像 $R(x, y)$ ， $S(x, y)$ 是相机采集到的图像，要求解 $R(x, y)$ ，仅需得到 $L(x, y)$ 即可。在通常情况下，我们使用高斯核对 $S(x, y)$ 卷积得到 $L(x, y)$ 。算法步骤：

1. 输入高斯模糊的高斯环绕尺度 σ （即二维高斯函数的标准差）；
2. 根据 σ 对原始图像数据 $S(x, y)$ 做高斯模糊得到 $L(x, y)$ ；
3. 根据表达式 2.1 对 $S(x, y)$ 和 $L(x, y)$ 分别取对数并作差得到 $r(x, y)$ ；
4. 将 $r(x, y)$ 的像素值量化到 0 到 255 的范围内得到 $R(x, y)$ ， $R(x, y)$ 即我们想得到的增强图像。量化的公式如下：

$$R(x, y) = (r_i - r_{\min}) * 255 / (r_{\max} - r_{\min}) \quad (2.4)$$

2.1.2 MSR(多尺度 Retinex)

MSR 是在 SSR 基础上发展来的，优点是可以同时保持图像高保真度与对图像的动态范围进行压缩的同时，MSR 也可实现色彩增强、颜色恒常性、局部动态范围压缩、全局动态范围压缩，也可以用于 X 光图像增强。MSR

计算公式如下:

$$r(x, y) = \sum_{i=1}^n w_i \{Lg[S(x, y)] - Lg[F_i(x, y) * S(x, y)]\} \quad (2.5)$$

2.5式中, n 是高斯中心环绕函数的个数。 w 为权重, 满足 $\sum_{i=1}^n w_i = 1$, 当 $n=1$ 时, MSR 退化为 SSR。通常来讲, 为了保证兼有 SSR 高、中、低三个尺度的优点来考虑, n 取值通常为 3, 且有: $w_i = \frac{1}{3}$

此外, 实验表明, C_i 分别取 15, 80, 200 可以得到较好效果。一般的 Retinex 算法对光照图像估计时, 都会假设初始光照图像是缓慢变化的, 即光照图像是平滑的。但实际并非如此, 亮度相差很大区域的边缘处, 图像光照变化并不平滑。所以在这种情况下, Retinex 增强算法在亮度差异大区域的增强图像会产生光晕。另外 MSR 常见的缺点还有边缘锐化不足, 阴影边界突兀, 部分颜色发生扭曲, 纹理不清晰, 高光区域细节没有得到明显改善, 对高光区域敏感度小等。

2.1.3 MSRCR(带色彩恢复的多尺度视网膜增强算法)

SSR 与 MSR 普遍存在偏色问题, 为此, 研究者又开发出一种称之为带色彩恢复的多尺度视网膜增强算法 (MSRCR, Multi-Scale Retinex with Color Restoration), 具体讨论的过程详见《A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes》论文。其主要量化公式:

$$R_{MSRCR_i}(x, y) = C_i(x, y) \cdot R_{MSR_i}(x, y) \quad (2.6)$$

$$\begin{aligned} C_i(x, y) &= f\left[\frac{S_i(x, y)}{\sum_{j=1}^N S_j(x, y)}\right] \\ &= \beta Lg\left[\frac{\alpha S_i(x, y)}{\sum_{j=1}^3 S_j(x, y)}\right] \end{aligned} \quad (2.7)$$

其中 S_i 或 S_j 为第 i 或第 j 个通道图像, C_i 表示第 i 个通道的彩色回复因子, 用来调节 3 个通道颜色的比例, α , β 分别为受控制的非线性强度, 增益常数。

但是 MSRCR 算法处理图像后, 像素值一般会出现负值。所以从对数域 $r(x, y)$ 转换为实数域 $R(x, y)$ 后, 需要通过改变增益 Gain, 偏差 Bias 对图像进行修正。使用公式可以表示为:

$$R_{correct} = G \cdot R_{MSRCR} + B \quad (2.8)$$

2.2 光照修复: gamma 矫正

图像增强的目的是改善图像的质量, 使图像更适合于特定的应用或显示。其中一种常用的图像增强方法是伽马矫正, 它可以通过改变图像的亮度和对比度来增强图像。伽马矫正是基于伽马函数的一种非线性操作。伽马函数可以表示为:

$$I_{out} = I_{in}^\gamma \quad (2.9)$$

其中, I_{in} 和 I_{out} 分别表示输入和输出图像的像素值, γ 表示伽马值。当 $\gamma > 1$ 时, 图像的对比度会增加, 即暗区域更暗、亮区域更亮; 当 $\gamma < 1$ 时, 图像的对比度会减小, 即暗区域更亮、亮区域更暗; 当 $\gamma = 1$ 时, 图像不会发生变化。

伽马值通常在 0.2 到 2.5 之间取值, 具体取值需要根据实际情况进行调整。在实际应用中, 伽马矫正通常被用于增强低对比度的图像, 如 X 光图像和红外图像等。

伽马矫正的原理是对输入图像的像素值进行非线性变换, 使得像素值在一定范围内分布更加均匀, 从而改善图像的对比度和清晰度。但是需要注意的是, 伽马矫正可能会导致图像的颜色失真和噪声增加, 因此需要在实际应用中根据具体情况进行调整。