文献阅读报告 II——Graph-Based Image Segmentation

黄磊　计 702　2022E8013282156

Short Summary

该论文提出基于图结构贪心聚类的图像分割算法，通过将像素看作图节点以构建图结构，利用图的拓扑结构快速遍历图像，并跟踪边来寻找所有层次，将每个层次看做一个独立的图像，然后对每个层次使用图像分割算法进行处理。相比于传统基于纹理的分割算法，该方法复杂度较低、效率更高，且精度可以与其他算法相媲美；此外，该算法还可以处理不同类型的图像，包括带纹路径图像和无纹图像。基于图分割的算法已经得到广泛应用，并是现代图像处理领域的重要工具。该论文的主要贡献是在基于图的图像分割领域中提出了一种高效的算法，为该领域的进一步发展打下了坚实的基础。

Introduction/background

对视觉的分组感知构成了人类视觉理解的重要部分，这对于机器视觉同样重要。如果可以很好地计算图像分割，原则上一系列的计算机视觉问题都能得到很好地突破和发展。为了实现图像分割的广泛性和实用性，该类算法应该能够捕捉感知上的重要区域，既反映全局方面，又能给出精确特征；并且应该足够高效，应至少具备与像素数量呈线性相关的复杂度，才能满足实际问题中的图像需求。

人眼视觉系统十分复杂，我们通常无法给出通用区域的定义，也不能单纯靠强度变化作为依据，因此捕捉视觉感知上重要的区域十分重要。已有的方法通常无法捕获图像感知上重要的非局部属性，或者运算效率低下，无法大规模部署。本文中所提出的基于图的分割方法的主要目的就是将图像分割成若干个具有独特性质的区域，然后从中提取感兴趣的目标，找出视觉上一致的区域，保持低变化区域的细节、忽略高变化区域的细节，并且能具有与像素数量呈线性关系的低复杂度。

Relative Work and Knowledge Preparation

1. **图的概念**

   用 $G(V,E)$ 表示一个图结构，$v \in V$ 表示顶点，$(v_i,\ v_j) \in E$ 表示相邻顶点的边，每个边带有一个权重值 $w(v_i, v_j)$，用于衡量两个顶点的差异性，一般差异越大权重越大。

2. **最小生成树**

   取其中一些边，使得无向图中任意两个顶点都有相同路径，构成连通图。如果连通图能包含所有顶点，则构成生成树。如果该生成树各边权重值和最小，则构成最小生成树。

3. **区域间的比较预测**

   需要给定一种度量方法，判断分割的两个区域是否应该被分开。其依据是，分割线两侧元素之间的差异相对于单个区域内部元素的差异，本质上是比较区域之间的差异与区域内部的差异，因此对图像的局部特征具有适应性。具体定义见下面算法部分。

Algorithm

每个像素点代表图的一个节点 $v_i \in V$，相邻像素点连接构成边 $(v_i,\ v_j) \in E$，计算每一个像素点 4 邻域或者 8 邻域的不相似度构成边的权重，以此将图像抽象表示为加权图的结构。定义以下度量方法：

1. **区域内部差：** $Int(C) = \max_{e \in MST(C,E)} w(e)$，其中 $MST(C,E)$ 为该区域的最小生成树，该公式得到 MST

边上的最大权重；

2. 区域间差别：$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_1, v_2) \in E} w(v_i, v_j)$ 得到两个分割区域之间顶点相互连接的最小权值；若没有连接，则定义为无穷大；

度量方法是两个区域之间的差别是否相对于各自区域内差别足够大，并给定一个阈值函数，控制两个区域之间的差异必须大于内部差异，以便表明区域之间有边界，对于区域 C ，定义阈值函数：

$$\tau(C) = k/|C|$$

其中，k 为待调节的常数，|C| 代表区域大小。据此定义度量为：

$$MInt(C_1, C_2) = \min\{ Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2) \}$$

$$D(C_1, C_2) = \begin{cases} true & if\ Dif(C_1, C_2) > MInt(C_1, C_2) \\ false & otherwise \end{cases}$$

对于任何一个图而言，都可以找到一个比较合适的分割。本文其分割算法如下：

输入为对图像抽象出来的图结构 $G(V, E)$ ，有 n 个顶点和 m 个边；输出为将图分割为多个分量 $S = (C_1, C_2, \dots, C_r)$ ：

Step1：将边按照权重降序排列： $\pi = (o_1, o_2, \dots, o_m)$ ；

Step2：从原始分割 S^0 开始，每个顶点都是其组成部分，即原始图结构；

Step3：对 $q = 1,2, \dots, m$ ，重复步骤 4；

Step4：已有 S^{q-1} ，(v_i, v_j) 是权重排序第 q 个边 o_q 的两个顶点，如果v_i 和 v_j 在互不相交的分割区域中，比较这条边的权值与这两个分割区域之间的最小分割内部差异 MInt，如果 $o(v_i, v_j) < MInt$ ，则合并两个区域，其他区域不变；否则无操作；

Step5：最后得到的就是所求的分割 S。

该算法使用排序和路径压缩的并查集失陷，算法耗时主要在 Step1 的排序以及迭代操作中，后者复杂度为 $O(m\alpha(m))$ ，且其中 α 是一个缓慢增长的 Ackerman 函数，可以近似看作一个常数，因此该算法时间复杂度与像素点数量呈线性增长相关。

Discussion

文中涉及多个引理证明，在此不介绍证明过程，梳理一下引理结论：

1. 两个通过边连接的两个互不相交的区域，如果它们不合并，这两个区域就将一直保留到最后；
2. 上述算法会得到一个较为合理的分割，不太精细（too fine）也不会太粗糙（too coarse）；
3. 上述算法中，相同权值的不同边的顺序不影响最后的结果；
4. 在阈值函数中，如果 $\tau = 0, K = 1 - \frac{1}{v+1}$，当且仅当一个图不是作为单个分割部分时，图 G 有一个分割器最小比分割值多 v；
5. 在使用第K 分位点定义的 Dif 时，找一个分割使得既不”太粗糙“（too coarse）也不”太精细“（too fine）是一个 NP-hard 问题。

References

[1]P. F. Felzenszwalb & D. P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision, vol. 59, pp. 167-181,