

显著目标检测 — 文献阅读

朱亚菲

2014 年 10 月 11 日

1 2007CVPR-Learning to Detect A Salient Object

1. 基本思想

先对显著目标进行特征提取，得到多尺度对比度，中央 - 周围对比度，颜色空间分布 3 种 feature maps，再用条件随机场模型进行组合，得到最终的显著性检测结果。

2. 方法流程

- Multi-scale contrast feature: 本文将多尺度对比度特征 $f_c(x, I)$ 定义为高斯图像金字塔中对比度的线性组合：

$$f_c(x, I) = \sum_{l=1}^L \sum_{x' \in N(x)} \|I^l(x) - I^l(x')\|^2$$

其中 I^l 是高斯金字塔第 l 层图像，金字塔的层数 L 取 6， $N(x)$ 为 9×9 的窗。

- Center-surround histogram: 计算像素点 x' 为中心的显著矩形区域与其周围矩形区域的 RGB 颜色直方图之间的 χ^2 距离；

由于目标尺寸不同，选择不同纵横比的矩形区域进行测试， χ^2 距离最大对应得到最独特矩形区域；像素点 x 的中央-周围直方图特征定义为其所属所有矩形区域的空间高斯加权 χ^2 距离之和。

$$f_h(x, I) \propto \sum_{\{x' | x \in R^*(x')\}} w_{xx'} \chi^2(R^*(x'), R_S^*(x'))$$

- Color spatial-distribution: 图像中的所有颜色用高斯混合模型来表示，每一个像素被分配给具有某概率的颜色成分，计算每一个颜色成分的水平方差和垂直方差，得到该成分的空间方差；颜色空间分布特征定义为中央加权的空间方差之和。颜色方差越小，该颜色越有可能属于显著目标。
- 条件随机场结合：能量函数为 K 个显著特征和配对特征的线性组合，通过条件随机场学习计算权重，得到最优化的线性组合。其中显著特征为前面得到的 3 种 feature map，用来描述一个像素点是否属于显著目标；配对特征为两个相邻像素点的空间关系，是对相邻像素标记为不同值的惩罚项。

3. 论文评价

采用条件随机场模型进行特征组合，融合颜色独立性，颜色空间分布和多尺度分布，考虑了局部信息，全局信息以及尺度信息。效果也不错。

ITTI98 论文主要提取亮度、颜色、方向三种特征，得到三种 feature map。随后这些 feature maps 被归一化以便综合，综合方法是简单的相加。随后的很多研究都采取了这样的框架，针对特征提取/特征综合等等不同的阶段分别进行优化。例如这篇：2006NIPS-Graph-based visual saliency. 假定仍采用原先的特征提取方式，但是综合阶段使用的不是线性组合而是马尔科夫随机场，获得了比 Itti 更好的效果。本文采用条件随机场，与马尔

可夫随机场相比，其优点是特征函数可以使用从整幅图像中提取的任意底层或高层特征。

2 2008ICVS-Salient Region Detection and Segmentation

这篇论文提出的算法的思想用其论文的一句话表达就是：

saliency is determined as the local contrast of an image region with respect to its neighborhood at various scales.

具体实现上，用这个公式表示：

$$c_{i,j} = D \left[\left(\frac{1}{N_1} \sum_{p=1}^{N_1} v_p \right), \left(\frac{1}{N_2} \sum_{q=1}^{N_2} v_q \right) \right]$$

这里所说的尺度不是指改变图像的尺度而是改变 R2 区域的尺度，这样就可以得到全分辨率的显著图。

```
1  %-----  
2  % Read image  
3  %-----  
4  img = imread('1.jpg');  
5  dim = size(img);  
6  width = dim(2);height = dim(1);  
7  md = min(width, height);%minimum dimension  
8  %-----  
9  % Perform sRGB to CIE Lab color space conversion (using D65)  
10 %-----  
11 cform = makecform('srgb2lab', 'AdaptedWhitePoint', whitepoint('d65'));  
12 lab = applycform(img,cform);  
13 l = double(lab(:,:,1));
```

```

14 a = double(lab(:,:,2));
15 b = double(lab(:,:,3));
16 %If you have your own RGB2Lab function...
17 %[l a b] = RGB2Lab(gfrgb(:,:,1),gfrgb(:,:,2), gfrgb(:,:,3));
18 %-----
19 %Saliency map computation
20 %-----
21 sm = zeros(height, width);
22 off1 = int32(md/2); off2 = int32(md/4); off3 = int32(md/8);
23 for j = 1:height
24     y11 = max(1,j-off1); y12 = min(j+off1,height);
25     y21 = max(1,j-off2); y22 = min(j+off2,height);
26     y31 = max(1,j-off3); y32 = min(j+off3,height);
27     for k = 1:width
28         x11 = max(1,k-off1); x12 = min(k+off1,width);
29         x21 = max(1,k-off2); x22 = min(k+off2,width);
30         x31 = max(1,k-off3); x32 = min(k+off3,width);
31         lm1 = mean2(l(y11:y12,x11:x12)); am1 = mean2(a(y11:y12,x11:x12)); bm1 = me
32         lm2 = mean2(l(y21:y22,x21:x22)); am2 = mean2(a(y21:y22,x21:x22)); bm2 = me
33         lm3 = mean2(l(y31:y32,x31:x32)); am3 = mean2(a(y31:y32,x31:x32)); bm3 = me
34         %-----
35         % Compute conspicuity values and add to get saliency value.
36         %-----
37         cv1 = (l(j,k)-lm1).^2 + (a(j,k)-am1).^2 + (b(j,k)-bm1).^2;
38         cv2 = (l(j,k)-lm2).^2 + (a(j,k)-am2).^2 + (b(j,k)-bm2).^2;
39         cv3 = (l(j,k)-lm3).^2 + (a(j,k)-am3).^2 + (b(j,k)-bm3).^2;
40         sm(j,k) = cv1 + cv2 + cv3;
41     end
42 end
43
44 imshow(sm, []);
45 %-----

```

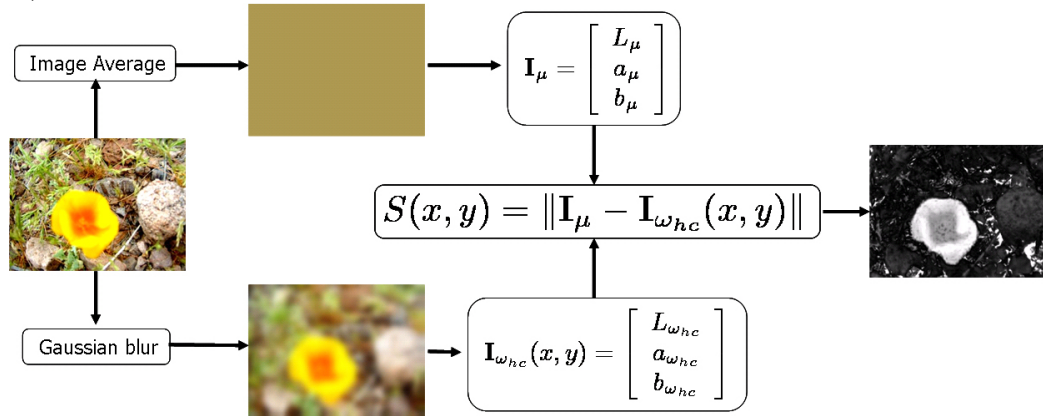
关于这个算法的理论分析，FT 算法那个论文里有这样一段话：

Objects that are smaller than a filter size are detected ompletely, while objects larger than a filter size are only artially detected (closer to edges). Smaller objects that are well detected by the smallest filter are detected by all three filters, while larger objects are only detected by the larger filters. Since the final saliency map is an average of the three feature maps (corresponding to detections of he three filters), small objects will almost always be better highlighted.

3 2009CVPR-Frequency-tuned Salient Region Detection

显著的区域和目标相比其周围背景会比较突出，本文的目的在于计算每个像素在颜色和亮度特征方面相对其领域的显著程度。大多数显著性检测方法都采用了相似的 center-versus-surround 方法，其中关键是用来计算显著性的领域的尺度大小。本文将整幅图像看成是某一像素的领域。这也使得相对现在 state-of-the-art 的方法可以利用更多空间频率。

一般对图像进行滤波会导致只利用了原图的有限范围的频率信息来计算显著图，本文采用 DoG (Difference of Gaussian) 带通滤波器对图像进行滤波，即用两个尺度不同的高斯滤波器对原图像进行滤波后再相减。



这篇文章之后，很多工作的评价标准就从传统的对注视点预测的评价，转移到对物体区域二值图的预测上了。从某种意义上讲，这篇文章对 Saliency detection 的问题做了重新的定义，让问题定义更加回归实际应用。

```
1  img=imread('3.jpg');
2  gfrgb=imfilter(img,fspecial('gaussian',3,3),'symmetric','conv');
3  %-----
4  % Perform sRGB to CIE Lab color space conversion (using D65)
5  %-----
6  cform=makecform('srgb2lab','AdaptedWhitePoint',whitepoint('d65'));
7  lab=applycform(gfrgb,cform);
8  %-----
9  % Compute Lab average values (note that in the paper this
10 % average is found from the unblurred original image, but
11 % the results are quite similar)
12 %-----
13 l=double(lab(:,:,1)); lm=mean(mean(l));
14 a=double(lab(:,:,2)); am=mean(mean(a));
15 b=double(lab(:,:,3)); bm=mean(mean(b));
16 %-----
17 % Finally compute the saliency map and display it.
18 %-----
19 sm=(l-lm).^2+(a-am).^2+(b-bm).^2;
20 imshow(sm,[]);
21 %-----
```

4 2014TPAMI-Global Contrast based Salient Region Detection

本文在基于以下四点考虑的基础上，提出了提取高分辨率的全局显著性图像的分析方法：

- 基于全局对比度的方法倾向于将大范围的目标和周围环境分离开。这种方法要优于那些通常只在轮廓附近产生较高显著性的局部对比度方法。
- 全局的考虑可以为图像中相似区域分配一个相近的显著性的值，并且可以均匀的突出目标。
- 一个区域的显著性，主要是由它和周围区域的对比度决定，相距很远的区域起的作用较小。
- 为了能够适应大规模图像集处理和高效的图像检索与分类的应用需求，显著图检测算法应该具有简单快速的特点。

1) HC：基于直方图对比度的方法，每一个像素的显著性值是由它与图像中所有其他像素的颜色差异来确定，得到全分辨率显著性图像；

$$S(I_k) = D(I_k, I_1) + D(I_k, I_2) + \dots + D(I_k, I_N)$$

2) RC：基于局部对比度的方法，先将图像分割成小区域，采用的分割方法是基于图的分割，基本分割思想是将每个像素点作为无向图的顶点，两个像素点之间的不相似度作为边的权重，要求连接相同区域内的两个顶点的边的最大权重要小于连接不同区域的顶点的边的最小权重，在迭代过程中进行顶点归纳与区域合并，具体参见论文 Efficient graph-based image segmentation；每个区域的显著性值由它与其他所有区域的空间距离和区

域像素数加权的颜色差异来确定；空间距离为两个区域重心的欧氏距离，较远区域分配较小权值；

3) 细节加速：

1 基于直方图的加速：将每个颜色通道由 256 个颜色值量化到 12 个颜色值后，对输入图像计算颜色直方图，保留高频颜色，覆盖 95% 图像像素，剩下颜色舍弃，用直方图中距离最近的颜色代替；

2 颜色空间平滑：量化本身可能会有瑕疵，因为一些相似的颜色可能被量化为不同的值。为减小量化误差，每个颜色的显著性值被替换为相似颜色显著性的加权平均；在 RGB 空间进行量化，用 Lab 空间度量距离；

5 2012CVPR-A Unified Approach to Salient Object Detection via Low Rank Matrix Recovery

本文主要是做矩阵分解，根据论文 Visual saliency detection via sparsity pursuit 改进。

1、一幅图像可以表示成一个低秩矩阵与稀疏噪声的和，其中非显著的区域可以用低秩矩阵解释，而用稀疏噪声表示显著区域，本文不是第一次这样表示，在论文 Visual saliency detection via sparsity pursuit 中已采用过，其中，先将一幅图像通过 uniformly sampling 分解成 8×8 的 patch，对每一个 patch 用超完备基进行稀疏编码，这些稀疏编码向量按列排列可形成一个矩阵。但这篇论文存在两个问题：(1) 对于显著性检测而言，稀疏编码并不是非常好的特征表示方法，因为对于每个 patch 的稀疏编码是与噪声的稀疏性（显著性）无关的。经过 uniformly sampling 的 patches 仅仅由位置决定，有些 sampled patches 可能既包含背景又包含显著目标，因此不能保证背景矩阵一定是低秩的。(2) 当显著目标较大时，这个显著目标

内部可能被分成很多 patch，可能两个 patch 对应的特征向量是相似的，这样用来表示显著性的噪声就可能不是稀疏的了。基于此，本文采用的方法是经过多尺度特征提取，并考虑空间信息，然后用 mean-shift clustering 方法将一幅图像分解成一些小区域（参照论文 Mean shift: A robust approach toward feature space analysis）。每个区域中所有像素的特征向量的均值可看作其特征向量，将所有区域的特征向量按列排列可得到这幅图像的矩阵表示。这样改进的优点：(1) 即使显著目标尺寸很大，其内部的区域分割数目也会很小，因为显著目标在空间位置和外观上都有连贯性。(2) 本文还通过标记数据训练得到一个线性特征变换来确保表示背景的矩阵在我们所学习的特征空间是低秩的。

2、论文 Visual saliency detection via sparsity pursuit 中用稀疏编码作来表示特征，本文用 53 个特征组成的特征向量来表示。

3、不是所有 unique 的区域都是显著的，因而需要高级先验知识来指导。在论文 Context-aware saliency detection 和论文 Learning to predict where humans look 中已经加入了这些高级先验知识，但其融合是对 saliency map 的后处理，也就是对由 higher level cues 产生的 saliency maps 与由 low level features 产生的 saliency maps 作加权平均，得到最终的 saliency map。因而需要一种更有效的融合方法。

1) 基本流程：文章提出了一种新的图像表示方法，将其表示为一个低秩矩阵（非显著区域）加上稀疏噪音（显著区域），再利用 Robust PCA 技术进行低秩矩阵恢复，得到的噪音就是显著区域，再根据高层次的先验知识来帮助修正显著区域。

2) 图像矩阵：

提取特征：R, G, B, hue, saturation, 3 尺度下 4 个方向共 12 个 steerable pyramids 响应，3 尺度 8 方向共 12 个 Gabor filters 响应，加起来一共 53 维。

矩阵构造：先利用 Mean-shift 算法将图像分割成很多较小的 segments

再用每个 segment 中所有特征向量的均值来表示这个 segment，从而构造成为矩阵。

特征空间变换：保证特征向量为低秩。

3) 高层先验融合：位置先验（基于图像中心高斯分布），语义先验（人脸检测），颜色先验（暖色更明显）