显著性检测领域之 feature contrasts 和 priors

朱亚菲

2015 年 1 月

目录

1. 引言

1.1 显著性检测

人类视觉系统对复杂场景具有强适应性，神经生理和心理学研究表明人眼面临复杂场景时，会迅速将注意力集中在少数重要区域，并利用有限的处理能力对其优先处理。图像显著性检测的目的是在图像中快速有效地找到重要并且信息量大的区域 [15]。目前在图像重定向、图像分类、图像分割等领域都有广泛的应用。

目前大多数显著性检测方法都是基于自底向上的计算框架。归纳起来通常分为三步：1. 特征提取，提取多种视觉特征，例如亮度、颜色、纹理等。2. 求像素或区域间的对比度。3. 求显著图

1.2 features

图像特征提取是图像分析与图像识别的前提，它是将高维的图像数据进行简化表达最有效的方式，从一幅图像的 $M \times N \times 3$ 的数据矩阵中，我们看不出任何信息，所以我们必须根据这些数据提取出图像中的关键信息，一些基本元件以及它们的关系。

图像特征可分为全局特征和局部特征[1]。其最大的区别是特征提取的空间范围不同。全局特征是从整个图像中提取的特征，而局部特征是从图像区域中提取的特征。全局特征容易受到环境的干扰，光照、旋转、噪声等不利因素都会影响全局特征。相比而言，局部特征点，往往对应着图像中的一些线条交叉、明暗变化的结构中，受到的干扰也少。

总的来说，全局特征是对图像内容的高度抽象的概括。如果用户对整个图像的整体感兴趣，而不是对前景本身感兴趣的话，用全局特征来描述图像是比较合适的。但是无法分辨出前景和背景却是全局特征本身就有的劣势，特别是在关注的对象受到遮挡等影响的时候，全局特征很有可能就被破坏掉了。

在显著性检测中，由于关注的是显著目标，并且是在同一幅图像中进行中央 -周围/局部/全局对比，而不是在图像间进行比较，因此用到的应该都是局部特征。显著性检测领域所说的局部和全局方法是指对某一像素或区域，在计算其对比度时是与周围相比还是与图像中所有其它像素或区域相比。

英国 Oxford 大学的 Andrea Valida，他是 VLFeat(Vision Lab Features Library) 的发起者和主要作者。VLFeat 官方主页：vlfeat.org。它一个实现了计算机视觉领域诸多算法的开源库，包括 SIFT，HOG 等等。底层代码用 C 语言实现，并提供了 MATLAB 接口。支持 Windows，Mac OS X 和 Linux。VLFeat 目前正在逐渐实现其他常用的特征描述子。和 OpenCV 相比，VLFeat 是一个轻量级的库，主要实现了在特征提取和聚类方面的高效算法，可以用在图像检索和物体识别领域中。

图像特征又可以分为低层次特征和高层次特征 [24]。其中低层次特征是不需要任何形状信息 (空间关系的信息) 就可以从图像中自动提取的基本特征。高层次特征提取关心的是在图像中找出形状。例如，要自动识别人脸，一种方法是提取组成部分特征。也就是说，需要提取眼睛、耳朵和鼻子这些主要的脸部特征。这些特征可以利用它们的形状找到：眼睛的白色部分是椭圆形的；嘴巴可以看做是两条直线，眉毛也一样。形状提取意味着找出它们的位置、朝向和尺寸。所有低层次方法都可以应用于高层次特征提取，从而在图像中找到形状。(这里将特征分为低层次特征和高层次特征是是从广义还是狭义的角度我还没有搞清楚)

1.3 feature contrasts

对于一幅图像，人们会更加关注与周围物体的对比度差异大的区域。

[1] 更多局部特征见http://www.sigvc.org/bbs/thread-165-1-1.html

1.4 color space

一般显著性检测方法中都会用到颜色特征。颜色通常用三个 (也可以更多或更少) 相对独立的属性来描述，三个独立变量综合作用，自然就构成一个空间坐标，这就是颜色空间。而颜色可以由不同的角度，用不同属性加以描述，就产生了不同的颜色空间。但被描述的颜色对象本身是客观的，不同颜色空间只是从不同的角度去衡量同一个对象。除了 RGB 颜色空间，另外两个常用的颜色空间就是 CIELab 和 HSV 颜色空间。

颜色空间按照基本结构可以分两大类：基色颜色空间和色、亮分离颜色空间。前者的典型是 RGB，还包括 CMY、CMYK、CIE XYZ 等；后者包括 YCC/YUV、Lab，以及一批“色相类颜色空间”。CIE XYZ 是定义一切颜色空间的基准，很奇妙的是，它即属于基色颜色空间，也属于色、亮分离颜色空间，是贯穿两者的枢纽。色、亮分离颜色空间中的子类型”色相类颜色空间“，是把颜色分成一个表亮属性和两个表色属性，其中有一个表色属性是色相，而色相以外的两个属性可以选用不同的变量来定义，色相的概念不变，因此就构成一族共同使用色相属性，另加表亮属性和表色属性各一个组成的颜色空间，它们是颜色空间中的一个家族，暂且统称为 HSB 颜色空间。

1.4.1 RGB 颜色空间

RGB 颜色空间是常用的表示彩色图像的一种颜色空间，它是以红、绿、蓝三种颜色为基础，亦称为“三原色”。所谓的“原色”是一种生物学概念，是根据人眼对光线感知的生理作用来定义的。每一种颜色按亮度进行分类，分成 256 个等级。不同比例的红、绿、蓝叠加，能产生丰富的颜色。例如，等比例的三原色进行相加可以产生白色，红色与绿色相加产生黄色。可见，RGB 空间属于“叠加型”原色系统，因此把 RGB 颜色空间作为最基础的颜色空间，通过对 RGB 的非线性或线性变换可以获得其它的颜色空间。

RGB 颜色空间最常用的用途就是显示器系统，彩色阴极射线管、彩色光栅图形的显示器都使用 R、G、B 数值来驱动 R、G、B 电子枪发射电子，并分别激发荧光屏上的 R、G、B 三种颜色的荧光粉发出不同亮度的光线，并通过相加混合产生各种颜色；扫描仪也是通过吸收原稿经反射或透射而发送来的光线中的 R、G、B 成分，并用它来表示原稿的颜色。RGB 色彩空间称为与设备相关的色彩空间，因为不同的扫描仪扫描同一幅图像，会得到不同色彩的图像数据；不同型号的显示器显示同一幅图像，也会有不同的色彩显示结果。

1.4.2 YUV 颜色空间

在现代彩色电视系统中，通常采用三管彩色摄像机或彩色 CCD(点耦合器件) 摄像机，它把摄得的彩色图像信号经分色分别放大校正得到 RGB，再经过矩阵变换电路得到亮度信号 Y 和两个色差信号

$R-Y$、$B-Y$，最后发送端将亮度和色差三个信号分别进行编码，用同一信道发送出去。这就是我们常用的 YUV 色彩空间。采用 YUV 色彩空间的重要性是它的亮度信号 Y 和色度信号 U、V 是分离的。如果只有 Y 信号分量而没有 U、V 分量，那么这样表示的图就是黑白灰度图。彩色电视采用 YUV 空间正是为了用亮度信号 Y 解决彩色电视机与黑白电视机的兼容问题，使黑白电视机也能接收彩色信号。根据美国国家电视制式委员会，NTSC 制式的标准，当白光的亮度用 Y 来表示时，它和红、绿、蓝三色光的关系可用如下式的方程描述：$Y = 0.3R + 0.59G + 0.11B$，这就是常用的亮度公式。色差 U、V 是由 BY、RY 按不同比例压缩而成的。如果要由 YUV 空间转化成 RGB 空间，只要进行相反的逆运算即可。

1.4.3 CIEXYZ 颜色空间

国际照明委员会 (CIE) 在进行了大量正常人视觉测量和统计的基础上，与 1931 年建立了“标准色度观察者”，从而奠定了现代 CIE 标准色度学的定量基础。在色彩管理中，选择与设备无关的颜色空间是十分重要的，与设备无关的颜色空间由国际照明委员会 (CIE) 制定，包括 CIEXYZ 和 CIELAB 两个标准。它们包含了人眼所能辨别的全部颜色。

1.4.4 CIELUV 颜色空间

论文 [21] 中用到。

LUV 色彩空间全称 CIE 1976(L*,u*,v*)(也作 CIELUV) 色彩空间，L* 表示物体亮度，u* 和 v* 是色度。于 1976 年由国际照明委员会 (International commission on Illumination) 提出，由 CIE XYZ 空间经简单变换得到，具视觉统一性。类似的色彩空间有 CIELAB。对于一般的图像，u* 和 v* 的取值范围为 -100 到 +100，亮度为 0 到 100。

1.4.5 CIELAB 颜色空间

Lab 模式是由国际照明委员会 (CIE) 于 1976 年公布的一种色彩模式。它既不依赖光线，也不依赖于颜料，是 CIE 组织确定的一个理论上包括了人眼可以看见的所有色彩的色彩模式。CIELab 颜色模型基于人对颜色的感觉，是一种接近人类视觉的颜色系统。Lab 中的数值描述正常视力的人能够看到的所有颜色。因为 Lab 描述的是颜色的显示方式，而不是设备 (如显示器、桌面打印机或数码相机) 生成颜色所需的特定色料的数量，所以 Lab 被视为与设备无关的颜色模型。CIELab 颜色空间是 CIE XYZ 颜色空间的一种数学变换的结果。

CIELAB 系统使用的坐标叫做对色坐标 (opponent color coordinate)，使用对色坐标的想法来自这样的概念：颜色不能同时是红和绿，或者同时是黄和蓝，但颜色可以被认为是红和黄、红和蓝、绿和黄以及绿和蓝的组合。CIELAB 使用 L, a 和 b 坐标轴定义 CIE 颜色空间。其中，L 值代表光亮度，其

值从 0(黑色) 100(白色)。a 和 b 代表色度坐标，其中 a 代表红 -绿轴，b 代表黄 -蓝轴，取值范围是 $[127, -128]$，如图 1。

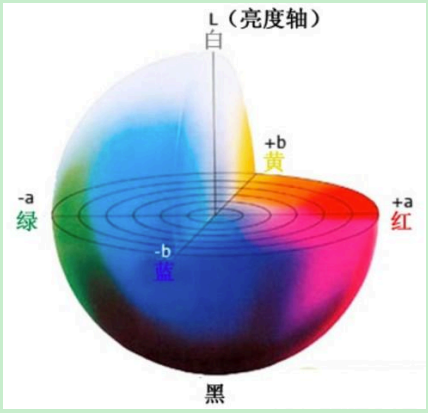


图 1: Lab 颜色模型

matlab 中由 RGB 空间转换到 Lab 空间的函数如下：

```
function [L,a,b] = rgb2lab(R,G,B)
% function [L, a, b] = RGB2Lab(R, G, B)
% RGB2Lab takes matrices corresponding to Red, Green, and Blue, and
% transforms them into CIELab.  This transform is based on ITU-R
% Recommendation  BT.709 using the D65 white point reference.
% The error in transforming RGB -> Lab -> RGB is approximately
% 10^-5.  RGB values can be either between 0 and 1 or between 0 and 255.
% By Mark Ruzon from C code by Yossi Rubner, 23 September 1997.
% Updated for MATLAB 5 28 January 1998.

if (nargin == 1)
  B = double(R(:,:,3));
  G = double(R(:,:,2));
  R = double(R(:,:,1));
end

if ((max(max(R)) > 1.0) | (max(max(G)) > 1.0) | (max(max(B)) > 1.0))
  R = R/255;
  G = G/255;
  B = B/255;
end

```

```
[M, N] = size(R);
s = M*N;

% Set a threshold
T = 0.008856;

RGB = [reshape(R,1,s); reshape(G,1,s); reshape(B,1,s)];

% RGB to XYZ
MAT = [0.412453 0.357580 0.180423;
       0.212671 0.715160 0.072169;
       0.019334 0.119193 0.950227];
XYZ = MAT * RGB;

X = XYZ(1,:) / 0.950456;
Y = XYZ(2,:);
Z = XYZ(3,:) / 1.088754;

XT = X > T;
YT = Y > T;
ZT = Z > T;

fX = XT .* X.^(1/3) + (¬XT) .* (7.787 .* X + 16/116);

% Compute L
Y3 = Y.^(1/3);
fY = YT .* Y3 + (¬YT) .* (7.787 .* Y + 16/116);
L  = YT .* (116 * Y3 - 16.0) + (¬YT) .* (903.3 * Y);

fZ = ZT .* Z.^(1/3) + (¬ZT) .* (7.787 .* Z + 16/116);

% Compute a and b
a = 500 * (fX - fY);
b = 200 * (fY - fZ);

L = reshape(L, M, N);
a = reshape(a, M, N);
b = reshape(b, M, N);
```

```
61
62 if ((nargout == 1) | (nargout == 0))
63   L = cat(3,L,a,b);
64 end
```

1.4.6 HSV 颜色模型

HSV(Hue, Saturation, Value) 是根据颜色的直观特性由 A. R. Smith 在 1978 年创建的一种颜色空间，也称六角椎体模型 (Hexcone Model)。

HSV 颜色模式是除了 RGB 颜色模式之外的另一种流行的颜色模式，RGB 被广泛运用于计算机中，而 HSV 则用在电视显示方面。它更符合人们对颜色的描述 (什么颜色 (H)，深浅度如何 (S)，亮度如何 (V))。其实在电视机上菜单中的饱和度就是 S，亮度就是 V。

如图 2，色调 (H) 用与水平轴之间的角度来表示，范围从 0 度到 360 度。六边形的顶点以 60 度为间隔。黄色位于 60 度处，绿色在 120 度处而青色在 150 度处，与红色相对。相补的颜色互成 180 度。

饱和度 (S) 从 0 到 1 变化。在此模型中它表示成所选色彩的纯度与该色彩的最大纯度 ($S = 1$) 的比率。当 $S = 0,5$ 时所选色彩的纯度为四分之一。当 S=0 时，只有灰度。

亮度值 (V) 从六边形顶点的 0 变化到顶部的 1，顶点表示白色。在六边形顶部的颜色强度最大。当 $V = 1$，$S = 1$ 时，有纯色彩。白色是 $V = 1$ 且 $S = 0$ 的点。

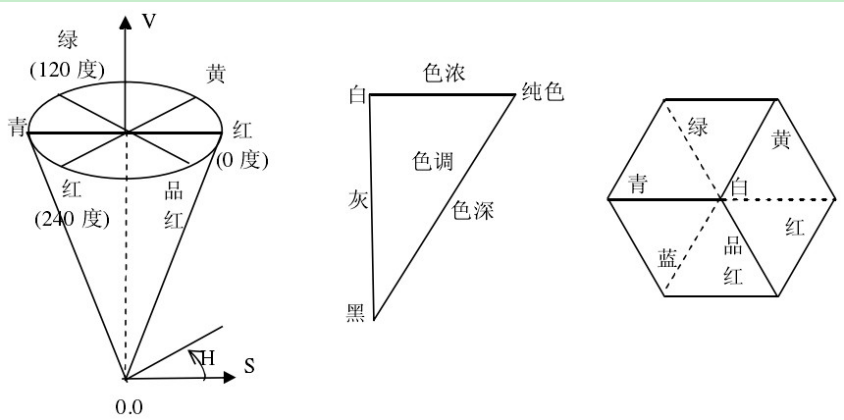


图 2: HSV 颜色模型

matlab 自带的 rgb2hsv 函数如下：

```
1 function [h,s,v] = rgb2hsv(r,g,b)
2 switch nargin
3   case 1,
```

```
4       if isa(r, 'uint8'),
5          r = double(r) / 255;
6       elseif isa(r, 'uint16')
7          r = double(r) / 65535;
8       end
9    case 3,
10      if isa(r, 'uint8'),
11         r = double(r) / 255;
12      elseif isa(r, 'uint16')
13         r = double(r) / 65535;
14      end
15
16      if isa(g, 'uint8'),
17         g = double(g) / 255;
18      elseif isa(g, 'uint16')
19         g = double(g) / 65535;
20      end
21
22      if isa(b, 'uint8'),
23         b = double(b) / 255;
24      elseif isa(b, 'uint16')
25         b = double(b) / 65535;
26      end
27
28   otherwise,
29       error(message('MATLAB:rgb2hsv:WrongInputNum'));
30 end
31
32 threeD = (ndims(r)==3); % Determine if input includes a 3-D array
33
34 if threeD,
35   g = r(:,:,2); b = r(:,:,3); r = r(:,:,1);
36   siz = size(r);
37   r = r(:); g = g(:); b = b(:);
38 elseif nargin==1,
39   g = r(:,2); b = r(:,3); r = r(:,1);
40   siz = size(r);
41 else
```

```
42    if ¬isequal(size(r),size(g),size(b)),
43      error(message('MATLAB:rgb2hsv:InputSizeMismatch'));
44    end
45    siz = size(r);
46    r = r(:); g = g(:); b = b(:);
47  end
48
49  v = max(max(r,g),b);
50  h = zeros(size(v));
51  s = (v - min(min(r,g),b));
52
53  z = ¬s;
54  s = s + z;
55  k = find(r == v);
56  h(k) = (g(k) - b(k))./s(k);
57  k = find(g == v);
58  h(k) = 2 + (b(k) - r(k))./s(k);
59  k = find(b == v);
60  h(k) = 4 + (r(k) - g(k))./s(k);
61  h = h/6;
62  k = find(h < 0);
63  h(k) = h(k) + 1;
64  h=(¬z).*h;
65
66  k = find(v);
67  s(k) = (¬z(k)).*s(k)./v(k);
68  s(¬v) = 0;
69
70  if nargout≤1,
71    if (threeD || nargin==3),
72      h = reshape(h,siz);
73      s = reshape(s,siz);
74      v = reshape(v,siz);
75      h=cat(3,h,s,v);
76    else
77      h=[h s v];
78    end
79  else
```

```
80    h = reshape(h,siz);
81    s = reshape(s,siz);
82    v = reshape(v,siz);
83  end
```

总结：可以看到 YUV/YIQ 颜色空间是由 RGB 空间线性变换而来，而 CIELab 和 HSV 颜色空间则是由 RGB 空间非线性变换而来 [13]。

2. 基于像素点的 feature contrasts

2.1 亮度

Itti1998 模型 [8] 中用到。亮度信息是最基本的视觉信息，它直接由视网膜的视杆细胞产生, 是最基本的视觉信息，视觉系统中的其他信息很多也是由亮度而来。而从计算的角度考虑，亮度特征也足够用于多数图像处理任务，比如分割、识别，同时计算量远小于彩色特征。一些特殊成像方法得到的图像没有颜色信息，直接利用获取的亮度信息。

设 r,g,b 为输入图像的红色、绿色、蓝色通道，则 Itti 原始模型中亮度特征可以简单的计算出来:

$$I = \frac{r+g+b}{3} \tag{1}$$

2.2 颜色

Itti1998 模型 [8] 中用到。颜色信息由视网膜的三种视锥细胞分别产生，而在初级视觉皮层中的颜色柱进行重新组合。颜色柱中的神经细胞对颜色的选择呈现双拮抗性质，即一种颜色使得细胞产生兴奋，而另一种颜色使得能够抑制该细胞的兴奋。在人类视觉皮层中存在四种拮抗颜色对，分别是红-绿、绿-红、蓝-黄、黄-蓝。

首先为了去除颜色与亮度之间的耦合关系, 用之前计算的亮度值对进行规一化处理。又因为视锥细胞只能在明亮的光线下感受颜色，所以只对亮度大于图像中最大亮度 10% 的区域进行规一化，而其他区域的均置为 0。

数字图像一般只有红、绿、蓝三个颜色通道，为了得到黄色通道，Itti 根据归一化后的, 使用下面

的公式计算广义上的红、绿、蓝、黄四个通道:

$$R = r - \frac{g+b}{2} \tag{2}$$

$$G = r - \frac{r+b}{2} \tag{3}$$

$$B = r - \frac{r+g}{2} \tag{4}$$

接下来利用这四个通道计算红 -绿、蓝 -黄颜色对。Itti 模型巧妙地利用绝对值，表示了两个相反的颜色对中兴奋的那一种。因此红 -绿与绿 -红颜色对可以用一个公式表示:

$$RG = |R - G| \tag{5}$$

$$BY = |B - Y| \tag{6}$$

2.3 方向

Itti1998 模型 [8] 中用到。初级视觉皮层细胞对特定方向的刺激有强烈反应。Dennis Gabor 经过研究发现，二维 Gabor 滤波器非常适合表示这种反应。二维 Gabor 滤波器是一种用于检测边缘的线性滤波器，由高斯核函数与一个余弦函数调制得到:

$$g(x, y; \lambda, \theta, \psi, \delta, \gamma) = exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\delta^2}\right) sin(\frac{2\pi x'}{\lambda} + \psi) \tag{7}$$

其中 $x' = xcos\theta + ysin\theta$，$y' = -xsin\theta + ycos\theta$。公式中 λ 表示余弦函数的波长，θ 表示 Gabor 滤波器的方向，ψ 表示相位，γ 表示空间长宽比，δ 表示高斯包络的标准差。

由于初级视觉皮层中有几千万个细胞，无法用模型将它们全部表示出来，Itti 对这种生理结构进行了适当的简化。通过比较生理实验结果与模型的结果，固定次要变量，设置 $= 7$，$\psi = 0$，$\gamma = 1$，$\delta = 2.333$。而主要的变量是 θ，对应着 Gabor 滤波器的方向,Itti 将连续的方向离散化, 选择了 4 个最有代表性的方向：0°，45°，90°，135°，这样就构造出 4 个 Gabor 滤波器，分别对输入图像滤波，得到 4 个方向特征图，记作 $O(\theta)$，其中 $\theta = 0°$，45°，90°，135°。

2.4 SIFT 特征

SIFT 特征 (Scale-invariant transform，尺度不变特征变换) 由 David Lowe 在 1999 年 [17] 所发表，2004 年 [18] 完善总结。它是一种计算机视觉的算法，用来侦测与描述图像中的局部性特征，它在空间尺度中寻找极值点，并提取出其位置、尺度、旋转不变量。SIFT 算法提取的特征点具有尺度不变性，

也就是说，同一物体在图像上不论尺度大小，都能根据 SIFT 算法提取到相同的特征点。

简单来说，SIFT 算法就是用不同尺度 (标准差) 的高斯函数对图像进行平滑，然后比较平滑后图像的差别，差别大的像素就是特征明显的点。

尺度空间理论的目的是模拟图像数据的多尺度特征。高斯卷积核是实现尺度变换的唯一线性核，于是一幅二维图像的尺度空间定义为：

$$L(x,y,e) = G(x,y,e) * I(x,y) \tag{8}$$

其中 $G(x,y,e)$ 是尺度可变高斯函数，

$$G(x,y,e) = \frac{1}{2*\pi*e^2} * exp[-\frac{x^2+y^2}{2e^2}] \tag{9}$$

(x,y) 是空间坐标，e 是尺度坐标。

为了有效地在尺度空间检测到稳定的关键点，提出了高斯差分尺度空间 (DOG scale-space)。利用不同尺度的高斯差分核与图像卷积生成。

$$D(x,y,e) = ((G(x,y,ke) - G(x,y,e)) * I(x,y) = L(x,y,ke) - L(x,y,e) \tag{10}$$

SIFT 特征一般用于匹配，在 eye fixation prediction 中有用到，而在显著性区域检测中没有见用的。

3. 基于区域的 feature contrasts

这里的 feature contrasts 是指由某种 visual cue(比如颜色、纹理、位置等) 对图像中两个区域 r_i、r_j 求出的区域间对比度。

基于区域的显著性检测方法一般会先对图像进行超像素分割，然后提取超像素区域内的特征，再根据特征计算区域间的对比度，然后通过一定的机制求出最终的显著图。

疑问：用到超像素分割，一定会将其映射为图吗？为什么要将超像素分割结果映射为图 (即每个超像素被看成是图中的节点，节点之间的边的权重为两个超像素区域的相似程度)？

应该是要用到图论中的一些概念 (比如最短路径等) 时会将其映射为图。

经统计，PBS [30]、RBD [34]、OptSeedProp [20]、MR [31]、论文 [23]、PDE [16]、论文 [29] 中都涉及图的概念。

HDCT [13] 中只将图像分割成超像素，没有涉及图的概念。(没有统计完)

3.1 颜色

3.1.1 Color Contrast

区域间的颜色对比度一般用两个超像素区域内颜色均值向量的距离来计算，不同方法中会乘上不同的权重 (例如区域面积、区域间的距离等)。

论文 [29] 中用到，公式如下：

$$C_i = \sum_{j=1}^{n} w(R_i)\Phi(i,j)||c_i - c_j||_2 \tag{11}$$

其中，c_i 和 c_j 分别是区域 R_i 和 R_j 内颜色的均值，$w(R_j)$ 表示区域 R_j 内的像素个数，$\Phi(i,j) = exp\{-D(R_i,R_j)/\sigma^2\}$。

论文 [33] 中用到，图像区域中方差不同的颜色之间的分布是相互独立的。区域 r_i 与 r_j 之间的 color contrast 定义如下：

$$D_c(r_i,r_j) = ||\mu_{c,i} - \mu_{c,j}||^2 \cdot \left(\frac{\sigma_{c,i}^2}{n_i} + \frac{\sigma_{c,j}^2}{n_j}\right)^{-\frac{1}{2}} \tag{12}$$

其中 $\mu_{c,i}$ 和 $\mu_{c,j}$ 分别表示区域 r_i 和 r_j 内的颜色均值，$\sigma_{c,i}^2$ 和 $\sigma_{c,j}^2$ 表示方差，n_i 和 n_j 表示相应区域内像素的个数。

3.1.2 颜色直方图

图像直方图是指统计图像中像素的灰度/颜色得到的图像灰度/颜色频数图。直方图由于其计算代价较小，且具有图像平移、旋转、缩放不变性等优点，广泛应用于图像处理的各个领域。Swain 和 Ballard 最先提出了使用颜色直方图作为图像颜色特征的表示方法。

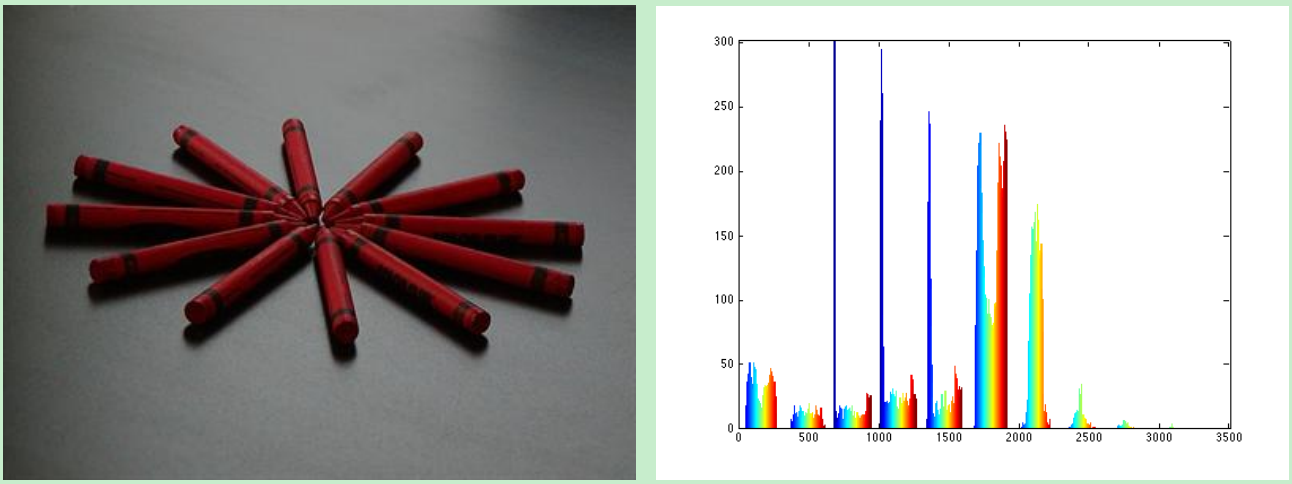
传统颜色直方图描述方法存在以下问题：

1）颜色特征维数高。以 8bit 的 RGB 颜色空间为例，全颜色数为 $256 \times 256 \times 256$ 种颜色，如果以全颜色数统计直方图，则存储空间和计算复杂度都较大。

2）颜色特征受光照影响。即对于两幅颜色分布很类似却因光照不同导致亮度差异大的图像，理论上，其颜色直方图应相似，但实际传统颜色直方图却不相似。

3）不能表达相近颜色间相关性，即传统颜色直方图的颜色间完全独立，不能反映相近颜色间的关联。理论上，对于发生较小颜色偏移的两幅图像间应相似。如，一幅完全红色的图像与另一幅完全浅红色的图像间相似度较高。而实际传统颜色直方图却不相似。

4）丢失空间位置信息，因此该特征无法区分颜色相同而空间分布不同的两幅图像。



(a) (b)

图 3: RGB 空间的颜色直方图

得到图像颜色特征后需要定义颜色特征的相似度量公式，以表示两幅图像间颜色的相似性。不同的相似性度量公式对实际应用结果可能影响很大。因此需要研究如何选择或设计合适的相似性度量算法。

显著性 Models 中，CB [10]、DRFI [11]、HC/RC [4]、HDCT [13] 方法都用到了颜色直方图。

DRFI 方法中关于图像 RGB 空间的颜色直方图代码如下：

```
image = imread('3.jpg');
image_rgb = im2double( image );
RGB_bins = [16 16 16];
R = image_rgb(:,:,1);
G = image_rgb(:,:,2);
B = image_rgb(:,:,3);
rr = min( floor(R*RGB_bins(1)) + 1, RGB_bins(1) );
gg = min( floor(G*RGB_bins(2)) + 1, RGB_bins(2) );
bb = min( floor(B*RGB_bins(3)) + 1, RGB_bins(3) );
Q_rgb = (rr-1) * RGB_bins(2) * RGB_bins(3) + ...
    (gg-1) * RGB_bins(3) + ...
    bb + 1;
```

首先对图像 (300×400) 的颜色空间进行量化，将颜色空间划分为若干个小的颜色区间，即直方图的 bin，例如将每个颜色通道量化为只有 16 个不同值，此时 $bin = 16 \times 16 \times 16$，然后计算矩阵 $Q(300 \times 400)$，用其中的值代表颜色，而不是用 (r, g, b) 向量表示颜色，Q 中有多少个不同值表示图像中有多少种颜色。结果如图 3。

对每个超像素区域，可以知道其中像素点的坐标值，能求出其对应到矩阵 Q 中的值，有多少个不同值就代表该区域内有多少种不同的颜色，然后算该区域在 $1-16^3$ 之间的颜色占的像素个数，没有的记为 0。对每个区域都能算出这样一个 16^3 维的向量，然后求区域之间的直方图的对比度，也就是求这样两个向量之间的距离。

PISA [26] 中没有将图像先分割成超像素，求超像素区域间的对比度，但也不是逐像素地计算 color contrast。对每个像素点 p，采用 CLMF 方法 [19] 构造一个 shape-adaptive observation region Ω_p(如图 4)，然后用该区域内的所有像素点 $q \in \Omega_p$ 计算颜色直方图 $h^c(p)$。这里用到的是 Lab 颜色空间，对每个颜色通道量化至 12 bins，得到的像素点 p 上的颜色直方图 $\mathbf{h}^c(p)$ 则是一个 36 维的描述子，如图 4。

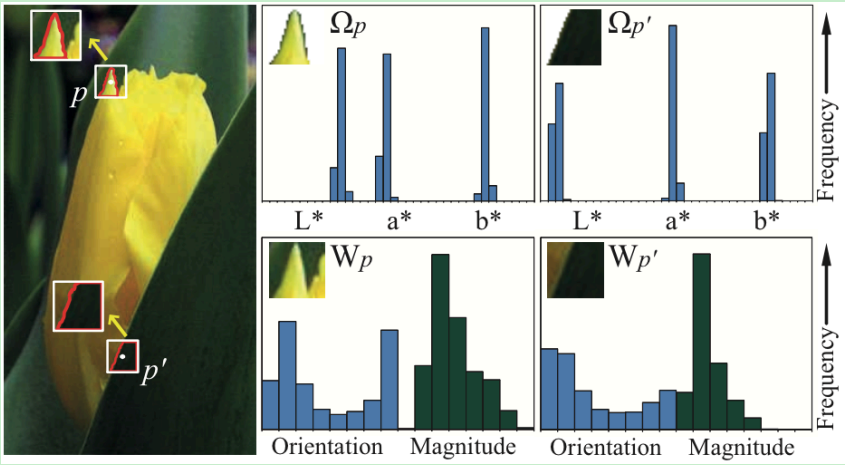


图 4: PISA 示例

接着，用 k-均值方法将拥有相似颜色直方图的像素聚类到一起。对整幅图像 I，其颜色特征空间被量化为 K_c 个聚类 $\{\Phi_1, \ldots, \Phi_{K_c}\}$。像素上的 rarity 或 contrast 计算就可以用聚类间的 rarity 来近似。假设像素点 p 属于聚类 Φ_i，则该点的颜色对比度定义如下：

$$U^c(p) = U^c(\mathbf{h}^c(p)) = \sum_{j=1}^{K_c} w_j ||\mathbf{h}^c(\Phi_i), \mathbf{h}^c(\Phi_j)|| \tag{13}$$

其中 w_j 表示属于聚类 Φ_j 的像素个数，$\mathbf{h}^c(\Phi_i)$ 表示聚类 Φ_i 内的颜色直方图的均值。

3.2 纹理特征

图像纹理一直到现在都没有一个一致的、公认的定义，它在图像中是一个重要但是又不太容易描述出来的特征。纹理是人们将人类的视觉与触觉联系起来，进而形成一个视觉信息，它起源于人类对事物的触感。

LBP(Local Binary Pattern，局部二值模式) 首先是由 Ojala 等人 [25] 于 1994 年提出，DRFI 方法 [11] 中用到。

LBP 有很多变种，或说改进。原始的 LBP 记录像素点与其周围像素点的对比信息，或说差异。对于图像上 9 个方格中中间方格 (方格中的值是像素点灰度值大小)，做一个阈值化处理。大于等于中心点像素的，标记为 1，小于的则标记为 0。最后将中心像素点周围的 11110001 二进制数化为十进制数，得到 LBP 值。如图 5所示。

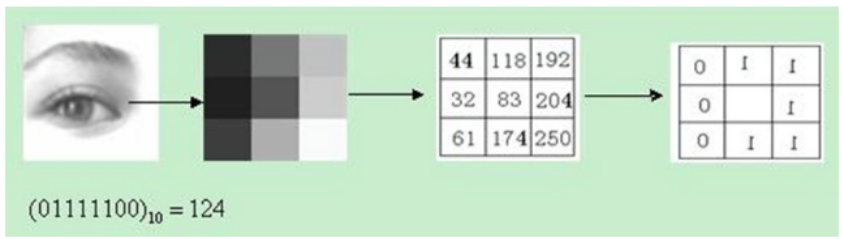


图 5: 原始 LBP

实验效果如图 6。

3.3 Location Contrast

超像素区域中心与图像中心的距离。每个区域的 location contrast 只与图像中心有关，而与其它区域没有关系。

论文 [29] 中用到，公式如下：

$$H_i = \frac{1}{w(R_i)} \sum_{x_i \in R_i} exp\{-\lambda ||x_i - x_c||^2\} \tag{14}$$

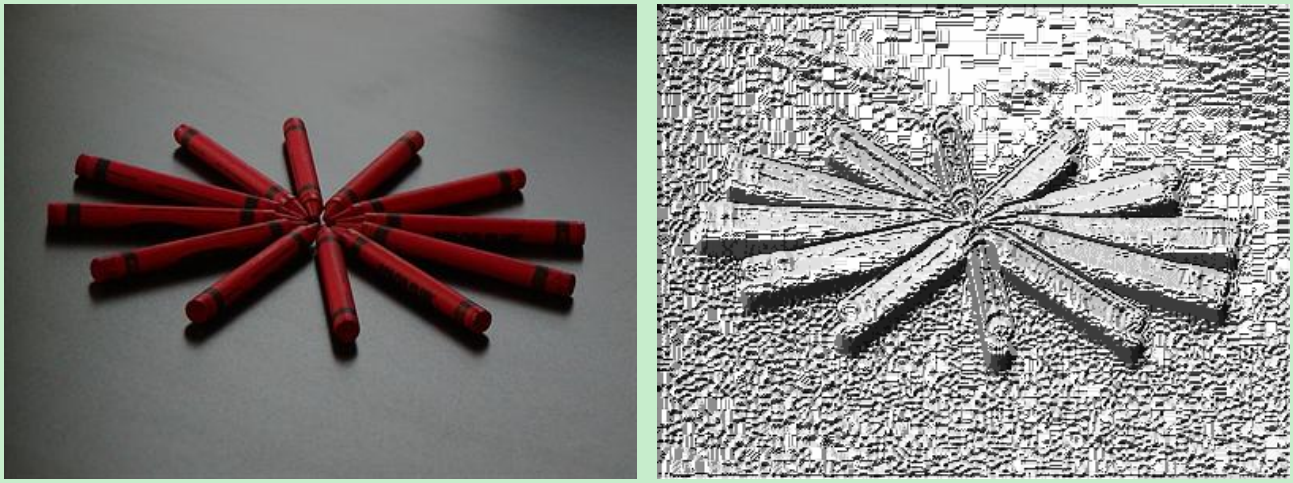
其中 $(x_0, x_1 \ldots)$ 是区域 R_i 中的像素坐标集，x_c 是图像中心的坐标，$w(R_i)$ 计算了区域 R_i 内的像素个数。由 H_i 的公式可看到，距离图像中心越近的区域拥有越大的权值。

3.4 奇异值特征

1、HDCT [13] 中用到。奇异值特征 (Singular Value Feature，SVF) [27] 被用来从测试图像中检测模糊区域，通常一幅图像中的模糊区域是背景的可能性较大。

3.5 HOG 特征

HDCT [13] 中用到。HOG(Histogram of Oriented Gradients，方向梯度直方图) 特征最早是由法国国家计算机技术和控制研究所 (INRIA) 的 Navneet Dalal 和 Bill Triggs 在 2005 年发表在 CVPR 上的



(a) (b)

图 6: LBP

论文 [5] 中提出的。

Dalal 提出的 HOG 特征提取的过程如图 7，进一步表述如下：

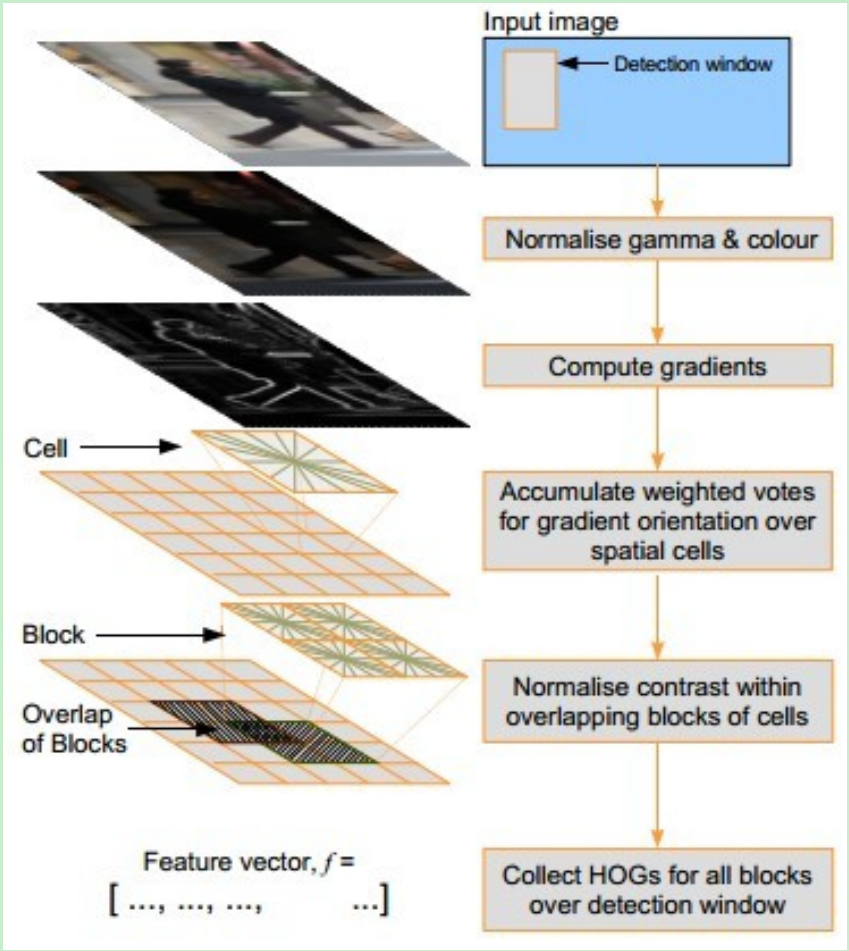


图 7: 算法流程图

1）灰度化

2）采用 Gamma 校正法对输入图像进行颜色空间的标准化 (归一化)；目的是调节图像的对比度，降低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰。Gamma 压缩公式：

$$I(x,y) = I(x,y)^{gamma} \tag{15}$$

3）计算图像每个像素的梯度 (包括大小和方向)，主要是为了捕获轮廓信息，同时进一步弱化光照的干扰。

图像中像素点 (x,y) 的梯度为

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \tag{16}$$

$$G_y(x,y) = H(x,y+1) - H(x,y-1) \tag{17}$$

式中 $G_x(x,y), G_y(x,y), H(x,y)$ 分别表示输入图像中像素点 (x,y) 处的水平方向梯度、垂直方向梯度和灰度值。像素点 (x,y) 处的梯度幅值和梯度方向分别为：

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{18}$$

$$\alpha(x,y) = tan^{-1}(\frac{G_y(x,y)}{G_x(x,y)} \tag{19}$$

最常用的方法是：首先用 $[-1,0,1]$ 梯度算子对原图像做卷积运算，得到 x 方向 (水平方向，以向右为正方向) 的梯度分量 gradscalx，然后用 $[1,0,-1]^T$ 梯度算子对原图像做卷积运算，得到 y 方向 (竖直方向，以向上为正方向) 的梯度分量 gradscaly。然后再用以上公式计算该像素点的梯度大小和方向。

4）将图像每 $16*16$(取其它也可以) 个像素分到一个 cell 中，对于 $256*256$ 的图像来说，就分成了 $16*16$ 个 cell 了。

5）对于每个 cell 求其梯度方向直方图，通常取 $bin = 9$(取其它也可以) 个方向 (特征)，也就是每 $360/9 = 40$ 度分到一个方向，形成每个 cell 的 descriptor。如图 8，当某像素的梯度方向是 20-40 度，然后它的梯度大小是 $|grad|$，那么直方图第 2 个 bin 的计数就要加上 $|grad|$。

6）由于局部光照的变化以及前景 -背景对比度的变化，使得梯度强度的变化范围非常大。这就需要对梯度强度做归一化。归一化能够进一步地对光照、阴影和边缘进行压缩。为此可以将每 $2*2$(取其它也可以) 个 cell 合成一个大的、空间上连通的 block，所以这里就有 $(16-1)*(16-1) = 225$ 个 block。一个 block 内所有 cell 的特征向量串联起来便得到该 block 的 HOG 特征。这些 block 是互有重叠的，这就意味着：每一个 cell 的特征会以不同的结果多次出现在最后的特征向量中。

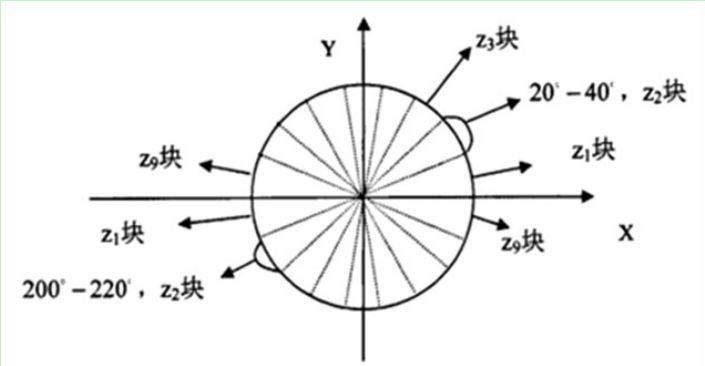


图 8: 梯度方向 bin 的划分

7）所以每个 block 中都有 $2*2*9$ 个特征，一共有 225 个 block，所以总的特征有 $225*36$ 个。

遇到的疑问：

1）在显著性检测中，先将图像分割成区域，然后怎么提取该区域的 HOG 特征？

2）一定要将彩色图像先转化为灰度图像才能提取 HOG 特征吗？

HDCT [13] 方法中用到 HOG 特征 [6]，采用的是 VLFeat 官网上的代码。论文中首先也是对图像进行超像素分割，然后求每个超像素区域内像素点坐标的平均值 (x_i, y_i)，i 代表第 i 个超像素区域，求该区域的 HOG 特征就是以该坐标点为中心的 $17*17$ 的网格作为输入图像 (只取该网格内图像 r 通道的值)，17 作为 cellSize，然后对每个超像素区域得到一个 31 维的 HOG 特征向量。

2、PISA [26] 中用到 Structure-Based Contrast。这种结构描述子是对像素点 p 周围的矩形区域 W_p 内的梯度分布用直方图 $h^g(p)$ 来建模。$h^g(p)$ 计算了包含梯度方向分量和幅值分量的向量出现的频率。论文中将每个分量量化为 8 个 bins，并且称得到的特征空间为 OM 空间 (16 维)。W_p 窗口被固定为 9×9，最终求出的描述子被称为 OM 结构描述子。

计算完每个像素上的结构描述子后，用 k-均值方法将 OM 特征空间分成 K_g 个聚类 $\{\varphi_1, \ldots, \varphi_{K_g}\}$。计算像素点 p 上的结构对比度等价于计算 p 所属聚类 φ_i 上的对比度

$$U^g(p) = U^g(\mathbf{h}^g(p)) = \sum_{j=1}^{K_g} w_j ||\mathbf{h}^g(\varphi_i), \mathbf{h}^g(\varphi_j)|| \tag{20}$$

w_j 是聚类 φ_j 中像素的个数，$\mathbf{h}^g(\varphi_i)$ 是聚类 φ_i 中的 OM 直方图的均值。

3.6 Visual Complexity Contrast

论文 [33] 中用到，信息论中可以用熵来计算 visual complexity，区域 r_i 和 r_j 之间的 visual complexity contrast $D_e(r_i, r_j)$ 就被定义为

$$D_e(r_i, r_j) = [H(r_i) - H(r_j)]^2 \tag{21}$$

其中，$H(r_i)$ 表示区域 r_i 内的熵

$$H(r_i) = \sum_{p=1}^{n_{c,i}} f(c_{p,i}) \cdot log_2 f(c_{p,i}) \tag{22}$$

其中，$c_{p,i}$ 是区域 r_i 中第 p 种颜色，$n_{c,i}$ 区域 r_i 中包含的颜色个数，$f(c_{p,i})$ 表示区域 r_i 中颜色 $c_{p,i}$ 出现的概率。

3.7 Background Weighted Contrast

RBD [34] 中用到，其中将 background weighted contrast 定义为

$$wCtr(p) = \sum_{i=1}^{N} d_{app}(p, p_i) w_{spa}(p, p_i) w_i^{bg}, \tag{23}$$

4. Hierarchical over-segmentation

方法 TS [33] 中用到，hierarchical over-segmentation 是通过先对原图像进行超像素分割，再通过迭代将每一分割层再分割成更精细的子区域来实现。

论文 [29] 中对原图像先提取 3 个 layers，即对原图像 (400×300) 用 watershed-like 方法 [7] 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。

5. Priors

5.1 Center Prior 或 Location Prior

SDSP [32] 中将 Location Prior 描述为：处于图像偏中央位置的物体更能吸引人的注意。这里 Location Prior 是按像素级计算的，定义如下：

$$S_D(x) = exp\left(-\frac{||x-c||_2^2}{\sigma_D^2}\right) \tag{24}$$

效果如图 9。

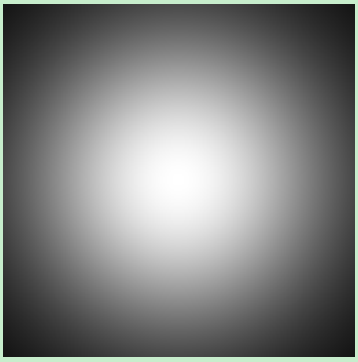


图 9: SDSP: Center Prior

PBS [30] 中用到 Convex-Hull-Based Center Prior，即首先估计显著目标的中心位置 (x_0, y_0)，再计算每个超像素 i 的显著性：

$$S_{ce}(i) = exp\left(-\frac{||x_i-x_0||^2}{2\sigma_x^2} - \frac{||y_i-y_0||^2}{2\sigma_y^2}\right) \tag{25}$$

5.2 Backgroundness Prior

通常是基于伪背景的假设，即假设处于图像周围狭窄边界上的区域是背景区域。

1、2012 年，论文 [28] 中提出了关于 background 的两种 prior，叫做 boundary and connectivity priors。boundary prior 来源于摄影构图的基本规则，大部分摄影师都不会将显著目标切断在视觉框架中，也就是说，位于图像边界上的通常是背景。这个 prior 比之前提出的 center prior 更通用一些，因为显著目标可能并不会正好位于图像正中央，例如三分构图法，但它们很少会处在图像边界上。connectivity prior 是从图像中背景的外观特征得来的，背景区域通常较大并且是同质的。也就是说，背景中绝大多数图像块可以很容易地彼此连接起来。如图 10。

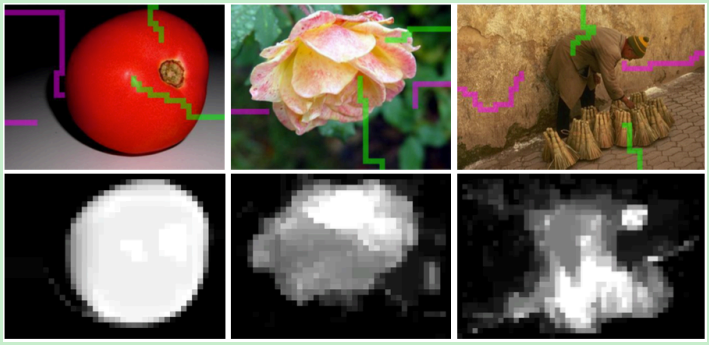


图 10: Geodesic saliency 图例

首先对一幅图像构造无向图 $G = \{V, \varepsilon\}$，图中的节点由两部分 (图像块 $\{P_i\}$ 和伪背景节点 B) 构成，即 $V = \{P_i\} \cup \{B\}$。边也包括两种类型，一种是 internal edges，连接了所有相邻的图像块，另一种是 boundary edges，连接了处于图像边界的块和背景节点，即 $\varepsilon = \{(P_i, P_j)|P_i \text{ 与 } P_j \text{ 相邻 }\} \cup \{(P_i, B)|P_i \text{ 在图像边界上}\}$。

图像块 P 上的 geodesic saliency 被定义为沿着从 P 到背景节点 B 的最短路径上的边的权值的累加和。

$$saliency(P) = \min_{P_1=P, P_2, \ldots, P_n=B} \sum_{i=1}^{n-1} weight(P_i, P_{i+1}), s.t.(P_i, P_{i+1}) \in \varepsilon \tag{26}$$

1）伪背景的节点是怎么找的?

2）如何求 internal edge weight?

internal edge weight 是相邻图像块之间的 appearance distance，采用的是一种简单有效的 weight clipping approach，图像块与块之间的 appearance distance 求的是两个块在 LAB 颜色空间上的颜色平均值的不同。对图像中的每一个块，算出它与所有 neighbors 的最小 appearance distance，然后取这些 distance 的均值作为“insignificance”distance threshold，当某距离比这个阈值小时就说明它是不重要的，将它置为 0。

3）如何求 boundary edge weight?

boundary edge weight 度量的是处于图像边界的图像块不是背景的可能性大小。当 boundary prior 是完全有效的时候，所有的 boundary edge weight 就都是 0。但现实没有这么理想化，显著目标只要有哪怕一小部分落到了图像边界，也会产生不好的效果。这里可以通过用其他的显著性的方法来实现，已知图像边界上的块，计算每个块 P_i 的显著性作为 the weight of boundary edge(P_i, B)。

MR [31] 算法步骤如图 11，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。

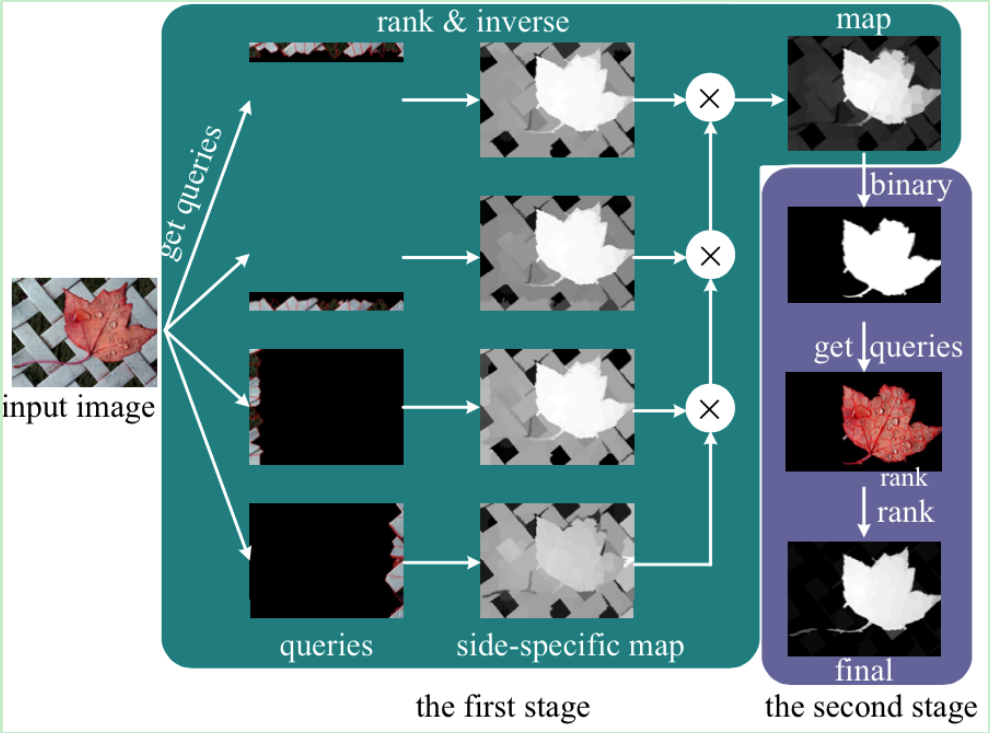


图 11: 算法步骤

2、论文 [9] 中也用到 backgroundness prior，即将图像四周边界看成是伪背景节点，将马尔可夫随机游走的性质与显著性检测联系起来。

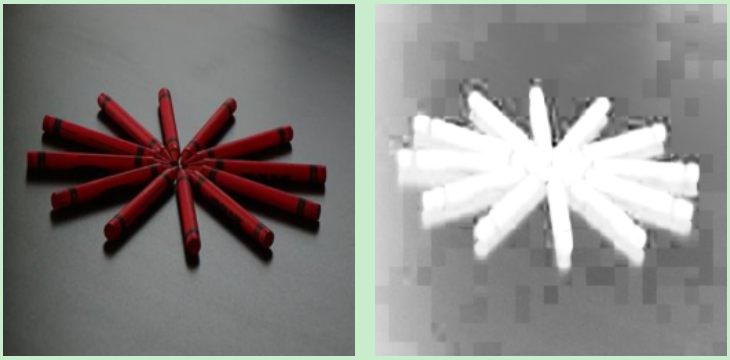
3、OptSeedProp [20] 中用到

5.3 Boundary connectivity prior

由于基于伪背景假设的显著目标检测方法在目标碰到图像边界时会失效，所以后来又提出了 boundary connectivity prior，这种 prior 被描述为：显著目标与图像边缘的连接程度比背景与边缘的连接程度小。

论文 [34] 中将一个区域内的 boundary connectivity score 定义为它沿图像边缘的长度与整个区域面积的比值：

$$BndCon(p) = \frac{Len_{bnd}(p)}{\sqrt{Area(p)}} \tag{27}$$



(a) (b)

图 12: SDSP: Color Prior

5.4 Color Prior

SDSP [32] 中将 Color Prior 描述为：暖色 (例如红色和黄色) 比冷色 (例如绿色和蓝色) 更能吸引人类视觉系统的注意。先将图像由 RGB 颜色空间转换到 $CIEL*a*b*$ 空间，$\{f_L(x)\}, \{f_a(x)\}, \{f_b(x)\}$ 分别代表 $L*$ 通道，$a*$ 通道和 $b*$ 通道。这里 Color Prior 是按像素级计算的，定义如下：

$$S_c(x) = 1 - exp\left(-\frac{f_{an}^2(x) + f_{bn}^2(x)}{\sigma_c^2}\right) \tag{28}$$

其中

$$f_{an}(x) = \frac{f_a(x) - mina}{maxa - mina}, f_{bn}(x) = \frac{f_b(x) - minb}{maxb - minb} \tag{29}$$

效果如图 12。

5.5 Objectness Prior

论文 [3] 中将 objectness 与 regional saliency 融合，形成一个 graphical model。

论文 [12] 中从两个方面估计 objectness，一是 pixel-level objectness estimation，二是 region-level objectness estimation。

Pixel-level Objectness：每个像素点的 objectness 值是以该像素点为中心的局部窗口内包含完整目标的概率。在图像上随机抽样 N 个窗口，求每一个窗口 w 上用来表示 objectness 的概率值 $P(w)$。将

所有窗口集记作 W，对每个像素点 x 可以得到它的像素级 objectness $O_p(x)$

$$O_p(x) = \sum_{w \in W \ and \ x \in w} P(W_x), \tag{30}$$

w 表示 W 中包含像素点 x 的任意窗口，论文中取 $N = 10000$。

Region-level Objectness：对每个区域 Λ_i，计算该区域内的 region-level objectness $O_r(\Lambda_i)$ 为：

$$O_r(\Lambda_i) = \frac{1}{|\Lambda_i|} \sum_{x \in \Lambda_i} O_p(x). \tag{31}$$

计算完一个区域内的 objectness 后，把这个值赋给该区域内的所有像素。这样就可以得到整幅图像 I 的 objectness map，简称为 $O(I)$。

5.6 Smoothness Prior

PBS [30] 中用到，平滑约束通常是用在基于图的目标分割中，目的是使图像中相邻的像素拥有同样的 label 值。论文中先将图像分割成超像素，然后将其映射为图，每个超像素对应图中的节点，有共同的边界的超像素之间有一条边，边上的权值为 $w_{ij} \in W$：

$$w_{ij} = exp\left(-\frac{||c_i - c_j||}{2\sigma_w^2}\right) \tag{32}$$

其中 c_i 和 c_j 是 CIELab 空间超像素区域内像素的颜色均值。可以看到关系矩阵 W 是一个稀疏矩阵。定义如下 saliency cost function 来表示这种 smoothness prior：

$$E(S) = \sum_i (S(i) - S_{in}(i))^2 + \lambda \sum_{i,j} w_{ij}(S(i) - S(j))^2 \tag{33}$$

$S(i)$ 和 $S(i)$ 分别表示节点 i 和 j 的所要求的显著值，$S_{in}(i)$ 是节点 i 的初始显著值，λ 是规范化系数。其中等式右边第一项是 fitting constraint，表示一幅好的显著图与初始显著图之间不会变化太多。第二项是 smoothness constraint，一幅好的显著图上相邻超像素的显著值不会相差太多。超像素上的最优显著值是通过最小化该 cost function 来计算。令该方程关于 S 的导数为 0 可得

$$S^* = \mu(D - W + \mu I)^{-1} S_{in} \tag{34}$$

其中 D 是三角矩阵，并且有 $d_{ii} = \sum_j (w_{ij})$，$\mu = 1/(2\lambda)$。

5.7 Focusness Prior

论文 [12] 中将 focusness 定义为焦点模糊程度。

6. Feature Contrasts/Priors 的融合

6.1 相乘

对相乘融合而言，要保证相乘的量取值在 $[0,1]$ 之间。

目前来看，分为两种。

1. 求出两个区域间关于多种关于 visual cues 的对比度之后，要先将这些对比度融合，例如可以相乘，求出最终的对比度公式。然后再对每个区域计算 local 或 global contrast。

论文 [33] 中，两个区域 r_i、r_j 间的最终的对比度定义如下：

$$D_r(r_i,r_j) = D_c(r_i,r_j) \cdot exp[\sigma_e^2 \cdot D_e(r_i,r_j)] \tag{35}$$

其中 $D_c(r_i,r_j)$ 是指 color contrast，$D_c(r_i,r_j)$ 是指 visual complexity contrast。

对每个区域 r_i，其空间加权 global contrast 定义为：

$$U(r_i) = \sum_{j \neq i} w_{ij} \cdot D_r(r_i,r_j) \cdot \phi_j \tag{36}$$

$$w_{ij} = \frac{1}{Z_i} \cdot exp[-\sigma_s^2 \cdot D_s(r_i,r_j)] \tag{37}$$

其中，ϕ_j 是指区域 r_j 内的像素个数，即区域 r_j 的大小。$D_s(r_i,r_j)$ 表示区域 r_i 和 r_j 之间的空间距离。ϕ_s 则用来控制空间加权 w_{ij} 的影响程度，ϕ_s 越大，对 $U(r_i)$ 的影响越小。$\frac{1}{Z_i}$ 是归一化因子，保证 $\sum_{j \neq i} w_{ij} = 1$。

2. 对每个 visual cue 求出 local 或 global contrast 之后，再对这些对比度进行融合。

例如论文 [29] 中，作者用了两种 cues：

1）local contrast

$$C_i = \sum_{j=1}^{n} w(R_i)\Phi(i,j)||c_i - c_j||_2 \tag{38}$$

其中 c_i 和 c_j 分别表示区域 R_i 和 R_j 中的颜色,$w(R_j)$ 指 R_j 中像素的个数。$\Phi(i,j) = exp\{-D(R_i,R_j)/\sigma^2\}$，控制了区域 R_i 和 R_j 之间的空间距离，其中 $D(R_i,R_j)$ 是区域 R_i 和 R_j 的中心的欧几里得距离的平方。

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。

$$H_i = \frac{1}{w(R_i)} \sum_{x_i \in R_i} exp\{-\lambda ||x_i - x_c||^2\} \tag{39}$$

其中 $\{x_0, x_1 \ldots\}$ 是区域 R_i 中的像素坐标的集合，x_c 是图像中心坐标。

然后将 C_i 与 H_i 组合起来，得到

$$\bar{s}_i = C_i \cdot H_i \tag{40}$$

方法 PBS [30] 中在求初始显著图时先计算了 spatially weighted contrast

$$S_{co}(i) = \sum_{j \neq i} ||c_i - c_j|| \cdot exp\left(-\frac{||p_i - p_j||^2}{2\sigma_p^2}\right) \tag{41}$$

然后又计算了 convex-hull-based center prior map，即首先估计显著目标的中心位置 (x_0, y_0)，再计算每个超像素 i 的显著性：

$$S_{ce}(i) = exp\left(-\frac{||x_i - x_0||^2}{2\sigma_x^2} - \frac{||y_i - y_0||^2}{2\sigma_y^2}\right) \tag{42}$$

最后将两者相乘得到初始显著图

$$S_{in}(i) = S_{co}(i) \times S_{ce}(i) \tag{43}$$

6.2 相加

6.3 利用机器学习进行优化

6.3.1 基于图的显著性检测方法

能量方程

1、论文 [34] 中是通过最小二乘实现全局优化。其中的 cost function 如下：

$$\sum_{i=1}^{N} w_i^{bg} s_i^2 + \sum_{i=1}^{N} s_i^{fg} (s_i - 1)^2 + \sum_{i,j} w_{ij} (s_i - s_j)^2 \tag{44}$$

其中第一项是背景约束，当某区域的 w_i^{bg} 较大时，是背景的概率较大，方程的第一项占的比重较多，为使整个方程值最小，需使 s_i 近似为 0，也就是使该区域的显著性近似为 0。第二项是前景约束，当某区域的 w_i^{fg} 较大时，第二项占的比重较大，需使 s_i 近似为 1。w_i^{fg} 可以通过目前已有的一些显著性方法或它们的组合来计算。第三项是平滑约束，保证显著值的连续性，w_{ij} 定义如下：

$$w_{ij} = exp(-\frac{d_{app}^2(p_i, p_j)}{2\sigma_{clr}^2}) + \mu \tag{45}$$

超像素 p_i 与 p_j 越相似，$d_{app}(p_i, p_j)$ 越小，w_{ij} 越大，越需要 s_i，s_j 近似相等。

2、方法 PBS [30] 中求最终显著图时用到如下 saliency cost function：

$$E(S) = \sum_i (S(i) - S_{in}(i))^2 + \lambda \sum_{i,j} w_{ij} (S(i) - S(j))^2 \tag{46}$$

$S(i)$ 和 $S(i)$ 分别表示节点 i 和 j 的所要求的显著值，$S_{in}(i)$ 是节点 i 的初始显著值，λ 是规范化系数。其中等式右边第一项是 fitting constraint，表示一幅好的显著图与初始显著图之间不会变化太多。第二项是 smoothness constraint，一幅好的显著图上相邻超像素的显著值不会相差太多。超像素上的最优显著值是通过最小化该 cost function 来计算。

条件随机场

条件随机场 (Conditional Random Fields, CRFs) 最早由 Lafferty 等人 [14] 于 2001 年提出，其模型思想主要来源于最大熵模型。

马尔可夫随机场

7. 显著图的融合

1、论文 [2] 中是将由图像 I 得到的 m 幅显著图 $\{S_i || 1 \leq i \leq m\}$ 通过如下方式融合得到最终的显著图：

$$S(p) = P(y_p = 1 | S_1(p), S_2(p), \ldots, S_m(p)) \propto \frac{1}{Z} \sum_{i=1}^{m} \zeta(S_i(p)), \tag{47}$$

其中，$S(p)$ 表示图像 I 上像素点 p 处的显著值，$S_i(p)$ 表示显著图 S_i 上像素点 p 处的显著值，y_p 是一个二值随机变量，当像素点 p 是显著的时值为 1，否则值为 0，Z 是一个常数。

由方程 (47) 可看出，当有一个方法产生的值很小时，在相乘时对整个结果影响较大，论文中对函数 ζ 采用了三种改进形式，包括：

$$\zeta_1(x) = x,\ \zeta_2(x) = exp(x),\ and\, \zeta_3(x) = \frac{-1}{log(x)} \tag{48}$$

当进行融合的方法的性能之间相差较大时，这种融合方法效果不是很好，主要原因是没有考虑到不同方法的效果差异而将它们同等对待。因此，性能较差的方法会在最终融合效果中占主导作用。

2、论文 [22] 中采用了两种显著性融合方法。

第一种 (pixel-wise aggregation)：对图像中的每个像素点 p,对应一个特征向量 $\mathbf{x}(p) = (S_1(p), S_2(p), \ldots, S_m$ 其中 $S_i(p)$ 是像素点 p 在显著图 S_i 中的显著值。这里也定义了一个二值的随机变量 y_p，表征某像素是否是显著的。如果像素点 p 是显著的值就为 1，反之值为 0。最终的显著值 $S(P)$ 可看成是求取后验概率 $P(y_p = 1|x(p))$，这个后验概率可以用 logistic model [1] 来建模，

$$P(y_p = 1|x(p); \lambda) = \sigma(\sum_{i=1..m} \lambda_i S_i(p) + \lambda_{m+1}) \tag{49}$$

其中模型参数 $\lambda = \{\lambda_i | i = 1..m+1\}$ 权衡了每幅显著图的贡献大小。这里 $\sigma(.)$ 指的是如下的 sigmoid 函数

$$\sigma(z) = \frac{1}{1 + exp(-z)} \tag{50}$$

参数 λ 可以通过在训练集上用 standard logistic regression technique 学习到。

第二种 (CRF-based aggregation)：第一种方法分别计算各个像素上的显著值，而忽略了相邻像素间的关系。第二种方法有效解决了该问题，通过二值条件随机场 [14] 来估计显著值。利用条件随机场可以获取相邻像素间的相互关系。

像第一种方法一样，将每个像素看成是一个节点，对应每个节点有一个特征向量 $\mathbf{x}(p) = (S_1(p), S_2(p), \ldots, S_m(p))$ 和一个二值随机标签 y_p，1 代表显著的，0 代表非显著的。每个像素点上的显著性标签不仅依赖于它的特征向量，还依赖于相邻像素点上的标签。像素之间的相互关系也依赖于其上的特征。论文中使用网格形状的条件随机场来对标签和特征间的关系、相邻像素间标签的特征依

赖关系进行建模。定义特征 $X=\{x_p|p\in I\}$ 上的标签 $Y=\{y_p|p\in I\}$ 的条件分布如下：

$$P(Y|X;\theta)=\frac{1}{Z}exp(\sum_{p\in I}f_d(\mathbf{x}_p,y_p)+\sum_{p\in I}\sum_{q\in N_p}f_s(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q)) \tag{51}$$

其中 p 是图像 I 中的一个像素，$\mathbf{x}_p$ 表示其上的特征，y_p 是它的显著性标签，θ 是条件随机场模型的参数。$f_d(\mathbf{x}_p,y_p)$ 是定义特征和标签之间关系的特征函数。$f_s(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q)$ 是另一个定义相邻像素 p 和 q 标签的特征依赖关系的特征函数。N_p 是与 p 相关的像素集合。这里表示 8 邻域关系。Z 是一个常数。

特征函数 $f_d(\mathbf{x}_p,y_p)$ 的定义仅仅与输入的显著图 S_i 有关，

$$f_d(\mathbf{x}_p,y_p)=\sum_{i=1}^{m}\lambda_i S_i(p)y_p+\lambda_{m+1}y_p \tag{52}$$

其中 $\{\lambda_i\}$ 是条件随机场模型的参数集，$S_i(p)$ 是显著图 S_i 中像素点 p 处的显著值。

特征函数 $f_s(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q)$ 由两个分量组成，

$$f_s(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q)=f_e(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q)+f_c(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q) \tag{53}$$

第一个分量 $f_e(\mathbf{x}_p,\mathbf{x}_q,y_p,y_q)$ 表示如果两个像素点用同一种显著性方法计算出的显著值不同，它们在融合结果中就可能会有不同的显著性标签。

参考文献

[1] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[2] Ali Borji, Dicky N Sihite, and Laurent Itti. Salient object detection: A benchmark. In *Computer Vision–ECCV 2012*, pages 414–429. Springer, 2012.

[3] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 914–921. IEEE, 2011.

[4] Ming-Ming Cheng, Guo-Xin Zhang, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Global contrast based salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 409–416. IEEE, 2011.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[6] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

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

[8] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[9] Bowen Jiang, Lihe Zhang, Huchuan Lu, Chuan Yang, and Ming-Hsuan Yang. Saliency detection via absorbing markov chain. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1665–1672. IEEE, 2013.

[10] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Tie Liu, Nanning Zheng, and Shipeng Li. Automatic salient object segmentation based on context and shape prior. In *BMVC*, volume 6, page 7, 2011.

[11] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2083–2090. IEEE, 2013.

[12] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1976–1983. IEEE, 2013.

[13] Jiwhan Kim, Dongyoon Han, Yu-Wing Tai, and Junmo Kim. Salient region detection via high-dimensional color transform. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 883–890. IEEE, 2014.

[14] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[15] Xi Li, Yao Li, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Contextual hypergraph modeling for salient object detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3328–3335. IEEE, 2013.

[16] Risheng Liu, Junjie Cao, Zhouchen Lin, and Shiguang Shan. Adaptive partial differential equation learning for visual saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3866–3873. IEEE, 2014.

[17] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[18] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[19] Jiangbo Lu, Keyang Shi, Dongbo Min, Liang Lin, and Minh N Do. Cross-based local multipoint filtering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 430–437. IEEE, 2012.

[20] Song Lu, Vijay Mahadevan, and Nuno Vasconcelos. Learning optimal seeds for diffusion-based salient object detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2790–2797. IEEE, 2014.

[21] Yu-Fei Ma and Hong-Jiang Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 374–381. ACM, 2003.

[22] Long Mai, Yuzhen Niu, and Feng Liu. Saliency aggregation: A data-driven approach. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1131–1138. IEEE, 2013.

[23] Rotem Mairon and Ohad Ben-Shahar. A closer look at context: From coxels to the contextual emergence of object saliency. In *Computer Vision–ECCV 2014*, pages 708–724. Springer, 2014.

[24] Mark Nixon, Mark S Nixon, and Alberto S Aguado. *Feature extraction and image processing for Computer Vision*. Academic Press, 2012.

[25] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, number 1, pages 582–585, 1994.

[26] Keyang Shi, Keze Wang, Jiangbo Lu, and Liang Lin. Pisa: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2115–2122. IEEE, 2013.

[27] Bolan Su, Shijian Lu, and Chew Lim Tan. Blurred image region detection and classification. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1397–1400. ACM, 2011.

[28] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *Computer Vision–ECCV 2012*, pages 29–42. Springer, 2012.

[29] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1155–1162. IEEE, 2013.

[30] Chuan Yang, Lihe Zhang, and Huchuan Lu. Graph-regularized saliency detection with convex-hull-based center prior. *Signal Processing Letters, IEEE*, 20(7):637–640, 2013.

[31] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3166–3173. IEEE, 2013.

[32] Lin Zhang, Zhongyi Gu, and Hongyu Li. Sdsp: A novel saliency detection method by combining simple priors. In *ICIP*, pages 171–175. Citeseer, 2013.

[33] Guokang Zhu, Qi Wang, and Yuan Yuan. Tag-saliency: Combining bottom-up and top-down information for saliency detection. *Computer Vision and Image Understanding*, 118:40–49, 2014.

[34] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2814–2821. IEEE, 2014.