

CodingProject2-1 Report

张景浩 PB20010399

2023.6.8

1 问题描述

参考文章 GrabCut -Interactive Foreground Extraction using Iterated Graph Cuts^[1] 中的算法, 实现基于 graphcut 的图像分割。

2 算法原理

GrabCut 是基于图割(Graphcut)的最小割理论的一种抠图算法, 主要改进点在于 Grabcut 使用高斯混合模型拟合像素分布, 并且进行迭代的分割, 直到实现前后景分离。我们首先介绍最基础的 Graphcut 算法, 它通过将灰度图的图像分割问题转化为最小割问题进行求解, 具体算法流程如下:

1. 将图片划分为三种区域: 前景 T_F , 背景 T_B , 以及未知部分 T_U , 图片是一个灰度值的数组 $z = (z_1, \dots, z_N)$, 用不透明度 $\alpha = (\alpha_1, \dots, \alpha_N)$ 来分割图片, 每个像素点的不透明度在 0 到 1 之间, 即 $0 \leq \alpha_n \leq 1$ 。

2. 用一对灰度直方图的参数

$$\theta = \{h(z; \alpha), \alpha = 0, 1\}$$

来描述前景和背景属于哪一个灰度级分布。灰度直方图是直接从 T_F 和 T_B 区域进行建模, 所以现在的分割任务是从图像 z 和灰度直方图 θ 中预测未知区域的不透明度 α 。

3. 定义能量函数

$$E(\alpha, \theta, z) = U(\alpha, \theta, z) + V(\alpha, z)$$

其中

$$U(\alpha, \theta, z) = \sum_n -\log h(z_n, \alpha_n)$$

用来评估不透明度 α 对数据 z 的拟合程度,

$$V(\alpha, z) = \gamma \sum_{(m,n) \in C} dis(m,n)^{-1} [\alpha_n \neq \alpha_m] \exp -\beta(z_m - z_n)^2$$

是光滑项, 如果相邻的像素值相差太大, 光滑项就对这相邻像素进行惩罚。

4. 给定能量函数, 我们可以通过

$$\alpha = \arg \min_{\alpha} E(\alpha, \theta)$$

找到要分割的区域, 然后使用最小割算法进行硬分割, 即将物体不透明度分为 0 或 1, $\alpha = \{0, 1\}$ 。

Grabcut 与 Graphcut 算法的原理大致相同, 但 Grabcut 的处理结果更好, 它利用了像素点之间的相似度进行聚类建模从而区别前景背景, 只需要少量的用户交互就能得到比价好的分割效果。它们之间的差异点表现在:

1. Graphcut 的模型是灰度直方图, Grabcut 的模型是 RGB 三通道的 Gauss 混合模型 GMM
2. Graphcut 的分割是一次性完成的, 而 Grabcut 是一个不断进行分割估计和模型参数学习的交互迭代过程
3. Grabcut 允许不完全的标注, Graphcut 需要用户指定目标和背景的一些种子点, 但 Grabcut 值需要用方框确定背景区域 T_B 和未知区域 T_U 。

接下来我们给出 Grabcut 算法的具体实现步骤:

1. 首先通过用户交互确定一个矩形, 将矩形外的区域设为背景区域, 矩形内的区域设为未知区域
2. 对输入的图像进行一些标记, 指定部分像素为前景, 记前景像素集为 S , 指定部分像素为背景, 记背景像素集为 T
3. 使用一个 Gauss 混合模型 (GMM) 对前景和背景建模, 根据输入, GMM 会学习并创建新的像素分布, 对分类为未知的像素可以根据未知与已知分类的像素关系来进行分类
4. 这样一来我们就可以根据像素的分布创建一个图, 图中的节点就是像素点, 我们让所有的前景像素都和虚拟节点 s 相连, 每条边流量为 $+\infty$, 让所有背景像素都和虚拟节点 t 相连, 每条边流量为 $+\infty$ 。

5. 对于图内部任意两个相邻的像素点，我们将它们连接，边流量为

$$W(u, v) = e^{\frac{-\alpha ||I(u) - I(v)||^2}{2\delta^2}} + \epsilon$$

其中 α 是相似度 Gauss 函数系数， δ 是相似度 Gauss 函数标准差， ϵ 是边流量常量。这使得如果两个像素的颜色相差很大，那么它们之间的边流量就会很小

6. 我们对这个图使用最大流最小割算法进行分割，沿着找到的分割将像素划分，与 s 相连接的为前景像素，与 t 相连接的为背景像素
7. 继续这个过程直到分类收敛

3 代码实现

通过配置 OpenCV 库并直接使用其中的库函数 (https://docs.opencv.org/4.5.4/d8/d34/samples_2c++_2grabcut_8c++-example.html#a38) 完成对图像前景与背景的分割。

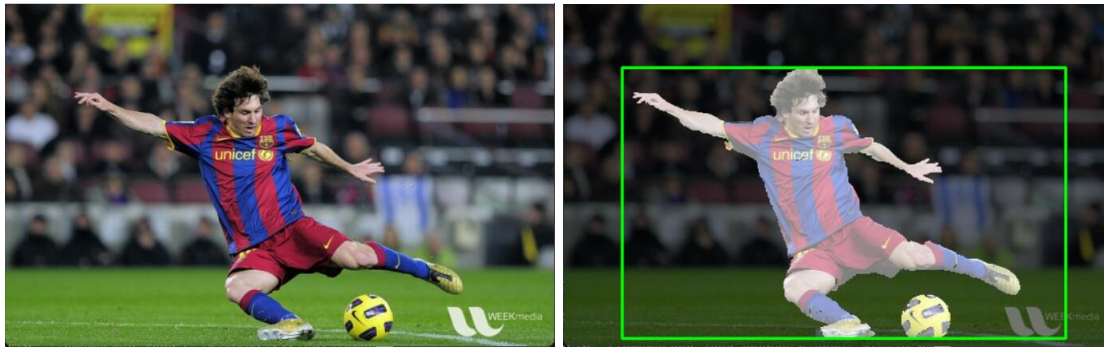
4 测试结果



源图像

图像分割结果

图 1



源图像

图像分割结果

图 2



源图像

图像分割结果

图 3

5 总结

从实验结果我们可以看出，使用 Grabcut 算法在经过一些用户交互后，得到的图像分割结果可以很好的在给定的矩形区域中区别前景和背景图像。其中通过用户指定部分前景像素和背景像素能得到更优秀的分割结果。

参考文献

- [1] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graph.*, p. 309–314.