图像处理算法

**一、锐化**  
  
　　锐化的算法很简单，就是比较相邻的几个像素，把当前像素加上和周围的像素的差就可以了。这里我给出一个示例：

  
　　假设有一个图片，4\*4，共16个像素，分别用A－－L来代表。我们先观察这个图片，只有中间的F,G,J,K这四个像素的“邻居”是全的。  
  
　　为了简便起见，我们只处理这4个像素，因为在实际的图片中由于图片的大小都很多像素组成，所以周围的一圈像素不做处理不会影响到最终的效果。  
  
　　先计算差值:

|  |
| --- |
| Delta= F - (A+B+C+E+G+I+J+K) / 8 (A+B+C+E+G+I+J+K) / 8就是F周围的像素的平均值， |

　　将这个平均值乘以一个系数再加到F上，就得到了一个新的F值：

|  |
| --- |
| F=F + Delta \* Alpha |

　　这个系数Alpha就是锐化度，改变这个系数就能得到不同的锐化效果。不过一般都是取得比较小的，如：0.3  
  
　　于是，我们只要使用两个循环来遍历整个图片的像素值（去除边界）就能得到一个锐化的效果了。  
  
　　但是大家或许会发现在处理后面几个点的时候，前面的点的值已经不是原来的值了，比如处理G的时候，需要用到F的值，而F则已经被改变，并且F的改变又和G的值有关系，这样就会变成一种循环引用。为了避免整个问题，这里给出一个改良的方法：

  
　　我们从A点开始做，将差值计算方法改成:

|  |
| --- |
| Delta= A - (B+E+F) / 3 F=F + Delta \* Alpha |

　　按照从左到右，从上到下的顺序来扫描所有像素，这时在计算中就不会遇到已经被处理过的像素了，并且因为减少了参与运算的像素，整个处理过程也得以加快。