论文

GrabCut展示

谢畅——计算机专业



关键内容说明

关键内容说明

• GrabCut的概述:

GMM+EM(迭代优化)+最大似然估计alpha+min cut

• 建模的过程:

GMM模拟RGB像素分布,最大似然优化参数alpha,使用min cut优化alpha

• 重要的初始化参数: γ, K, β

• GMM vs 颜色直方图建模

GMM建模

Tripmap:{TU,TF,TB}

分别对TU中的alpha:0,alpha:1进行GMM建模

• GMM原理

适用于数据点不明显分属于单一聚类或类别的情况 π , μ , Σ in every cluster $\theta = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, K = 5\}$

 $\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), K = 5$ 是固定的或者从数据中计算出来, α 通过能量函数最小化拟合

• n维高斯分布

$$p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} exp[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)]$$

• GMM混合概率公式

$$p(\mathbf{x}|\mu, \Sigma) = \sum_{k=1}^{K} \pi_k * p(\mathbf{x}|\mu, \Sigma)$$

• 使用GMM来对数据进行建模,并计算数据点 出现的概率

进而使用概率的负对数设计能量函数

能量函数

- $U(\alpha_n, k, \theta, z) = \sum_n D(\alpha_n, k_n, \theta, z_n)$
- $D(\alpha_n, k_n, \theta, z_n) = -\log\{\pi(\alpha_n, k_n) * p(z_n | \alpha_n, k_n, \theta)\}$ 评估数据对于含 α 的GMM分布拟合效果
- $V(\alpha, z) = \gamma \sum_{(m,n) \in \mathcal{C}} [\alpha_n \neq \alpha_m] exp\{-\beta * ||z_m z_n||^2\}$

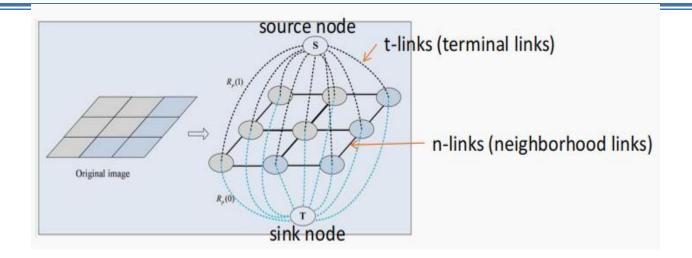
平滑项,为分类不同的惩罚项,但是高对比度的惩罚较低

• E = U + V

能量函数

- $U(\alpha_n, k, \theta, z) = \sum_n D(\alpha_n, k_n, \theta, z_n)$
- $D(\alpha_n, k_n, \theta, z_n) = -log\{\pi(\alpha_n, k_n) * p(z_n | \alpha_n, k_n, \theta)\}$
- $V(\alpha, z) = \gamma \sum_{(m,n) \in \mathcal{C}} [\alpha_n \neq \alpha_m] exp\{-\beta * ||z_m z_n||^2\}$

- E = U + V
- 目标:将能量项最小化,<mark>将TU分成前景、后景二分类</mark>
- 手段: 使用最小割可以最小化能量E, 从而更新参数α
- 建图: s, t为前景/后景, s、t与各个像素的边为数据项评估拟合, 像素节点之间的边为平滑项, 评估类别不同的惩罚。



迭代过程:

- 1. 给每个像素 Z_n 求出对应的kn,GMM组件索引
- 2. 从data z中学出GMM的参数 $(\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k))$
- 3. 使用最小割最小化能量函数E, 更新每个像素对应的 α_n
- 4. 采取EM的步骤, 即先估计, 再最大化似然概率
- 5. 用户可以交互修改Trimap, 再进行以上过程

Initialisation

- User initialises trimap T by supplying only T_B . The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets α_n = 0 and α_n = 1 respectively.

Iterative minimisation

1. Assign GMM components to pixels: for each n in T_U ,

$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$

2. Learn GMM parameters from data z:

$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$

3. Estimate segmentation: use min cut to solve:

$$\min_{\{\alpha_n: n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$

- 4. Repeat from step 1, until convergence.
- 5. Apply border matting (section 4).

User editing

- Edit: fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- Refine operation: [optional] perform entire iterative minimisation algorithm.

Figure 3: Iterative image segmentation in GrabCut

TWO 实验参数及探究

γ , β , K参数探究

• $\theta = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, K = 5\}$

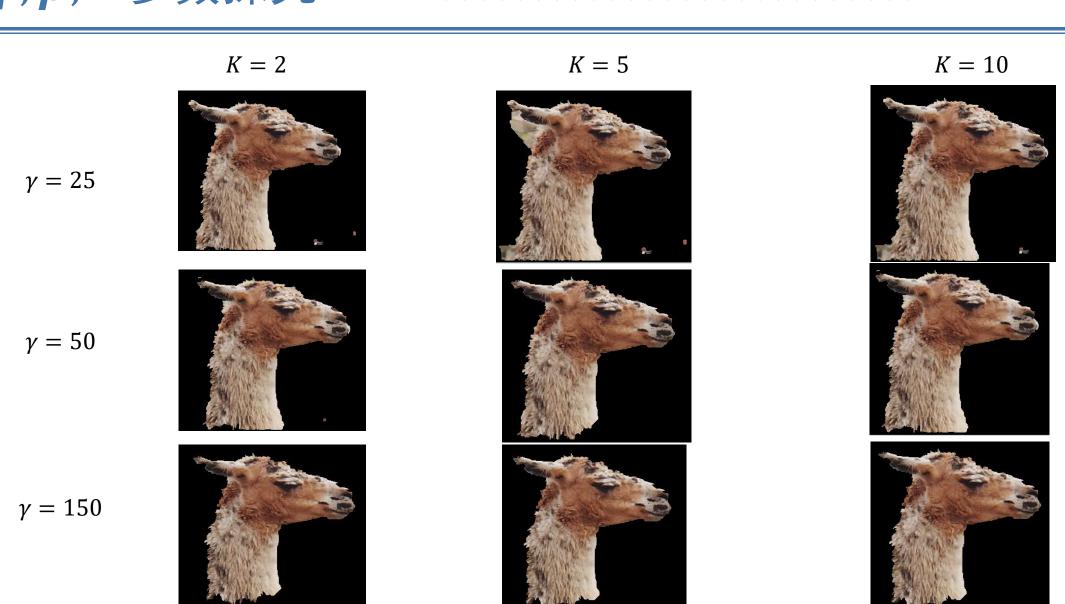
K影响GMM组件数量

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

 γ 控制平滑项的权重,即分类不同的惩罚力度,设置为50.

 β 减少高对比度的分类不同的惩罚值,根据论文[Boykov and Jolly 2001]通过Z计算出来,为了确保对不同图像均合适(高对比度、低对比度)。

γ, β, K 参数探究



γ , β , K参数探究

• K是GMM组件的超参数,对结果影响不大(小范围内)

• γ越小背景会被划分为前景,越大前景会被划分为背景,50最佳

• β减少高对比度的分类不同的惩罚值,为了确保对不同图像均合适(高对比度、低对比度),采取对全图像素差异进行一个求和..的方式。

Three

GMM vs 颜色直方图

GMM vs 颜色直方图

• 颜色直方图

颜色量化后得到颜色直方图(比如每个通道量化12个颜色值),但量化本身可能会产生略微的瑕疵。因为把一个范围的像素点归为一类。

而GMM是连续的变化,每个样本点是不同的概率值。

- 1. 这种方法简单且计算效率高
- 2. 但在处理复杂背景或颜色分布重叠的图像时,精度可能较低
- 3. GMM的参数复杂,依赖于初始化;而颜色直方图能够更稳定地表示图像中的颜色信息,同时有整体性。
- 4. 可能颜色直方图有时更合适,避免局部最优解。

PART
Four

文文进

改进

• 改进处理速度

• 把粒度放大,取代以像素为节点的力度,而是以某一个小的区域为min cut的节点预处理,再进行grabcut处理,提高处理效率。

• 采取颜色直方图与GMM相结合方式



THANK YOU