Atitit 素描滤镜艺术滤镜的实现

在计算机图形学发展史中，真实感绘制一直是主旋律。不过从20实际90年代中期开始，非真实感图像绘制(Non-Photorealistic Rendering，NPR）逐渐成为一个研究热点。说白了，真实感绘制目标是像照片般真实地再现客观世界，而非真实感图像绘制专注于图形个性化和艺术化的表达，它主要用来表现图形的艺术特质，以及模拟艺术作品（甚至包括作品中的缺陷）。

自己使用算法实现

Java里面没有库实现。。

这个系列，我们就主要使用PIL来进行滤镜方面的处理，包括素描、铅笔画、油画等等滤镜效果的实现。在以后我会把代码托管出来

在开始素描效果之前，我们需要首先进行灰度图像预处理。所幸的是，用PIL非常容易实现。设img是Image类的实例，我们只要用convert函数强制转换为L模式即可。

[?](http://www.qinxuye.me/article/implement-sketch-and-pencil-with-pil/)

|  |  |
| --- | --- |
| 1 | img **=** img.convert("L") |

素描滤镜的处理关键是对边缘的查找。通过对边缘的查找可以得到物体的线条感。在对图像进行灰度化处理后，我们首先定义一个阈值（threshold）。我们知道素描主要强调的是明暗度的变化，绘制时是斜向方向，通过经验，我们将每个像素点的灰度值与其右下角的灰度值进行比较，当大于这个阈值时，就判断其是轮廓并绘制

对于铅笔画来说，原理和素描十分相似，但是大家学过画画的就知道，素描强调的是阴影的效果，是斜向作画，而铅笔画主要是勾勒轮廓。因此在对每个像素点的处理上，就和素描产生变化。对于任意一个像素点，求出这个像素点的R、G、B三个分量与周围8个点的相应分量的平均值的差，如果这三个差都大于或者等于某个阈值，就画出线条。最后，铅笔画的作画不是单调的一种颜色，因此加入Alpha分量，大小等于对应点的alpha分量即可。于是，代码如下；

用PIL实现滤镜（一）——素描、铅笔画效果 - 残阳似血的博客.html