

## 自适应锐化

其实没什么好说的，就是在 USM 基础上加一个方差的调整。来源：<http://www.giantpandacv.com/academic/%E4%BC%A0%E7%BB%9F%E5%9B%BE%E5%83%8F/%E4%B8%80%E4%BA%9B%E6%9C%89%E8%B6%A3%E7%9A%84%E5%9B%BE%E5%83%8F%E7%AE%97%E6%B3%95/OpenCV%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86%E4%B8%93%E6%A0%8F%E4%BA%94%20%20%E3%80%8AReal-time%20adaptive%20contrast%20enhancement%20for%20imaging%20sensors%E3%80%8B%E8%AE%BA%E6%96%87%E8%A7%A3%E8%AF%BB%E5%8F%8A%E5%AE%9E%E7%8E%B0/>

假设图像中的某个点表示为  $x(i, j)$ ，窗口大小为  $(2n + 1) \times (2n + 1)$ ，计算均值和标准差。

$$\begin{aligned} \bullet \quad m_x(i, j) &= \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} x(k, l) \\ \bullet \quad \sigma_x^2(i, j) &= \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} [x(k, l) - m_x(i, j)]^2. \end{aligned}$$

算法可以表示为：

$$f(i, j) = m_x(i, j) + G(i, j)[x(i, j) - m_x(i, j)]$$

$G(i, j)$  可以直接取一个常数  $C$ ，这样其实就类似是 USM，只不过是均值滤波而不是高斯滤波。也可对每个位置使用不同的增益，Lee 等人提出了下面的解决方案：

$$f(i, j) = m_x(i, j) + \frac{D}{\sigma_x(i, j)} [x(i, j) - m_x(i, j)]$$

其中  $D$  是一个常数，这样系数和局部均方差成反比，在图像的边缘或者其他变化剧烈的地方，局部均方差比较大，因此值就比较小，这样就不会产生振铃效应。

1. 在平滑的区域，局部均方差就会很小，这样  $G$  的值比较大，从而引起了噪音的放大，所以需要给  $G$  的最大值做一定的限制才能获得更好的效果。
2.  $D$  这个常数一些文章认为取图像的全局均值，也可以使用全局均方差。