**相似图片搜索的原理**

作者： [阮一峰](http://www.ruanyifeng.com/)

日期： [2011年7月21日](http://www.ruanyifeng.com/blog/2011/07/)

上个月，Google把["相似图片搜索"](http://www.google.com/insidesearch/searchbyimage.html)正式放上了首页。

你可以用一张图片，搜索互联网上所有与它相似的图片。点击[搜索框](http://images.google.com.hk/)中照相机的图标。

[](http://images.google.com.hk/)

一个对话框会出现。



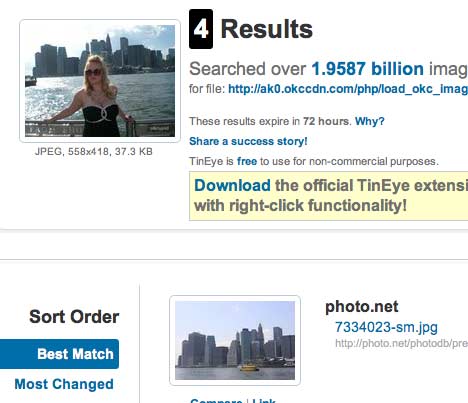
你输入网片的网址，或者直接上传图片，Google就会找出与其相似的图片。下面这张图片是美国女演员Alyson Hannigan。



上传后，Google返回如下结果：



类似的"相似图片搜索引擎"还有不少，[TinEye](http://www.tineye.com/)甚至可以找出照片的拍摄背景。



==========================================================

这种技术的原理是什么？计算机怎么知道两张图片相似呢？

根据[Neal Krawetz](http://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html)博士的解释，原理非常简单易懂。我们可以用一个快速算法，就达到基本的效果。

这里的关键技术叫做"感知哈希算法"（Perceptual hash algorithm），它的作用是对每张图片生成一个"指纹"（fingerprint）字符串，然后比较不同图片的指纹。结果越接近，就说明图片越相似。

下面是一个最简单的实现：

**第一步，缩小尺寸。**

将图片缩小到8x8的尺寸，总共64个像素。这一步的作用是去除图片的细节，只保留结构、明暗等基本信息，摒弃不同尺寸、比例带来的图片差异。

http://image.beekka.com/blog/201107/bg2011072107.png http://image.beekka.com/blog/201107/bg2011072107.png

**第二步，简化色彩。**

将缩小后的图片，转为64级灰度。也就是说，所有像素点总共只有64种颜色。

**第三步，计算平均值。**

计算所有64个像素的灰度平均值。

**第四步，比较像素的灰度。**

将每个像素的灰度，与平均值进行比较。大于或等于平均值，记为1；小于平均值，记为0。

**第五步，计算哈希值。**

将上一步的比较结果，组合在一起，就构成了一个64位的整数，这就是这张图片的指纹。组合的次序并不重要，只要保证所有图片都采用同样次序就行了。

http://image.beekka.com/blog/201107/bg2011072109.png = http://image.beekka.com/blog/201107/bg2011072109.png = 8f373714acfcf4d0

得到指纹以后，就可以对比不同的图片，看看64位中有多少位是不一样的。在理论上，这等同于计算["汉明距离"](http://zh.wikipedia.org/wiki/%E6%B1%89%E6%98%8E%E8%B7%9D%E7%A6%BB)（Hamming distance）。如果不相同的数据位不超过5，就说明两张图片很相似；如果大于10，就说明这是两张不同的图片。

具体的代码实现，可以参见[Wote](http://www.reddit.com/r/programming/comments/hql8b/looks_like_it_for_the_last_few_months_i_have_had/c1xkcdd)用python语言写的[imgHash.py](http://www.ruanyifeng.com/blog/2011/07/imgHash.txt)。代码很短，只有53行。使用的时候，第一个参数是基准图片，第二个参数是用来比较的其他图片所在的目录，返回结果是两张图片之间不相同的数据位数量（汉明距离）。

这种算法的优点是简单快速，不受图片大小缩放的影响，缺点是图片的内容不能变更。如果在图片上加几个文字，它就认不出来了。所以，它的最佳用途是根据缩略图，找出原图。

实际应用中，往往采用更强大的[pHash](http://www.phash.org/)算法和[SIFT](http://en.wikipedia.org/wiki/Scale-invariant_feature_transform)算法，它们能够识别图片的变形。只要变形程度不超过25%，它们就能匹配原图。这些算法虽然更复杂，但是原理与上面的简便算法是一样的，就是先将图片转化成Hash字符串，然后再进行比较。