

显著性检测

朱亚菲

2015 年 2 月

目录

1	引言	1
2	多尺度的概念	1
3	Hierarchical Saliency Detection	1
4	2014CVPR-LuSong-Learning optimal seeds for diffusion-based salient objectt detection	4
5	2013CVPR-MaiLong-Saliency Aggregation: A Data-driven Approach	5
6	Saliency Detection via Graph-Based Manifold Ranking	5
7	Saliency Optimization from Robust Background Detection	5

1. 引言

由论文“A closer look at context: from coxes to the contextual emergence of object saliency”知道

2. 多尺度的概念

多尺度

3. Hierarchical Saliency Detection

这篇论文主要解决的是当图像中显著前景或背景中存在小尺度大对比度 patterns, 而在生成的显著图中并不突出这些 patterns 的情况。论文框架如图 1。主要步骤是三步: 首先从原图像中提取 layers, 然后从每个 layer 中计算 saliency cues, 最后把它们融入一个分层模型以得到最终的结果。

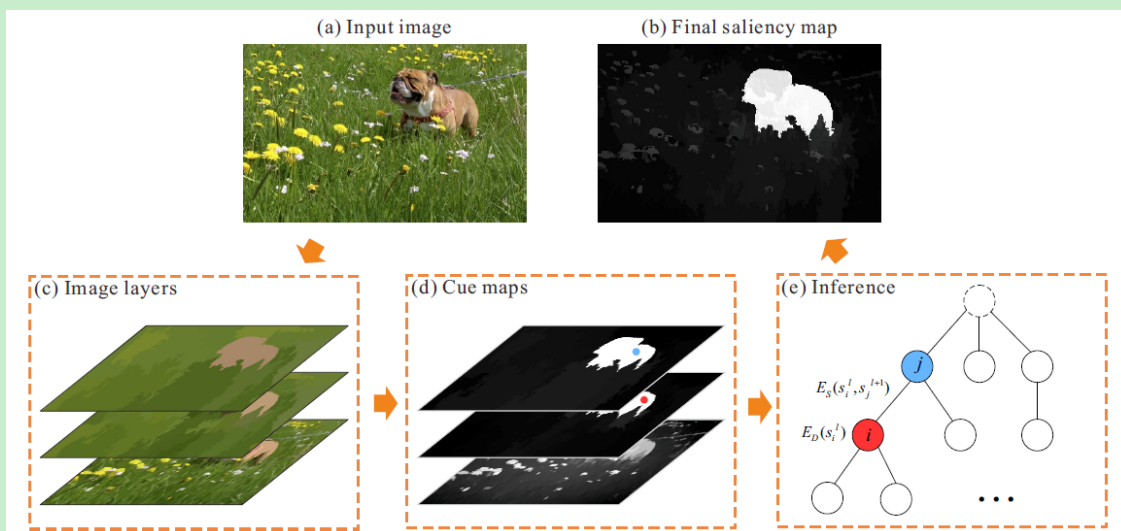


图 1: 框架

分层与多尺度、多分辨率的区别？

1. 如何提取这三个 layers？如图 2



图 2: 不同尺度下的区域合并结果

先对原图像 (400×300) 用 watershed-like 方法 [1] 进行初始的过分割, 对每个分割区域计算一个 scale 值, 然后对所有区域的 scale 值按从小到大排序, 如果一个区域的 scale 值小于 3, 就将它和最近的区域合并 (通过判断两个区域内 CIELUV 颜色均值的距离), 然后更新它的 scale, 并更新合并区域的颜色均值, 等对所有区域都处理后, 得到的结果就是 L^1 层。 L^2 层是通过对 L^1 层采取同样的步骤, 只不过用一个更大的阈值 17。 L^3 层也是如此, 阈值取 33。

2. 如何求每个区域的 scale?

通常在 Mean shift、graph-based segmentation 等超像素分割方法中, 区域的 size 是指该区域内所有像素的个数。本论文指出了这样的不合理性, 就人类视觉感知而言, 较多的像素个数和大尺度的区域并不完全符合。如图 3, 尽管弯曲的区域 a 包含了很多像素, 但对于我们的视觉感受却并不觉得它很大, 而 b 看着会更大一些, 尽管它的像素个数并不是很多。根据这样的现象, 作者基于 shape uniformities 定义了一个新的 encompassment scale measure, 以用来在合并阶段获取区域的 size。

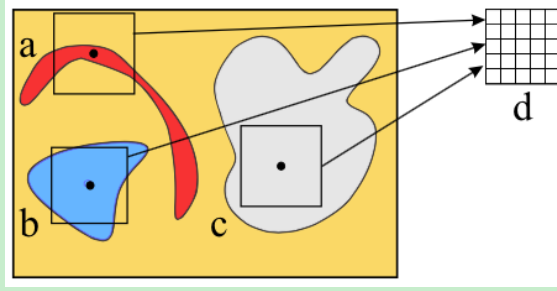


图 3: scale

关于 scale 的定义如下:

$$scale(R) == \arg \max_i R_{t \times t} | R_{t \times t} \subseteq R \quad (1)$$

其中, $R_{t \times t}$ 是一个 $t \times t$ 的正方形区域。也就是说, 一个区域的 scale 是指该区域内所能包含的最大方形区域的边长。这里并不需要通过复杂的计算来算出每个区域的 scale 是多大, 只要判断其相对于阈值是大还是小, 这样就简化了, 可以对每个区域用一个 $t \times t$ 的模板进行滤波, 如果滤完后该区域内所有像素值都被更新了, 说明该区域的 scale 小于 t , 反之说明大于 t , 如图 4 所示。

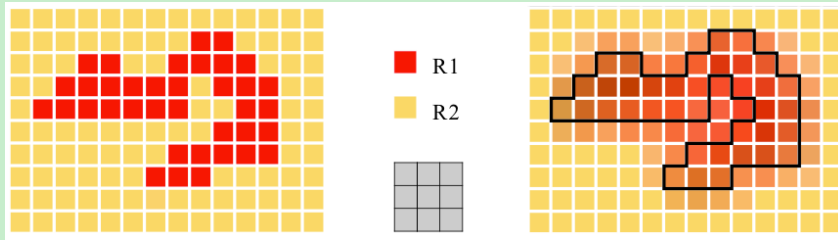


图 4: scale

3. 如何计算每一层的 saliency cues?

主要从颜色、位置、大小三个方面提取 saliency cues, 以找到该层比较重要的 pixels, 作者用了两种 cues:

1) local contrast

$$C_i = \sum_{j=1}^n w(R_i) \Phi(i, j) ||c_i - c_j||_2 \quad (2)$$

2) location heuristic

心理物理学方面的研究表示人类视觉注意偏好图像的中央区域, 所以在通常情况下越靠近图像中

央的像素越显著。

$$H_i = \frac{1}{w(R_i)} \sum_{x_i \in R_i} \exp\{-\lambda \|x_i - x_c\|^2\} \quad (3)$$

然后将 C_i 与 H_i 组合起来, 得到

$$\bar{s}_i = C_i \cdot H_i \quad (4)$$

由于 local contrast 和 location cues 都被归一化到 $[0, 1]$, 它们各自的重要性由 λ 来控制。当对三个 layers 均计算完 \bar{s}_i 后, 就可以得到每一层的初始显著图, 最后通过一种 hierarchical inference procedure 来对多尺度显著性检测结果进行融合。

4. 最后一步 Hierarchical Inference 是怎么进行的?

Cue maps 显示了不同尺度下的显著性, 效果很不一样。在底层, 会产生很多小区域, 而在高层会包含大尺度的结构。由于图像的多样性, 单独的一层并不能保证效果是完美的, 也很难判别哪一层是效果最好的。

由于背景或前景的复杂性, 单纯通过求这三层产生的显著图的平均值来融合并不是一个好的选择。作者构造了一个基于树结构的图, 见图 1 中的 (e), 其中的节点代表相应层中的区域。节点 j 在下一层中包含两个分割区域, 因而有两个子节点。其中父节点代表整幅图像的最粗糙表示。

将图中对应于第 L_l 层中第 i 个区域的节点上的显著性定义为变量 s_i^l , 设 S 是包含图中所有节点的集合。最小化如下的能量方程:

$$E(S) = \sum_l \sum_i E_D(s_i^l) + \sum_l \sum_{i, R_i^l \subseteq R_j^{l+1}} E_S(s_i^l, s_j^{l+1}) \quad (5)$$

4. 2014CVPR-LuSong-Learning optimal seeds for diffusion-based salient objectt detection

5. 2013CVPR-MaiLong-Saliency Aggregation: A Data-driven Approach

方法动机：目前在视觉显著性分析方面有很多的方法，每种方法可能对某一些图像或图像中的某些部分适用，能产生较好的结果，却没有哪一种方法能对所有的图像都适用。并且不同的显著性计算方法之间通常是互补的。因而可以尝试对这些显著性计算方法进行组合以获得比使用单一的一种方法好的效果。由于每种方法在不同图像上的表现也不同，所以对每幅图像使用的方法组合也是不同的。

1. saliency aggregation 的含义是什么？

saliency aggregation 应该是显著性融合的意思，一般的融合方法都是预先定义好一个组合方程，并且其中每种方法的重要程度都是一样的。

6. Saliency Detection via Graph-Based Manifold Ranking

方法动机：以往的显著性检测方法大多是通过计算某像素或区域在局部的上下文中或整幅图像中的对比度来得到该像素或区域上的显著性值，这篇论文则是通过计算图像元素（像素或区域）和 foreground cues 以及 background cues 的相似度来得到该图像元素上的显著度，相似度是通过 graph-based manifold ranking 求得的。

background cue：背景通常与图像四个边界呈现局部或整体上的外观关联性

foreground cue：foreground presents appearance coherence and consistency

算法步骤如图 5，先对图像进行超像素分割，然后将分割后的图像映射为图，每个超像素为图中的节点。第一个阶段是将图像的每一条边界上（共 4 条）的节点看作是 labelled background queries，依次算图中的每个节点与这些 queries 的相关性来得到 4 幅 labelled maps，然后对其进行融合得到显著图。在第二阶段，对第一阶段得到的显著图二值化，然后将得到的 labelled foreground 节点看作是 salient queries，最终每个节点的显著度就是通过计算其与 foreground queries 的相关性得到。

第一阶段与第二阶段中的 ranking function 是什么呢？

7. Saliency Optimization from Robust Background Detection

论文的基本步骤如下：

1. 用 SLIC 方法将图像先分割成超像素，对一幅 300×400 的图像，取超像素个数为 200 个足够了。
2. 构造无向图，超像素作为节点，相邻的超像素之间连成边，边的权值用两个超像素区域在 CIE-Lab 颜色空间上的颜色均值的欧几里得距离 $d_{app}(p, q)$ 表示。

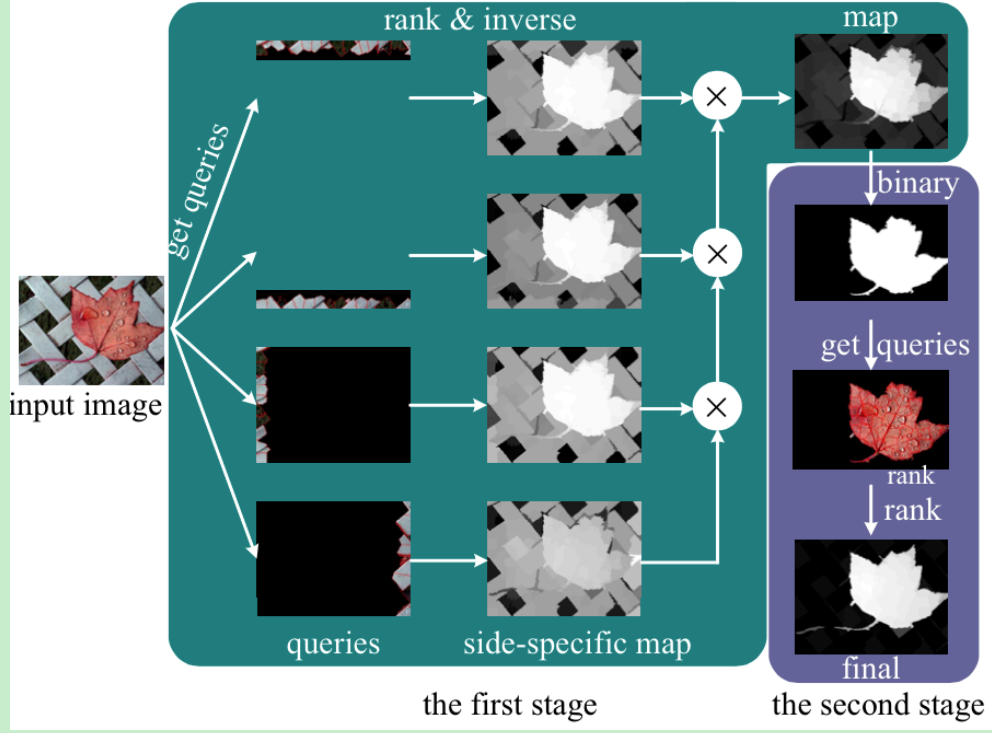


图 5: 算法步骤

两个超像素之间的 deodesic distance 被定义为它们在图上最短路径的累计权值和：

$$d_{geo}(p, q) = \min_{p_1=p, p_2, \dots, p_n=q} \sum_{i=1}^{n-1} d_{app}(p_i, p_{i+1}) \quad (6)$$

为方便体现，定义 $d_{geo}(p, p) = 0$ 。每个超像素 p 的 spanning area 定义为：

$$Area(p) = \sum_{i=1}^N \exp\left(-\frac{d_{geo}^2(p, p_i)}{2\sigma_{clr}^2}\right) = \sum_{i=1}^N S(p, p_i) \quad (7)$$

其中， N 是超像素的个数。可以看到， $S(p, p_i)$ 的范围一直在 $(0, 1]$ 之间，并且表征了超像素 p_i 对 p 的贡献大小。当 p_i 和 p 在一个较平坦的区域内，则有 $d_{geo}(p, p_i) = 0$ ，并且 $S(p, p_i) = 1$ ，保证 p_i 对区域 p 增加了一个单位区域。而当 p_i 和 p 在不同的区域中时，它们的最短路径中至少有一个边满足 $d_{app}(*, *) \geq 3\sigma_{clr}$ ，使得 $S(p, p_i) \approx 0$ ，保证 p_i 不对 p 没有影响。通过实验发现参数 σ_{clr} 在 $[5, 15]$ 之间时效果是稳定的，在作者试验中取 $\sigma_{clr} = 10$ 。

同样地，将 length along the boundary 定义为：

$$Len_{bnd}(p) = \sum_{i=1}^N S(p, p_i) \cdot \delta(p_i \in Bnd) \quad (8)$$

最后，计算 boundary connectivity

$$BndCon(p) = \frac{Len_{bnd}(p)}{\sqrt{Area(p)}} \quad (9)$$

参考文献

- [1] Rafael C Gonzalez. *Digital image processing*. Pearson Education India, 2009.