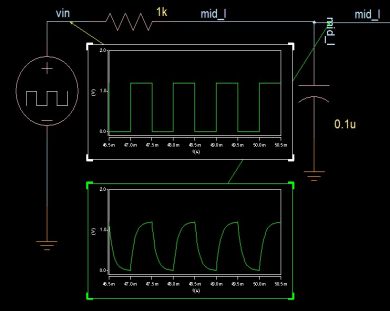
1. **滤波器运算**

前一章我们讨论了图像的像素运算，像素运算可以概括为v’ = f(v) 既每个像素是原像素值的某种映射。新图象上的点，完全取决于映射关系与原图对应位置的点，由于有这种特性，像素运算又叫做**点运算**。点运算可以改变一幅图的亮度，对比度，但是却不能描述相对关系。比如使图像变得模糊，锐利，或者凸显图像的边缘等。为了满足这样一些需求，我们来学习本章的知识。

**认识滤波器**

说到滤波器，似乎对于很多人都是一个虽不陌生，但充满厌恶的词，然而它却充斥在生活中的各个角落这里我们就从物理，机械与人的认知的角度分别讨论下滤波器，经过下面的阅读你会发现，滤波器并不是那么冰冷的概念，而是一个切实存在，并且处处产生作用的信号加工系统。

示波器中方波经过电容 山地车骑行在沙石路上 人的思维惯性

**模拟滤波器**

模拟滤波器是根据电子元器件的特性，对他们进行适当的组合，从而达到整流的目的。比如我们高中学过的，电容会阻断直流，但可以通交流。而电感（线圈）可以通过直流或低频，但当频率升高，电磁感应产生的内阻增加，会阻断交流。而滤波其实就可以理解为有一个波形的电流，通过一个或多个原件之后，以另一种的形式出来。

**机械滤波器**

如果前面的物理知识还是感觉颇为遥远，那么这个生活尝试，大家一定都可以理解。当洗衣机甩干时，会剧烈抖动，一个简单有效的办法时在洗衣机与地面之间垫上一层胶皮垫。这样能达到很好的隔振效果。同理自行车在钢圈外要加一层充气轮胎，也是为了不让地面的崎岖不平直接的作用于车子，而是通过轮胎。轮胎把来自于地面的不平产生的抖动进行了整合，之后再传递给车子。

**人的思维惯性**

其实滤波不仅仅是物理，机械现象，甚至是我们的思想，也是自带了滤波系统的。我们经常说，他竟然能做出这种事，简直不可思议。是的，我们人并非看到什么就是什么，一切都已经经过了大脑的整合。比如一个好学生，考试成绩稳定在90分以上，这次忽然考了70分，那么了解他的人一定会认为这次他发挥失常了，或者这次身体不舒服，状态不好。成绩就是他输出的波形，代表的他的学识，然而我们人也会根据固有经验去判断，用经验去修正他的实际成绩。这是因为我们用思想已经对他的成绩做出了整合。

**数字滤波器与卷积**

以上介绍了各种形式的滤波器，其实广义上的滤波器，就是把一个信号转化成另一个信号的系统，并不局限于任何形式，而接下来我们讨论数字滤波器。

**差分系统**

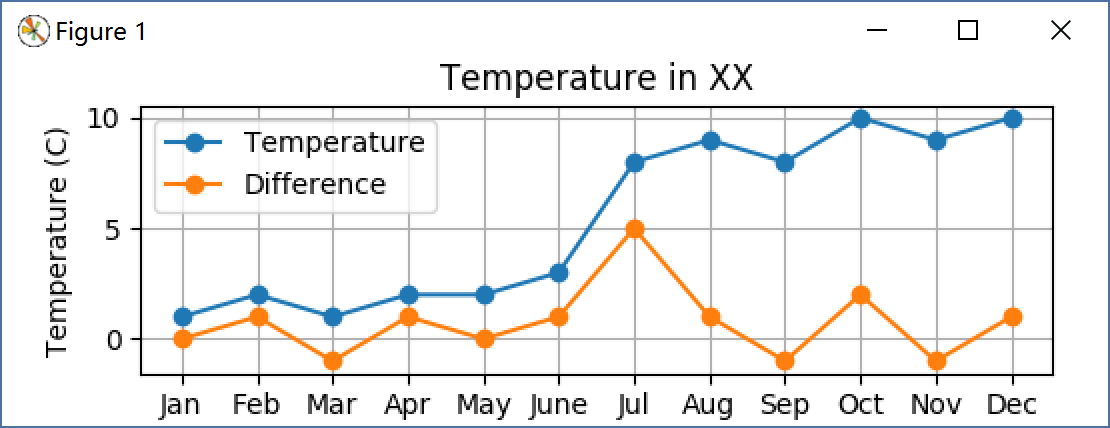
我们之前讨论过一年当中的温度，那现在我们如果问温度上升最快的是几月份？处理这个问题，我们是要用每个月的温度减去上个月的，得到一个所谓的递进温差，由于成都一年四季温差不大，这里我们选择另一个城市，暂且不管温度的合理性，或许不是地球上的城市吧。

**XX市某年的各月平均气温以及与上个月的温差**

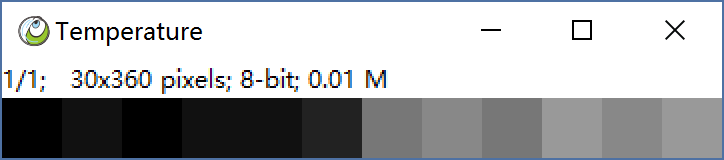
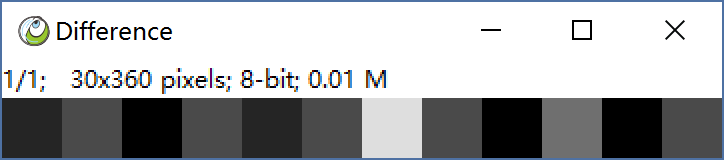
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1月** | **2月** | **3月** | **4月** | **5月** | **6月** | **7月** | **8月** | **9月** | **10月** | **11月** | **12月** |
| **1℃** | **2℃** | **1℃** | **2℃** | **2℃** | **3℃** | **8℃** | **9℃** | **8℃** | **10℃** | **9℃** | **10℃** |
|  | **1℃** | **-1℃** | **1℃** | **0℃** | **1℃** | **5℃** | **1℃** | **-1℃** | **2℃** | **-1℃** | **1℃** |

我们用每个月的温度减去上个月的温度，得到了温差，而温差又构成了一个新的图像。

**$ IBook > Chapter4 Digital-Filter > Temperature Difference**



温度曲线和温差曲线

温度图像 温差图像

**观察描述:**橙色线为温差线，当橙色线在0以上，代表这个月在升温，在0以下，则代表这个月降温了，从图上可知六月 - 七月之间升温很剧烈。而从图像上看，温度图像多少有点左黑右白的意思，而温差图像只有一个点比较亮，这个点恰好是作图黑白的分界点（对于二维图像我们习惯叫做边缘）。

**线性系统与卷积**

以上每个位置对前一个位置做差的运算叫做差分，类似于连续函数的求导。那么如何用一种数学运算来描述它呢？我们先对上面的差分系统做一个描述：

1. 新的图像是一个元素一个元素算出来的（滑动）
2. 每个元素是原图对应位置及周围元素进行某种运算得到

带着这样的描述，我们给出卷积的概念：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | **12** | **16** | **22** | **26** | **28** | **30** | **30** | **25** | **21** | **16** | **11** |

\* \* \* 对应位置相乘 \* \* \*

|  |  |  |
| --- | --- | --- |
| **-1** | **1** | **0** |

|  |  |  |
| --- | --- | --- |
| **-1** | **1** | **0** |

----------- 滑动 ---------->

(11\*-1)+(12\*1)+(0\*16) = 2 相乘后求和 (30\*-1)+(25\*1)+(21\*0) = -5

V …… V

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **x** | **2** | **4** | **6** | **…** | **…** | **…** |  | **-5** |  |  | **x** |

以上是卷积的形象描述，中间的滑动窗口叫卷积核，为了简单明了，我们选取了一维形式，并且滤波核选用了长度为三的最简形式，但实际上针对二维，或高维也可以进行卷积运算，滤波核可以是任意尺寸，此外倦意运算一般要对滤波核进行一个反转，但对于我们甚至可以忽略。下面解释几个问题：

1. 为什么滤波核右边要加一个0

因为一般来说卷积核尺寸一般是奇数，从而明确中心元素的，由于卷积窗口是一块区域，而求和后必须赋给新的图像的某个元素，中心元素就是与新图像目标对应的那个元素。

1. 为什么最左边，最右边没有计算结果

由于窗口是滑动的，因此可知，当中心元素对准最左边时，滤波核左边的-1就没有原图的像素与之对应。但这样一来卷积的结果就变短了，有时这不是我们希望的，因而又有几种模式处理这个问题，具体如下：

**[ 10 ], 10, 12, 16 …… 用原图最边缘的元素填充**

**[ 12 ], 10, 12, 16 …… 用原图边缘的镜像填充**

**[ 11 ], 10, 12, 16 …… 从另一侧截取需要的长度填充（类似周期性）**

我们看到，卷积是滤波核滑动经过原始信号的每一个位置，脉冲（加，乘）得到一个新的结果，赋值给新图上滤波核中心元素对应的元素。由于加，乘全部是线性运算，因而可以称卷积运算是一个线性系统。

**数字滤波的物理意义**

**减震模拟 $ IBook > Chapter4 Digital-Filter > Shake Damping**

我们用一个摆动序列来模拟来自于地面的一个震动，震动开始很大，振幅达到10，然后正负摆动，在阻尼作用下渐渐平息。

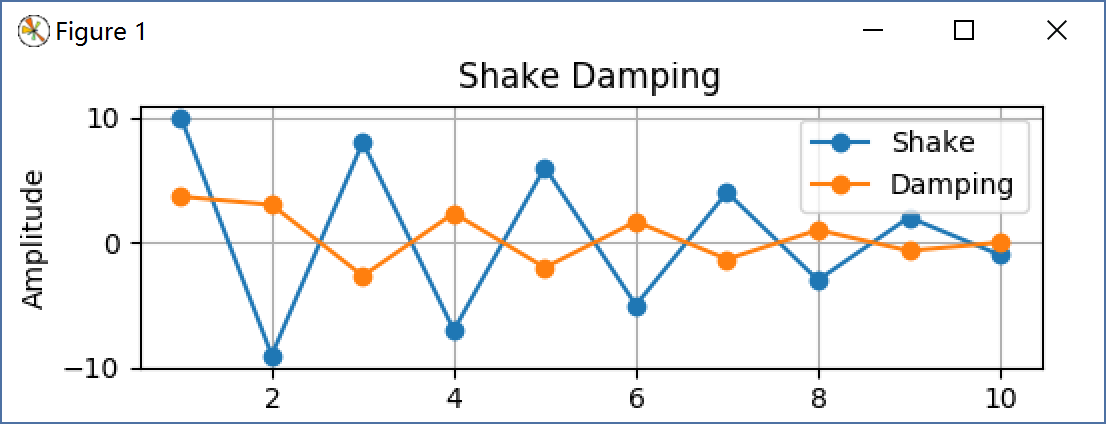
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | **-9** | **8** | **-7** | **6** | **-5** | **4** | **-3** | **2** | **-1** |

\* \* \*

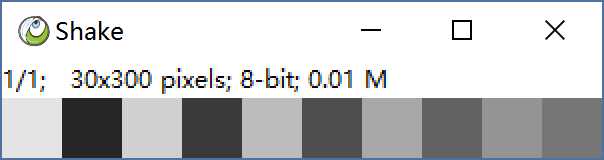
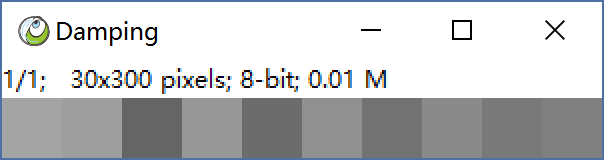
|  |  |  |
| --- | --- | --- |
| **1/3** | **1/3** | **1/3** |

用三个1/3作为滤波核，对原始信号进行卷积

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3.67** | **3** | **-2.67** | **2.33** | **-2** | **1.67** | **-1.33** | **1** | **-0.67** | **0** |



原始信号与滤波信号的曲线对比

原始信号图像 滤波信号图像

从常识上讲，轮胎起到减震作用，专业的说法叫缓冲，因为轮胎是一个弹性介质，可以把冲击转化为势能存储起来，类似于电容，轮胎是一个（力容器）。这样就不会让冲击力直接作用给自行车身，而是存储在轮胎中，进而传递给车身，而这种传递不是实时的，而是具有一定滞后，比如瞬间的挤压，然后快速松开，这个过程轮胎自身完成了一次（充放电），而传递给车身的是融合后的综合作用，我们这里用一个均值滤波器来模拟轮胎对震动的缓解，虽然与物理现实仍有差距，但可以看出数字滤波是如何用卷积完成所谓缓冲和减震的。

**思维惯性 $ IBook > Chapter4 Digital-Filter > Psychological Inertia**

以下是一个学生历次考试的成绩

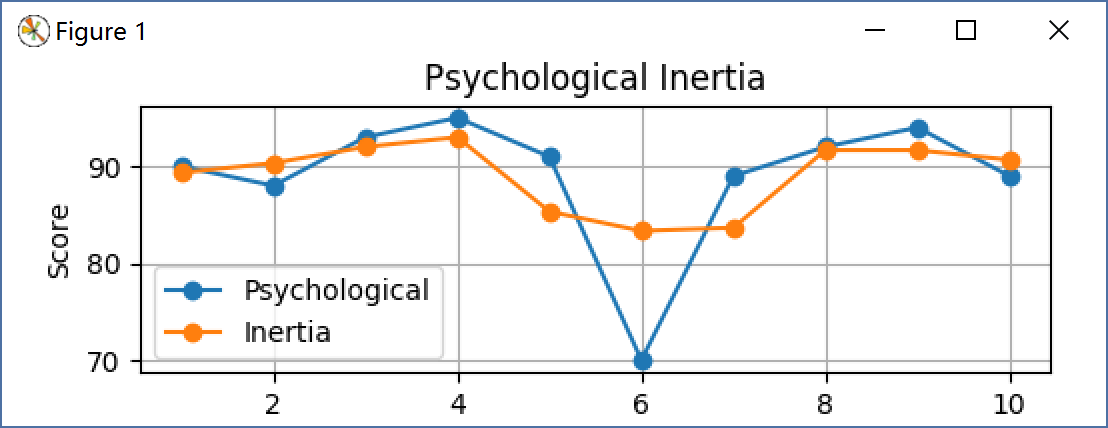
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **90** | **88** | **93** | **95** | **91** | **70** | **89** | **92** | **94** | **89** |

\* \* \*

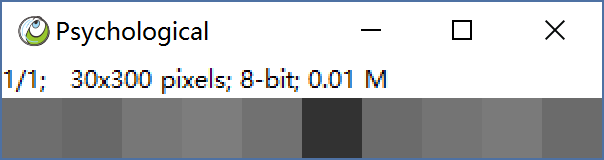
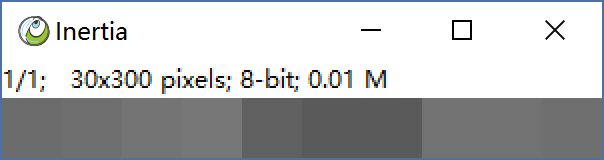
|  |  |  |
| --- | --- | --- |
| **1/3** | **1/3** | **1/3** |

用三个1/3作为滤波核，对原始信号进行卷积

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **89** | **90** | **92** | **93** | **85** | **83** | **84** | **92** | **92** | **91** |



原始成绩与惯性成绩对比结果

原始成绩 惯性成绩

**观察描述：**这个学生历次成绩都稳定在90分左右，所以我们相信这是他的真实水平，而中间出现的70分，我们宁可相信他这次发挥失常，而不会直接下定论说他已经沦为差学生，这里就是思维惯性在起作用，而一个均值滤波，也的确很大程度上可以削弱某一次对结果的影响，因此均值滤波一定程度上可以模拟思维惯性。

**对于其合理性的讨论**

有读者会问，本来是真实的成绩，这样处理之后，还能体现学生真实水平吗？这是不是不尊重事实？首先能提出这个问题说明你是严谨的，不过试想，小学时老师都教过你，测量一段距离，多次，然后取平均值。为什么要取平均？另一种解释是这段距离本身随时在变，这样不更尊重事实吗？实则不然，我们除了测量之外，还有先验知识，那就是我们认定这段距离是不变的，其次我们知道测量是带有偶然误差的，那么我们要做的其实是通过有限的测量更好的估计这段距离，在正太分布的前提假设下，平均值其实是线性最优估计（其实简单一个平均数，却一言难尽，道出了极大似然估计，最小二乘法的本质）。那么看我们的问题，某次考试其实也知识对学生真实水平的一次测量，其中不可避免的带有误差，而另外，我们也应当相信，一个学生的水平在一定时间内是不会变化太大的，因而做了卷积之后的结果，某种意义上，更能体现学生的真实水平。其实很多对高考的质疑也源于此，也曾有人呼吁用一个更综合的考察方式来对学生做一个更综合的评价。

**经典滤波器介绍**

在此之前我们用了大量的篇幅来认识滤波器，这是因为滤波器和卷积在图像处理中是一个非常重要的概念，我希望读者阅读到这里的时候，都已经能够很好的认识滤波器与卷积运算，而接下来的学习我们来见识一些常见，经典的滤波器，并且认识其效果，更重要的是试着建立一种感性的认识，并且在面对图像处理任务的时候，知道自己需要凸显哪些信息，需要怎样的一种滤波器。

**均值滤波器**

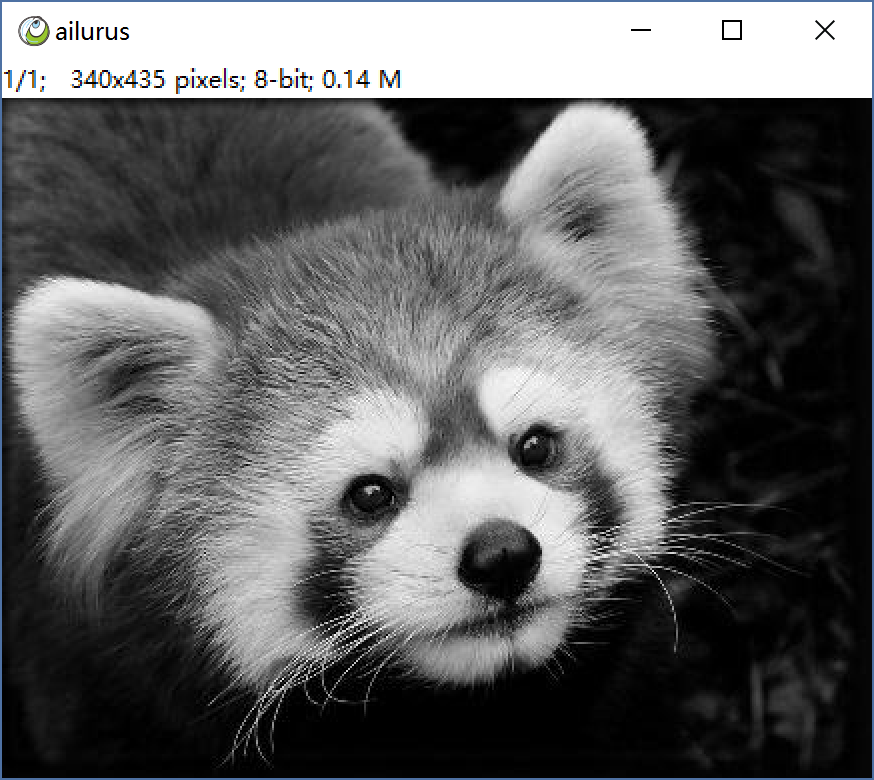
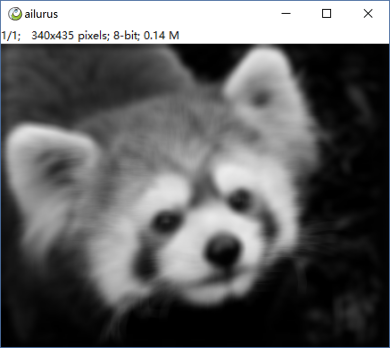
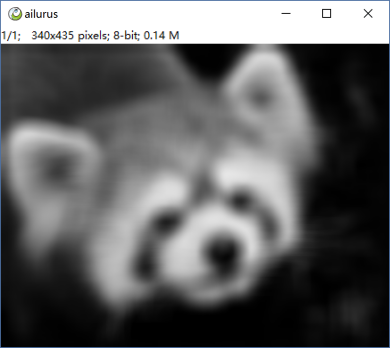
**滤波核 $ Process > Filter > Uniform**

均值滤波器是用一个相同元素构成的目标对原始二维图像进行卷积。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 1/9 | 1/9 | 1/9 | | 1/9 | 1/9 | 1/9 | | 1/9 | 1/9 | 1/9 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 1/25 | 1/25 | 1/25 | 1/25 | 1/25 | | 1/25 | 1/25 | 1/25 | 1/25 | 1/25 | | 1/25 | 1/25 | 1/25 | 1/25 | 1/25 | | 1/25 | 1/25 | 1/25 | 1/25 | 1/25 | | 1/25 | 1/25 | 1/25 | 1/25 | 1/25 | | |  |  |  | | --- | --- | --- | | 1/n2 | ... |  | | ... |  |  | |  | ... |  | |  |  | ... | |  | ... | 1/n2 | |

3X3 均值模板 5x5 均值模板 nxn 均值模板

**$ IBook > Chapter4 Digital-Filter > Show Uniform Filter**

原图 均值滤波 r=10 均值滤波 r=20

**参数：**均值滤波器有一个参数，窗口半径r，而滤波核尺寸是2\*r+1。因此r=10其实是一个21\*21的窗口。

**观察描述：**均值滤波可以用于图像的平滑，原图中毛发，胡须清晰可见，而第二幅图中胡须就看不见了，第三幅图中甚至只能隐约分辨鼻头，眼睛。

**高斯滤波器**

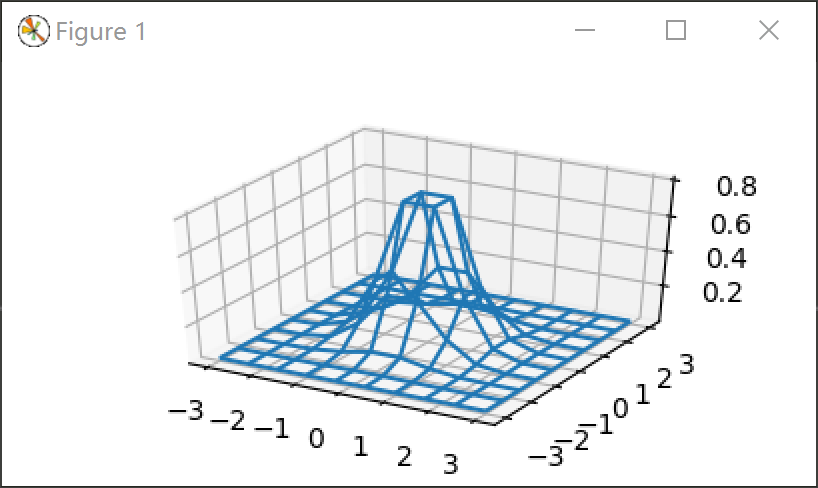
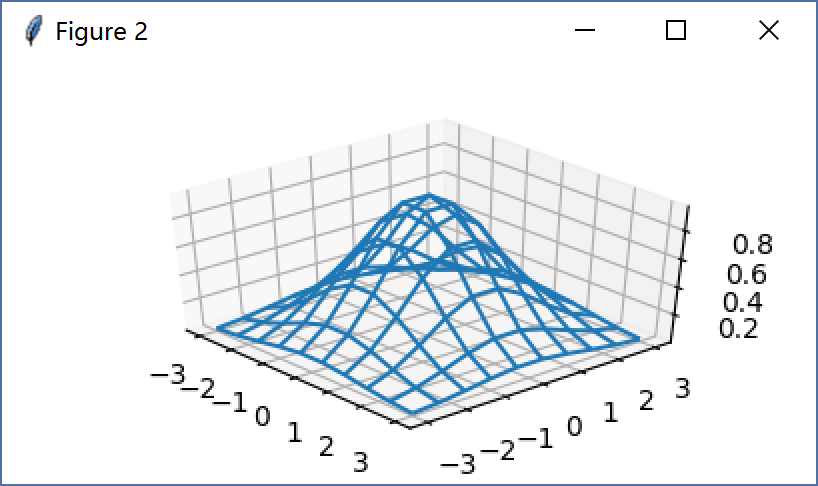
**滤波核 $ Process > Filter > Gaussian**

不同于均值滤波器，网格内像素等权平均，高斯滤波器是用一个二维正太分布作为卷积模板。

**$ IBook > Chapter4 Digital-Filter > Gaussian Core**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  | |  | 0.008 | 0.072 | 0.008 |  | |  | 0.072 | 0.682 | 0.072 |  | |  | 0.008 | 0.072 | 0.008 |  | |  |  |  |  |  | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 0.002 | 0.011 | 0.019 | 0.011 | 0.002 | | 0.011 | 0.059 | 0.103 | 0.059 | 0.011 | | 0.019 | 0.103 | 0.188 | 0.103 | 0.019 | | 0.011 | 0.059 | 0.103 | 0.059 | 0.011 | | 0.002 | 0.011 | 0.019 | 0.011 | 1/25 | |

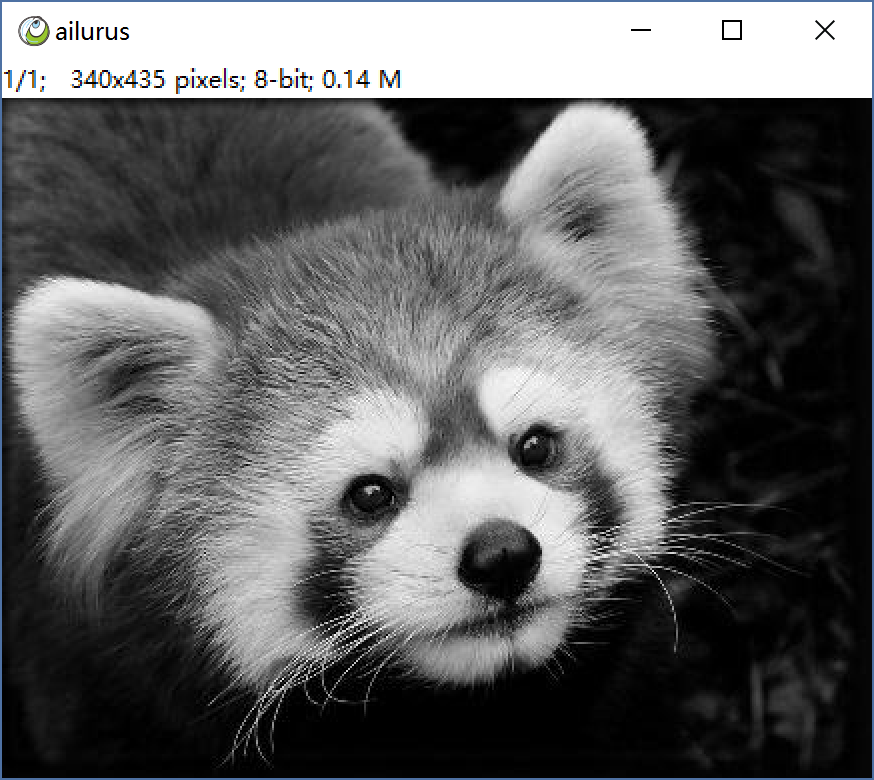
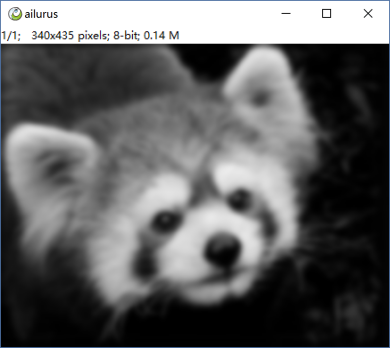
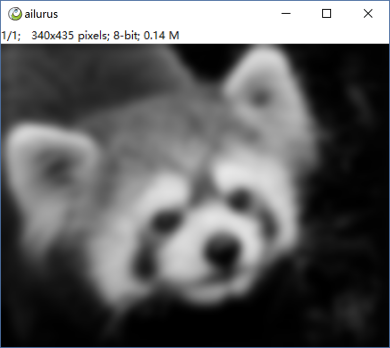
Sigma = 1的正太模板 Sigma = 2的正太模板

Sigma = 1的三维图表 Sigma = 2的三维图表

以上是为了便于展示，简化了表格，一般来说正态分布在大于3倍sigma的区域，概率分布近似为0，因而模板大小应该为6 sigma + 1。

**$ IBook > Chapter4 Digital-Filter > Show Gaussian Filter**

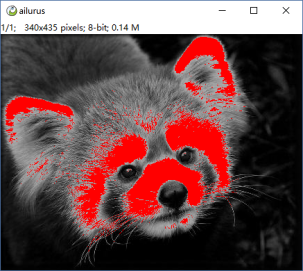
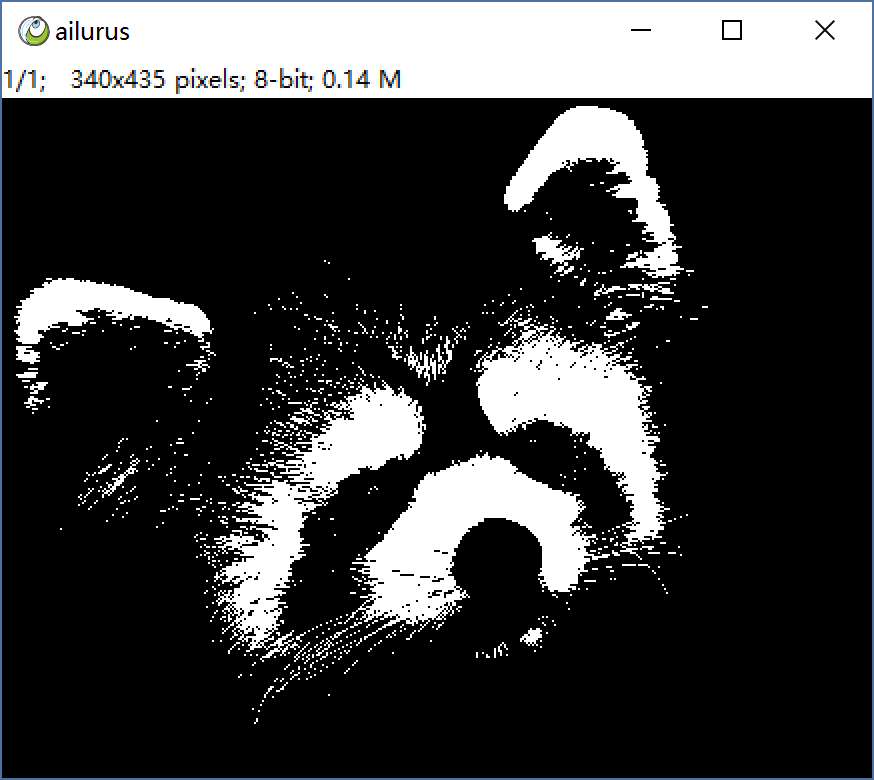
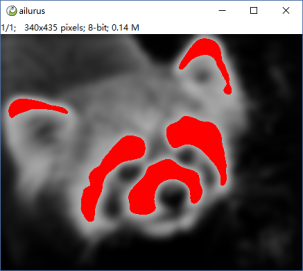
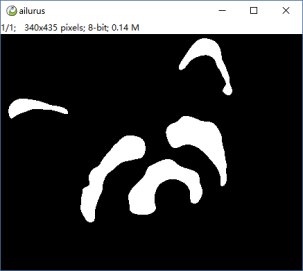
原图 高斯模糊sigma = 3 高斯模糊sigma=5

视觉效果上或许均值滤波器和高斯滤波器会有些类似，毕竟他们同属于低通滤波，都会产生模糊作用，甚至两者还颇有渊源（多次的均值滤波器可以认为是高斯滤波器），而高斯滤波器也是一个非常重要的滤波器，我们下面做些相关讨论。

**为什么要把图变模糊**

很多读者会问，清晰点不是挺好的，谁希望弄模糊呢？这是一个非常好的问题，正如我们之前讨论的一个学生的成绩，我们已经认可了，平滑过后的成绩更能够代表真实情况，因为平滑之后屏蔽了局部细节，进而更能够凸显整体特性。比如我们现在要做的一件事，观察小熊猫的脸上，有几片白色绒毛？

**$ IBook > Chapter4 Digital-Filter > Show Smooth And Threshold**

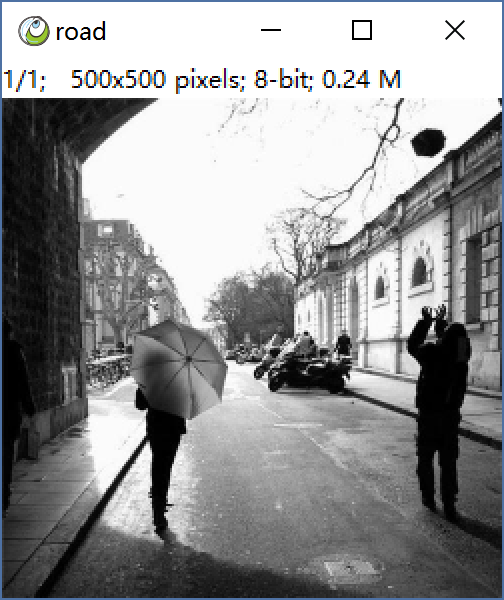
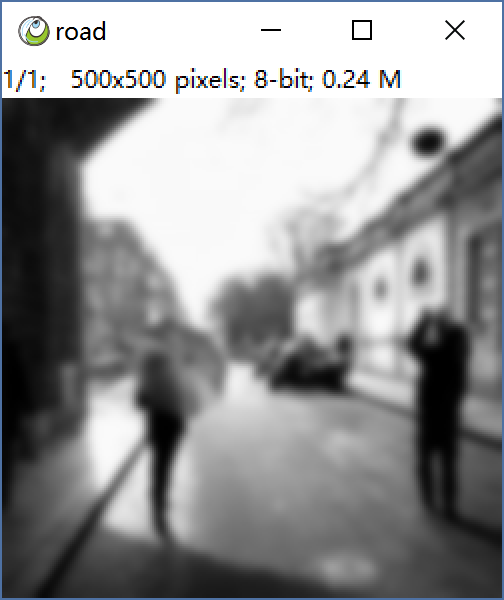
原图阈值效果 高斯模糊后阈值效果

**观察描述：**我们看到原图由于有各种毛发，胡须的干扰，我们很难用阈值分割出理想的结果，而模糊过后，抑制了细节信息，反而凸显了更大尺度的块面信息。正可谓一叶障目不见泰山，而高斯滤波正是去除树叶的方法。

**自然界中的高斯模糊**

其实说到高斯模糊，大家可能觉得陌生，但是看效果一定很常见，成都是一个多雾的城市，雾气中一切是朦胧的，这与我们介绍的高斯模糊有很多相近之处（其实本质完全一样）。

**$ IBook > Chapter4 Digital-Filter > Show Gaussian In Nature**

道路，行人 雾气笼罩的街道

下面我们用概率知识来解释以下为什么下雾的天气一切看起来是朦胧的，（即便在晴朗的日子里，远处的山也是依稀的，也就是美术表达技巧中的近实远虚）。

**钉子概率与卷积**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| u=2533447880,1732479788&fm=27&gp=0[1] |  |  |  | 1 |  |  |  |
| **[0.5, 0.5] 卷积** / \* \ **每一层都是** | | | | | | |
|  |  | 0.5 |  | 0.5 |  |  |
| / \* \ / \* \ | | | | | | |
|  | 0.25 |  | 0.5 |  | 0.25 |  |
| / \* \ / \* \ / \* \ | | | | | | |
| 0.125 |  | 0.375 |  | 0.375 |  | 0.125 |

钉子墙 一颗球的命运

我们看，左图是很多颗球经过钉子墙后落下来的累计状态，而右侧是一颗球落下的概率，在经历第一层的时候，经历了一个左或右的抉择，因此在第二层到达概率分别是0.5，0.5，而到达第二层，就经历了两次抉择，能到达左边意味着之前两次都选择了左，0.5\*0.5=0.25，右边亦然，而中间却可以是左右，右左两种可能。因而最终到达概率，可以用一个排列组合表示，C(N,M)。我们可以推测，**当层数足够多，到达的概率就趋于正态分布（大数定律）**。而更有趣的是，我们观察，**这种层之间的概率传递，恰好可以用一个[0.5,0.5]的卷积来表示**（暂且忽略灰色的格子，这是受到物理限制）这意味着每一层之间都是一个均值滤波。

或许对于球我们不得不用排列组合思考，但事实上我们完全可以把球按照概率打碎，如果是光线我们就更容易理解，光束到达第二层时就已经被分成两个0.5的子光束，进而又分成四个0.25，但中间两个0.25又融合成0.5...那么这与雾天的朦胧景象又有什么关系呢？一个物理常识，**当光线穿过两个两个密度不同的介质时会发生折射，方向偏移，而大气是一个随机介质，各处密度都有细微的差别，其实光线每前进一步都经历了一个向左向右的抉择，最终到达眼睛的时候，经过了无数抉择，成为一个概率意义的到达，且符合正太分布。**

**为什么正太分布如此普遍**

我们都知道一个运动员的成绩，一个学生的考试往往都稳定在一个值附近，偏离很多的可能性不大，也就是说成正太分布。那其背后的原因是什么呢？很多事情都可以理解为一次测量行为，比如运动员的成绩，主要取决于其真实水平，但是又会受到很多次要因素的影响，比如最近的训练是否系统，早餐吃得是否营养，今天的体育场是逆风还是顺风，以及临场发挥等等。。。这些因素都会对其结果产生一个正的或负的影响，其最终成绩是真实水平加上所有次要因素的总和，而**众多次要因素的总和成正太分布！**如果某次意外的发生了远低于正常水平成绩我们会说背到极点了，反之也会说运气好到爆，这些结果不经常发生的原因是恰好今天早餐营养丰盛，赛场顺风，自己荷尔蒙兴奋，一切因素都朝着有利的方向作用，就像那个钉子墙，每次都选择了向右，而概率意义上，这样的情况很少发生，更多的是喜忧参半。

**Prewitt与Sobel滤波器**

**$ IBook > Chapter4 Digital-Filter > Prewitt $ IBook > Chapter4 Digital-Filter > Sobel**

Prewitt与Sobel是基于一阶导数的滤波器，也叫差分滤波器，可以凸显图像的变化。

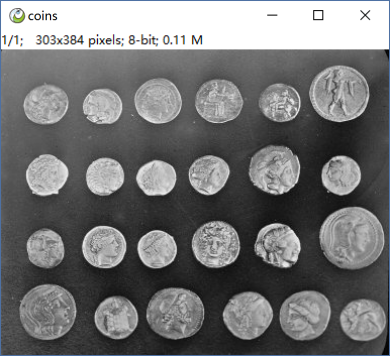
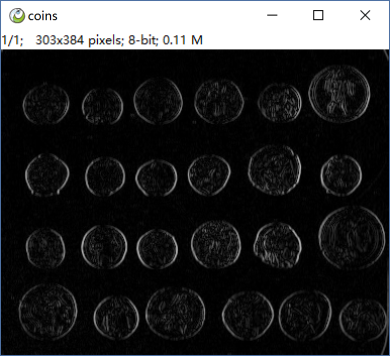
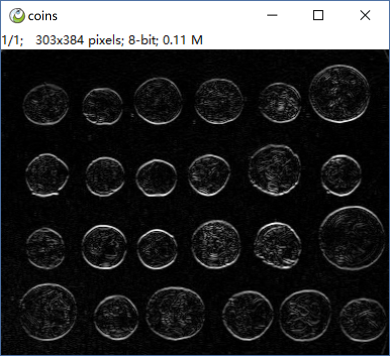
**滤波模板**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | -1 | 0 | 1 | | -1 | 0 | 1 | | -1 | 0 | 1 | | |  |  |  | | --- | --- | --- | | -1 | -1 | -1 | | 0 | 0 | 0 | | 1 | 1 | 1 | |  | |  |  |  | | --- | --- | --- | | -1 | 0 | 1 | | -2 | 0 | 2 | | -1 | 0 | 1 | | |  |  |  | | --- | --- | --- | | -1 | -2 | -1 | | 0 | 0 | 0 | | 1 | 2 | 1 | |

Prewitt Sobel

Prewitt由一对差分形式构成，分别用于检测横向和纵向，而Sobel几乎一致，唯一的不同是中间占据更大的权，以下效果图以Sobel为例，但是梯度是有正负之分的，这里取了绝对值。

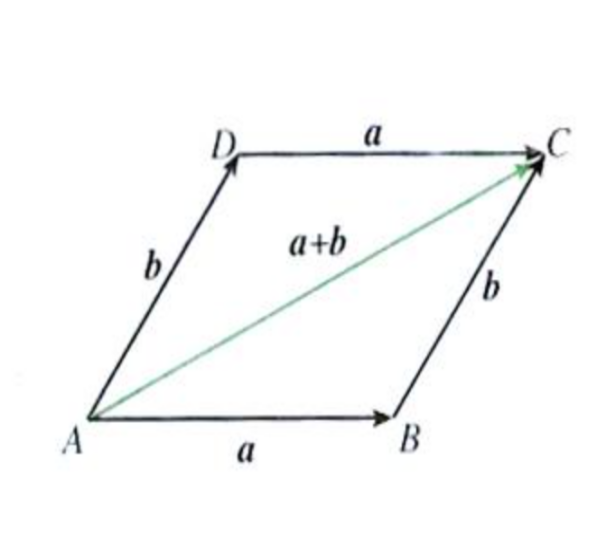
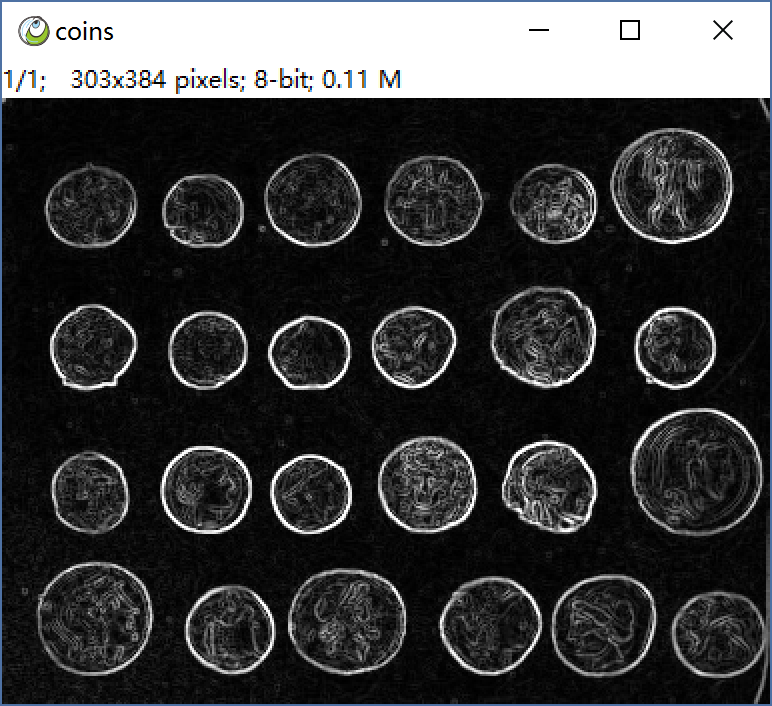
**$ IBook > Chapter4 Digital-Filter > Show Sobel Filter**

原始图像 横向差分 垂直差分

**梯度的方向与值**

我们分别对横向纵向进行了差分运算，得到了两个方向的梯度值，记为(dx, dy)。这样我们其实是得到了一个梯度向量，符合矢量运算法则。我们可以得到梯度的方向，以及梯度值。梯度值可以根据勾股定理得到 dv = (dx\*2 + dy\*2)^0.5。由于平方和开方运算耗费时间较多，因而在精度要求不高的时候，可以用加法得到近似结果。

矢量加法 Sobel两个方向合成图

**拉普拉斯滤波**

**$ Process > Filter > Laplace**

差分滤波器是基于一阶导数的滤波，而拉普拉斯滤波是基于二阶导数的，相比之下二阶导数更为精准，但一定程度上对噪声更敏感。

**滤波模板**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | 0 | -1 | 0 | | -1 | 4 | -1 | | 0 | -1 | 0 | | |  |  |  | | --- | --- | --- | | -1 | -1 | -1 | | -1 | 8 | -1 | | -1 | -1 | -1 | | |  |  |  | | --- | --- | --- | | -1 | -2 | -1 | | -2 | 12 | -2 | | -1 | -2 | -1 | |

几种3X3的拉普拉斯模板

虽然只是一个3\*3的粗略模板，但是可以看出他是一个各项均质的模板，因而对各个方向的变化有相同的响应。

**$ IBook > Chapter4 Digital-Filter > Show Laplace Filter**

原图 拉普拉斯滤波

**观察描述：**可以看出拉普拉斯滤波后细节被凸显了，但同时硬币内部的任何细微变化都被凸显了，这就是所谓的对噪声敏感。

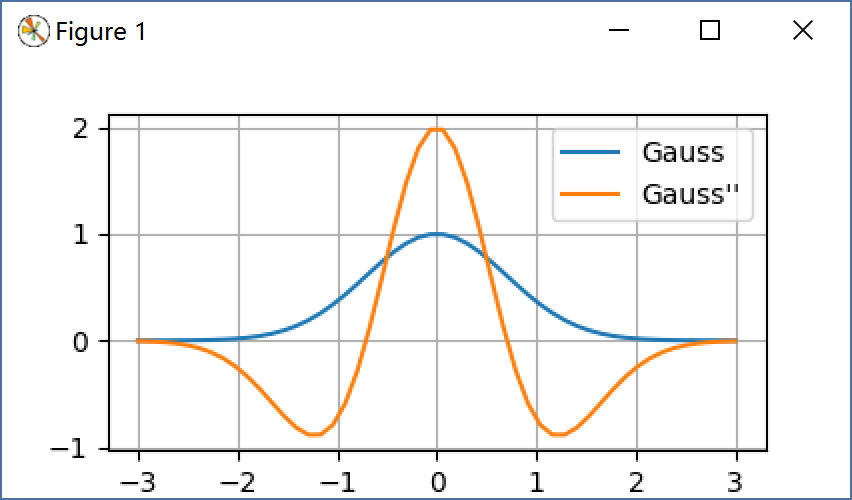
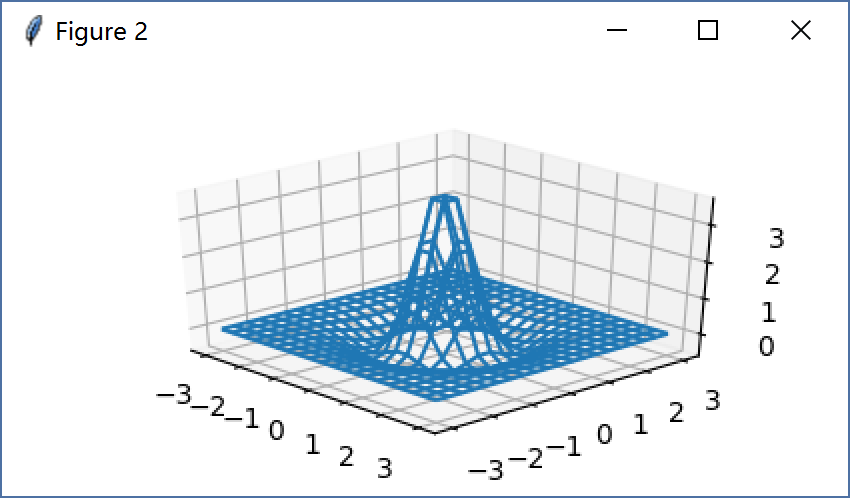
**高斯拉普拉斯**

**$ Process > Filter > Gaussian Laplace**

由于拉普拉斯滤波对噪声非常敏感，针对这个问题，有人提出预先用某种平滑算法处理，再做拉普拉斯滤波，而更多时候我们采用高斯平滑，这样就形成了高斯拉普拉斯滤波。

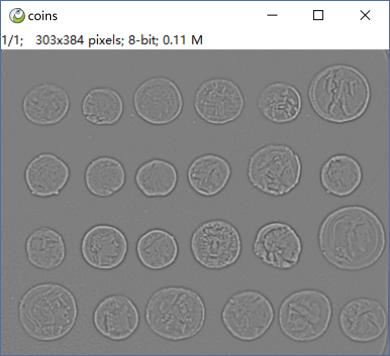
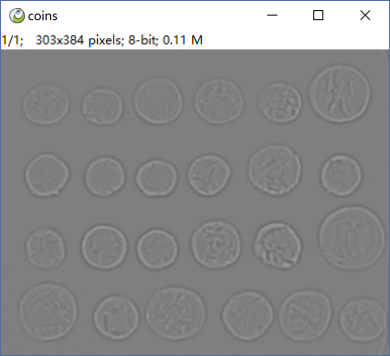
形式上是二维高斯函数的二阶导数，其滤波核图像如下。

**$ IBook > Chapter4 Digital-Filter > Laplace of Gaussian Core**

一维高斯二阶导数 二维高斯二阶导数

**$ IBook > Chapter4 Digital-Filter > Show Laplace of Gaussian**

LoG Sigma = 0.7 LoG Sigma = 1 LoG Sigma = 1.5

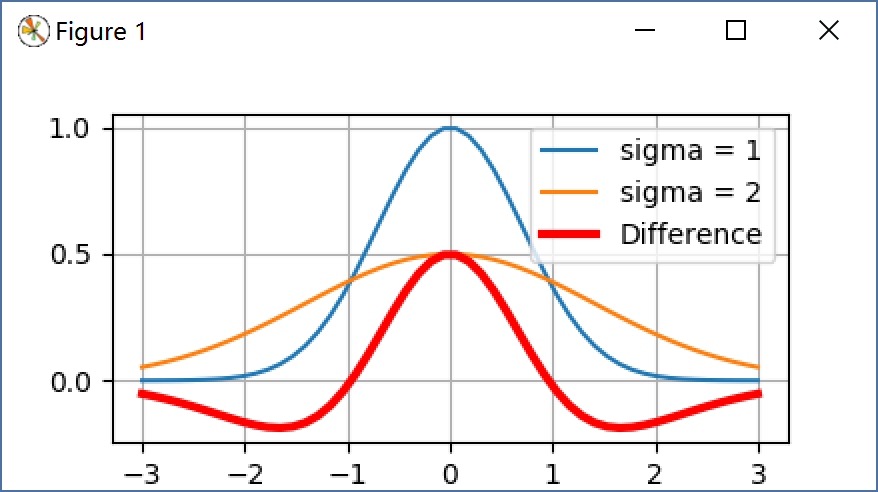
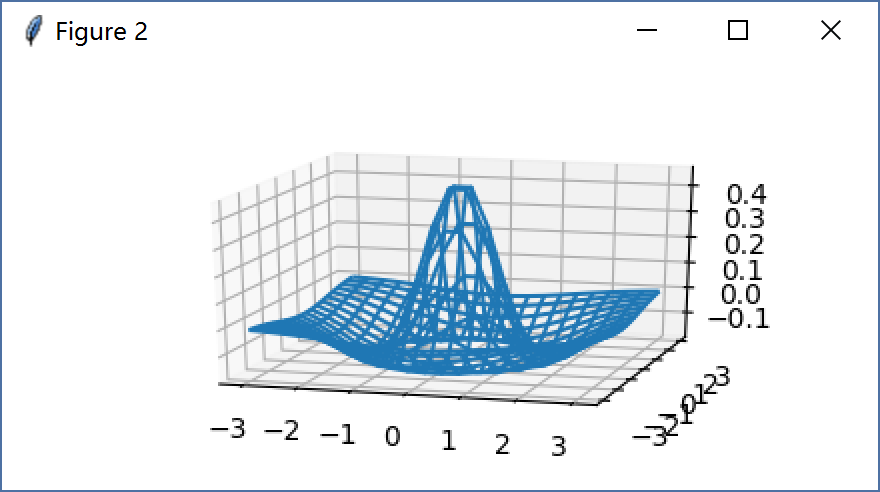
由于高斯拉普拉斯卷积有一个可调节的高斯预处理，因而对结果的可控性增强，如上图，sigma=1的时候细节被凸显，随着sigma的变大，最后几乎只剩下了硬币外轮廓，sigma可以调节凸显的目标的尺度。

**差分高斯滤波**

**$ Process > Filter > DOG**

高斯拉普拉斯是针对高斯函数的二阶导数，而这里介绍的差分高斯滤波器是一种类似的滤波器，值得一提的是，高斯滤波器可以分解成两个一维卷积，因而它的计算性能比高斯拉普拉斯要好。

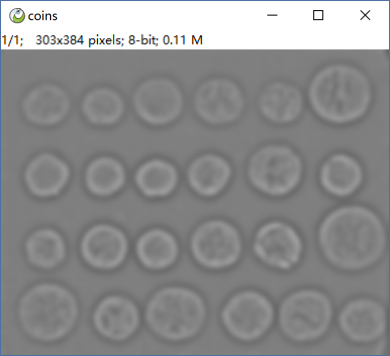
**$ IBook > Chapter4 Digital-Filter > Difference Of Gaussian**

两个不同sigma的高斯函数的差 两个不同sigma的二维正太分布的差

**观察描述：**红色曲线和拉普拉斯函数是类似的，右侧二维网格图也与拉普拉斯函数类似。

**$ IBook > Chapter4 Digital-Filter > Show DOG Filter**

sigma1=0 sigma2=2 sigma1=0 sigma2=5 sigma1=1 sigma2=3

不同的sigma组合能够得到不同的效果，如果sigma1为0，意思是保持原始，而sigma2不为零，意味着原始图像减去一个平滑的图像，可想而知在平坦区域模糊之后不发生太多变化，因而原图和平滑图很接近，只有在变化区域敏感。观察图2可知，sigma2越大，就会把这种对变化区域的影响范围就越大。而sigma1可以一定程度上控制细节。

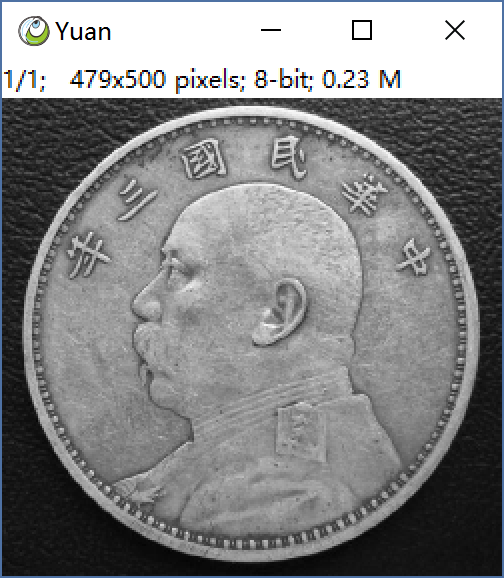
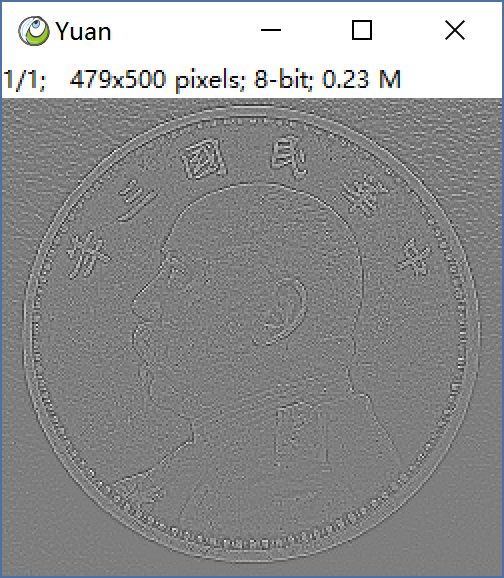
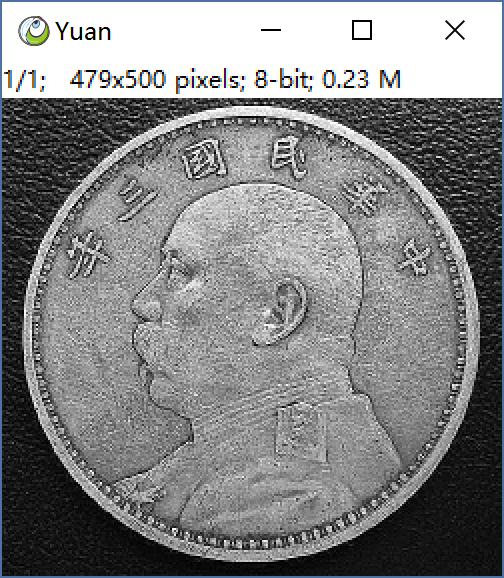
**图像锐化**

前一节我们介绍了基于一阶导数的sobel滤波，基于二阶导数的拉普拉斯滤波，以及其基础上改进的高斯拉普拉斯滤波和差分高斯滤波。以上的滤波器都属于高通滤波器，既凸显图像变化的部分，而在平坦区域响应值为零。而这里我们讨论的锐化，就是凸显图像的细节，使用的方法多是在原图基础上叠加一个差异信号，这样与拉普拉斯信号进行融合的锐化叫拉普拉斯锐化，与差分高斯滤波信号进行融合的锐化叫USM锐化，事实上，如果愿意，可以与高斯拉普拉斯或者任意其他的高通滤波器进行融合。

**拉普拉斯锐化**

**$ Process > Filter > Laplace Sharp**

**$ IBook > Chapter4 Digital-Filter > Show Laplace Sharp Filter**

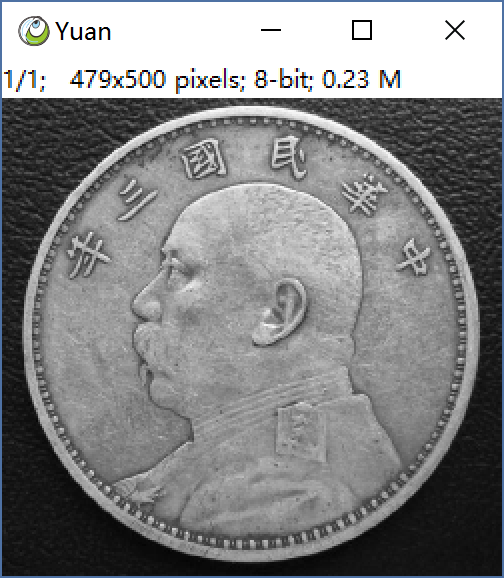
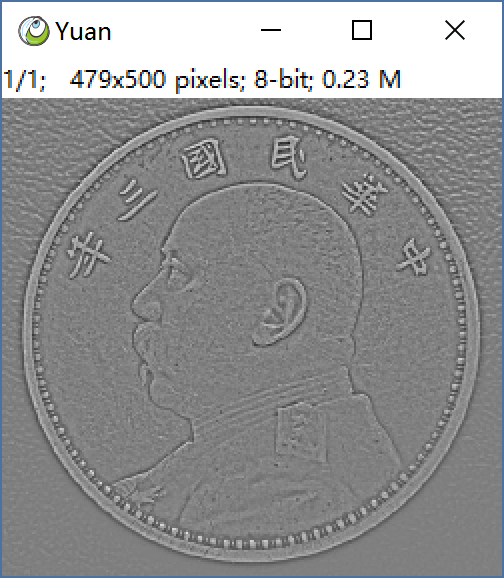
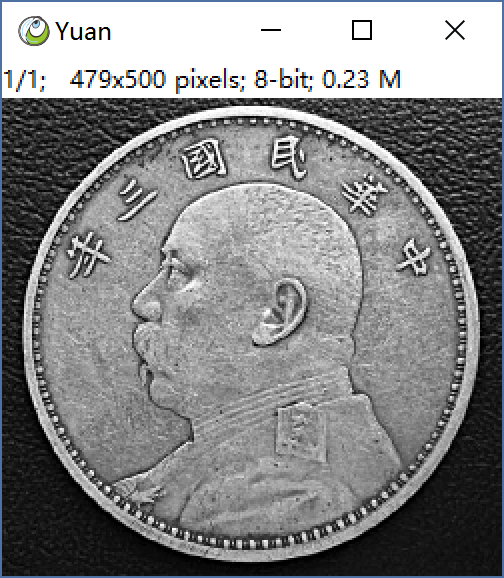
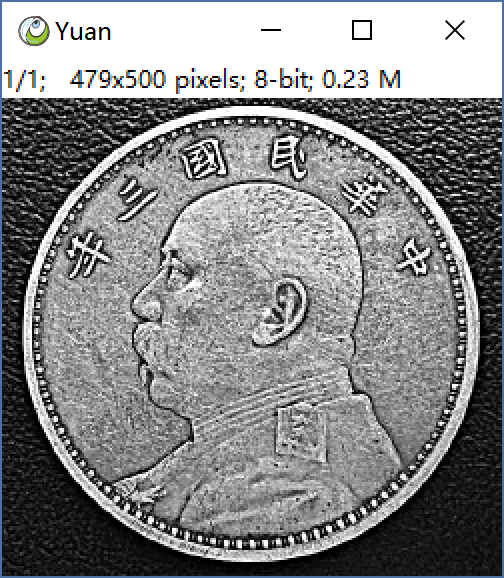
原图 拉普拉斯滤波 原图 + 拉普拉斯

拉普拉斯变换的本质是原始凸显叠加其拉普拉斯滤波结果乘以系数K，K代表增强的程度。上图是一颗民国硬币进行拉普拉斯锐化的结果。

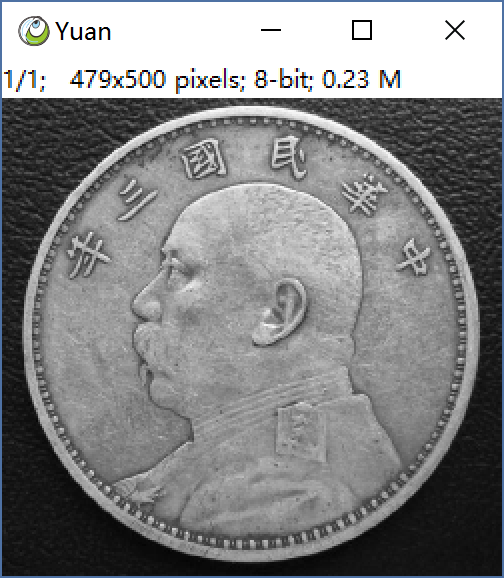
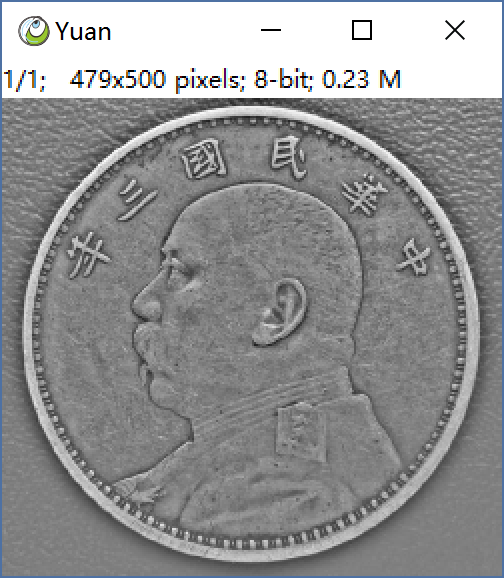
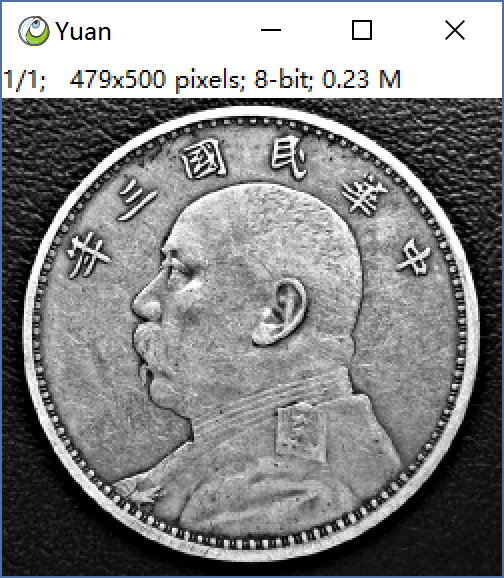
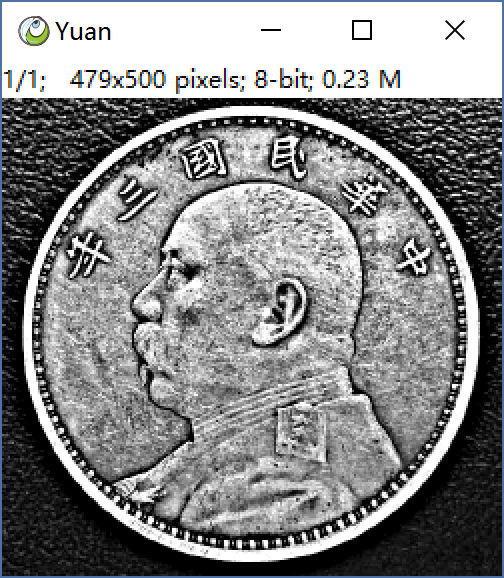
**Unsharp Mask (USM)锐化**

**$ Process > Filter > Unsharp Mask**

**$ IBook > Chapter4 Digital-Filter > Show Unsharp Mask Filter**

原图 原图 - 高斯 sigma=3 原图 + 差异 \* 1 原图 + 差异 \* 3

原图 原图 - 高斯 sigma=8 原图 + 差异 \* 1 原图 + 差异 \* 3

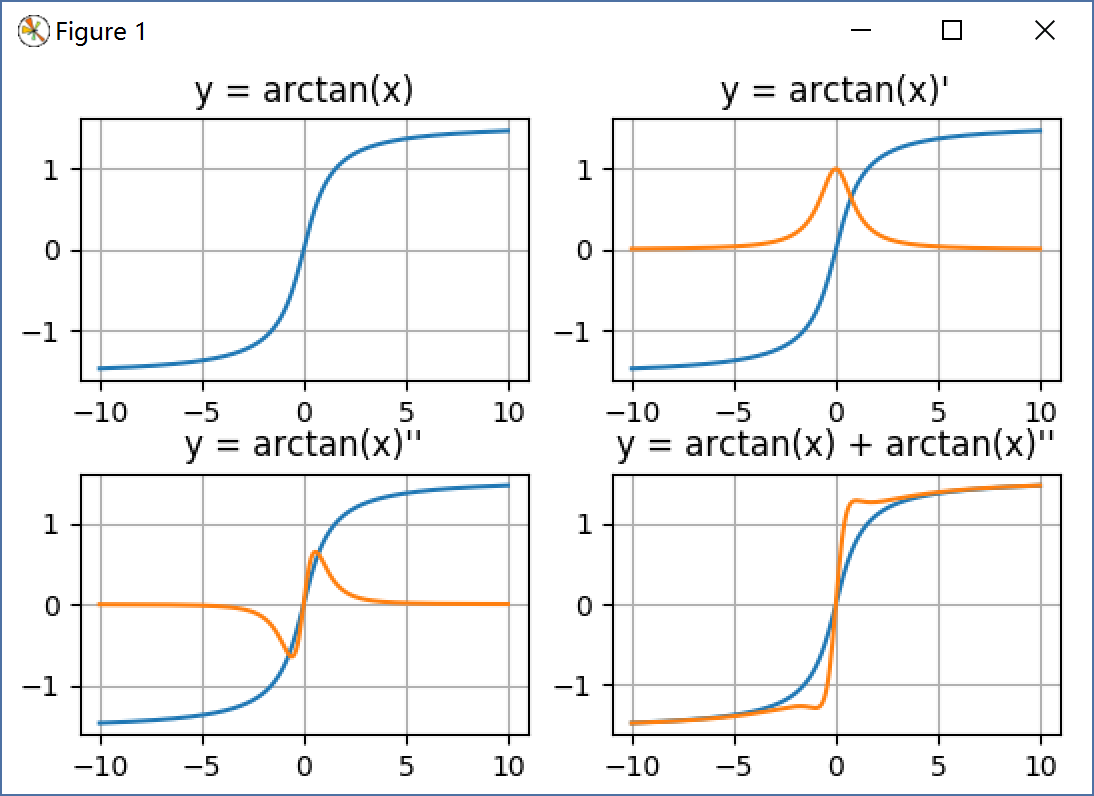
我们知道差分高斯滤波器是两个不同sigma的高斯滤波结果相减，而当sigma1=0的时候，退化成原图减去一个高斯滤波结果，如上图所示，当sigma选取不同时，得到的结果有所不同，sigma越大，变化波及的范围就更大，同样有一个参数K，决定增强的程度。上图例举了两个sigma分别为3和8的情况下，K取1或3的锐化结果，简单概括，sigma越大，变化响应范围越大，K越大，细节凸显程度越强。

**高通，低通，锐化的本质及其关系**

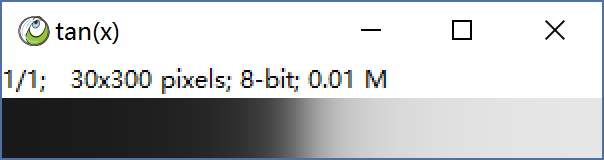
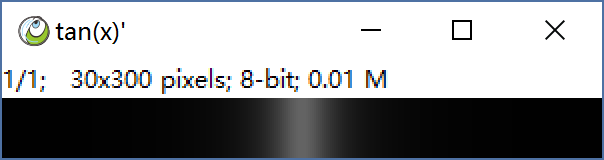
总结以下我们已经学过的滤波器，均值，高斯滤波属于低通，Prewitt，Sobel属于基于一阶导数的高通滤波，拉普拉斯是基于二阶导数的高通滤波。以及改进的高斯拉普拉斯和差分高斯滤波也属于高通滤波。而把高通滤波结果以某个比例叠加到原图像称之为锐化。当学习完这些，我们再次拉通来理解这些滤波器的本质及他们之间的关系。

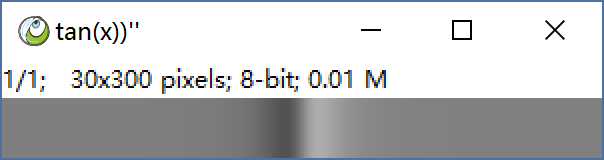
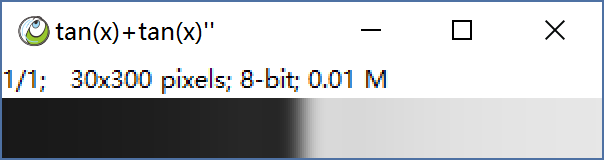
**Arctan函数及其一阶导数，二阶导数和锐化**

**$ IBook > Chapter4 Digital-Filter > Show How to Laplace Sharp**



1. 原始函数 2）一阶导数 3）二阶导数 4）二阶导数叠加回原始函数

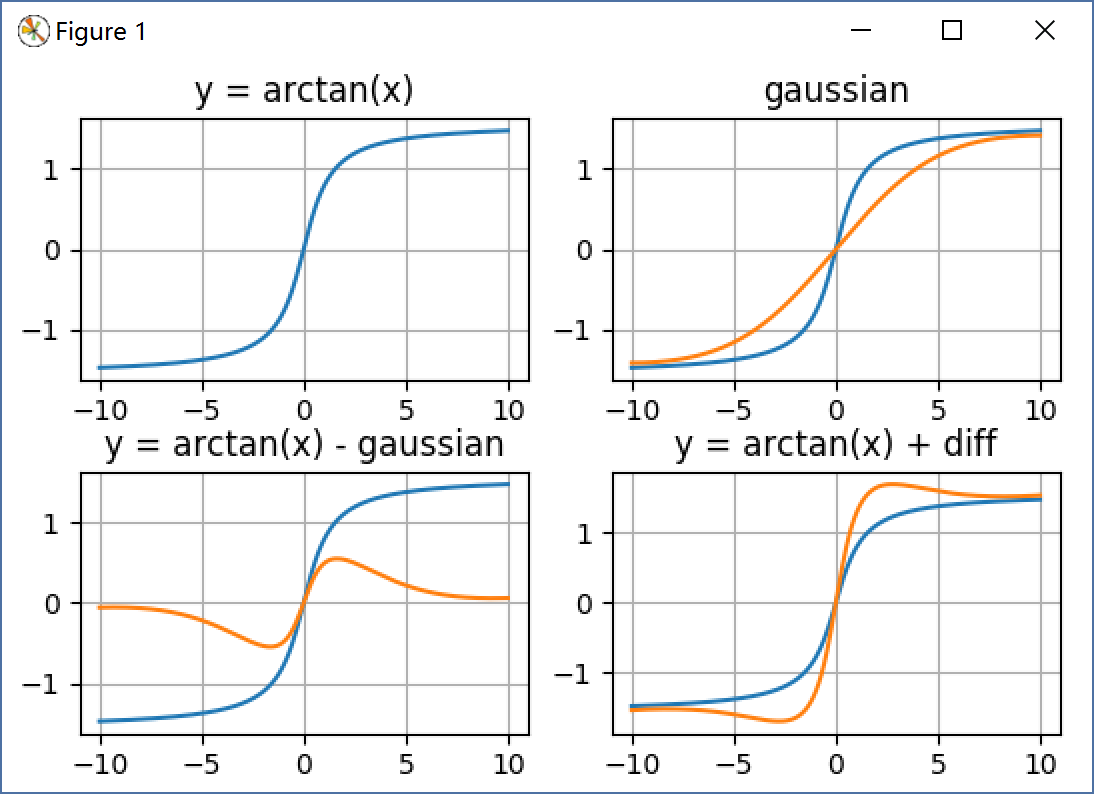
 

1）原始图像 2）Sobel滤波 3）拉普拉斯滤波 4）拉普拉斯滤波结果叠加回原始图像

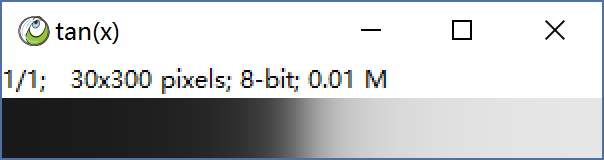
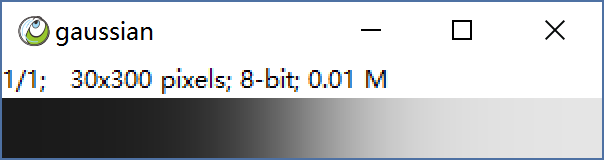
我们看到叠加后的函数曲线上更接近于直角，图像更加黑白分明，视觉效果更加锐利。从上图我们可以清楚看到原图变化最大的点，对应于一阶导数的极值点，二阶导数的过零点（由于屏幕只能渲染0-255，因而整个图像都加了一个128便于显示），二阶导数回在原始图像变化的部位产生一个小块的波浪线，而我们也正是利用这个波浪线来实现的锐化。

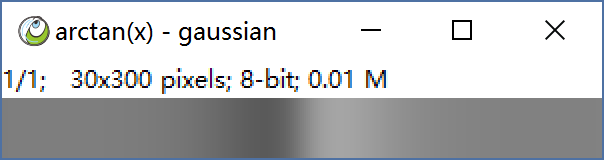
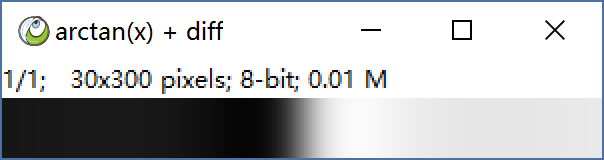
**Arctan函数及其高斯卷积，差分高斯卷积，USM锐化**

**$ IBook > Chapter4 Digital-Filter > Show How to Unsharp Mask**



1. 原始函数 2）高斯平滑 3）原始函数减去高斯平滑 4）差异叠加回原始函数

1）原始图像 2）高斯滤波 3）原始函数减去高斯平滑 4）差异叠加回原始函数

可以看到采用USM锐化结果与拉普拉斯锐化的结果类似，最后也起到视觉效果更加锐利的作用。我们可以看到，高斯滤波本质上是平滑，使得原图的变化不再那样剧烈，而原始信号减去平滑信号，同样也得到了一个类似于二阶导数的波浪线，叠加回原图，达到锐化效果。这里的关系可以理解为未经平滑的信号减去平滑的信号，那么差异就是那些被平滑掉的细节。

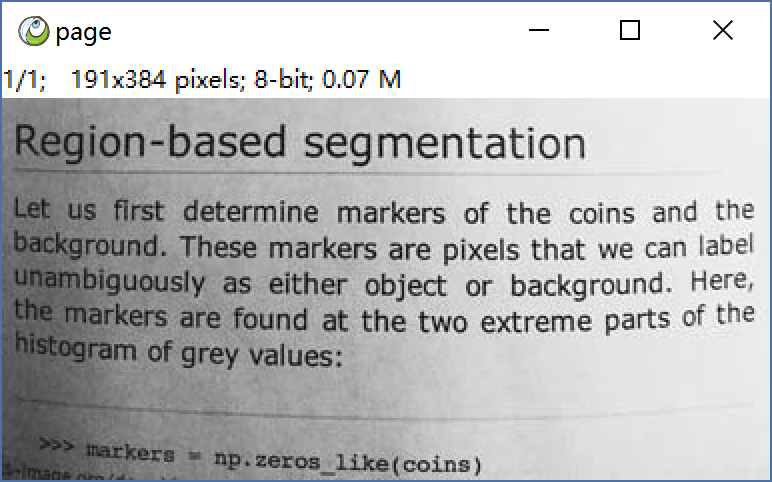
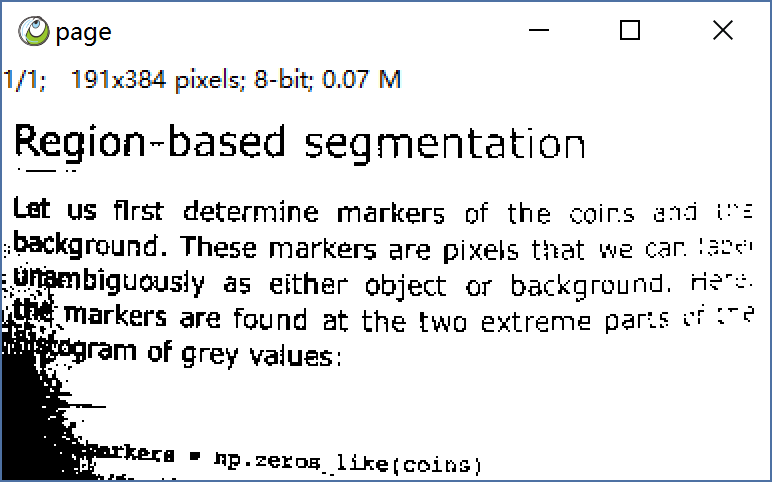
**为什么叫高通，低通**

本章我们不止一次出现了高通，低通这个词语，或许读者已经建立了平滑属于低通，边缘凸显属于高通，这里介绍以下这些名字的意义，从频率的角度来看，变化缓慢的叫低频，而变化快的叫高频。这个严格的说需要用到傅里叶变换，但是本书不打算涉及频谱技术，但不严谨的方式，读者可以这样认为。均质的大块区域变化缓慢，属于低频信息，而边缘，细节变化快，属于高频信息。这样看，均值，高斯滤波都是弱化细节的，可以理解为高频信息被滤掉，低频信息通过了，因而叫低通，反之，梯度，边缘算子变化部分被凸显，因而叫做高通。

**文字提取中的一个应用**

**$ IBook > Chapter4 Digital-Filter > Use Filter To Extract Text**

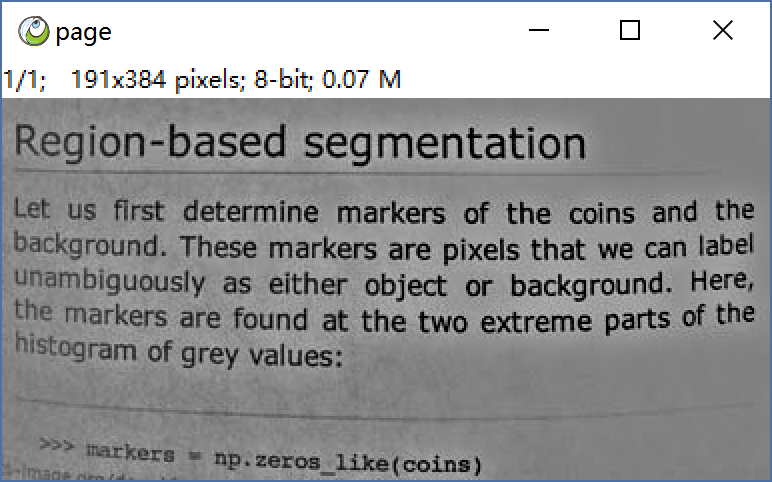
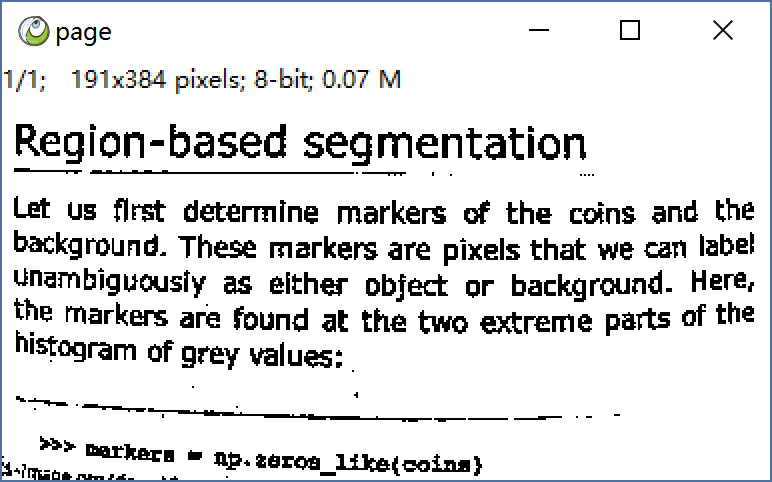
这里是一页书的照片，由于拍摄环境光线不均，导致书的左侧严重偏暗，给分割带来困难。虽然处理这种问题后面我们会介绍到局部自适应阈值分割，但是通过我们本章的知识，一样可以很好的解决问题，并且很简单。

原图 阈值

**观察描述：**由于光照不均，导致背景色不一致，当我们在阈值调整时，左侧的背景还没有去除，而右侧的文字已经丢失了。

**分析：**这张图由于光照原因，使得无法用统一阈值分割，但是视觉上我们时可以感知文字的，因为我们并不只是通过它的绝对亮度来判别，人的肉眼会通过局部对比关系来识别。而我们本章学习的高斯拉普拉斯，或者差分高斯滤波正是一个凸显局部变化的滤波器。

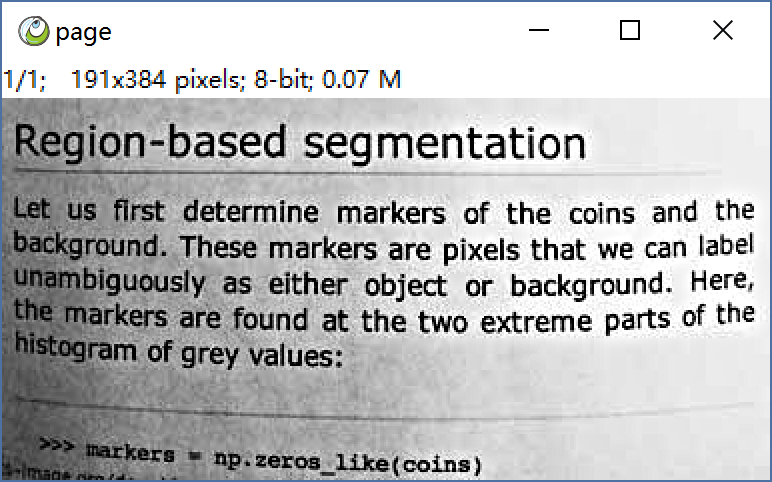
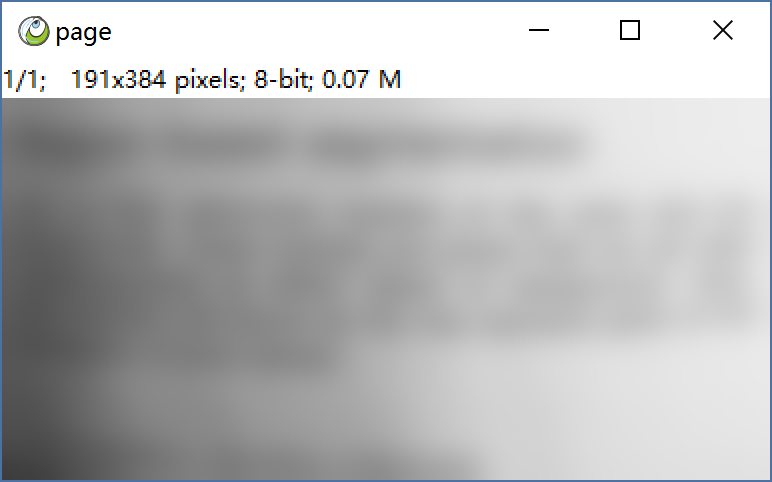
差分高斯滤波 阈值分割

**观察描述：**经过差分高斯滤波之后，平坦区域的值都接近于零，文字周围变量，文字部分变暗，从而实现了背景均一化，右侧为分割结果，较为理想。

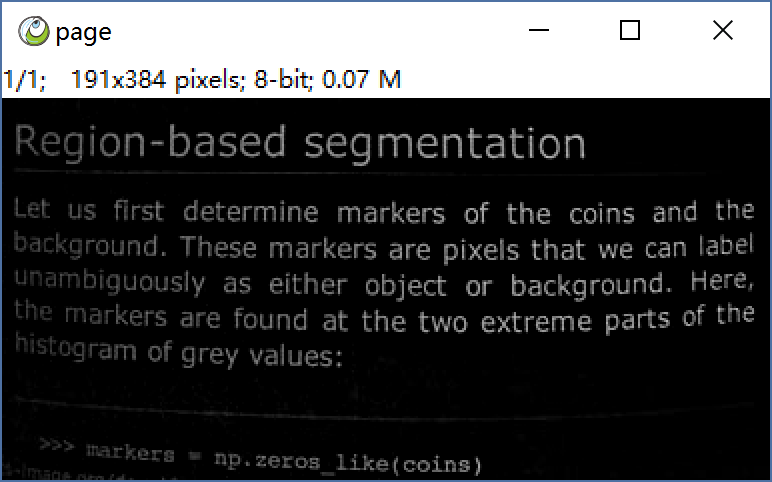
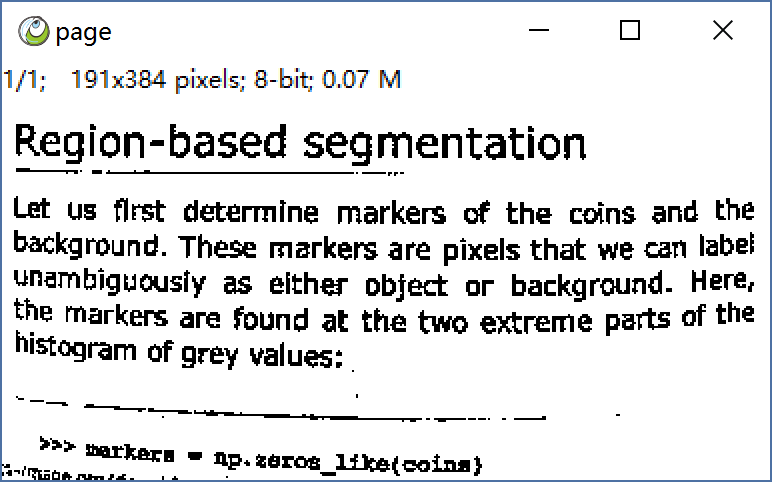
**另一个视角看待**

我们知道差分高斯滤波的本质时两个不同sigma的差，我们把这个过程分解来看，同时也更有助于理解本章内容。

**$ IBook > Chapter4 Digital-Filter > Use Filter To Extract Text Step**

原图 高斯滤波 sigma=8

原图减去高斯滤波 差异做阈值处理

**观察分析：**原图背景光照不均，进行高斯滤波后，细节被平滑掉，文字被融合掉了，而背景基本上被保留，某种意义上高斯滤波保留了渐变的背景，滤掉了文字。而差异自然就是被滤掉的文字，在此基础上做阈值处理，就可以得到较为理想的分割结果。另外你也可以常识用USM锐化原图，然后再进行阈值提取，那样相当于综合考虑原图与差异进行的分割。

**扩展阅读：**正如之前介绍的，高斯滤波可以模拟雾气中的景象，也可以模拟景物远去的过程。高斯模糊其实是像素之间的相互融合，而随着sigma增大，这种融合越来越强烈，细节的信息会被一步步被环境吞噬掉。而我们差分其实是找到其所在的尺度和距离。当某种程度的融合能够吞噬掉文字，而对背景不会造成太多改变，那么这个就是我们需要的尺度。

**排序滤波器**

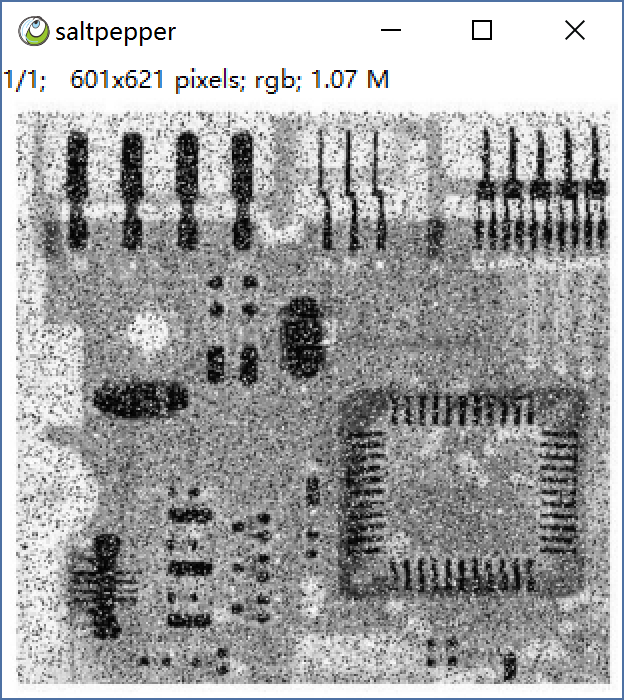
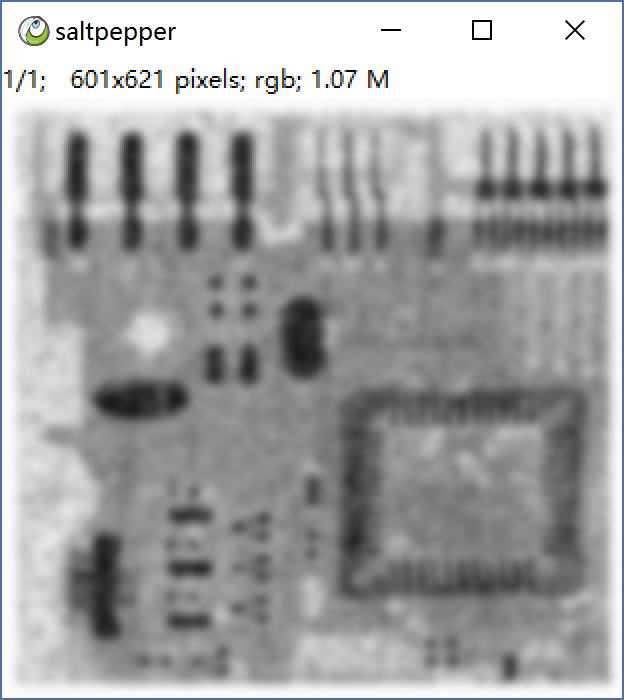
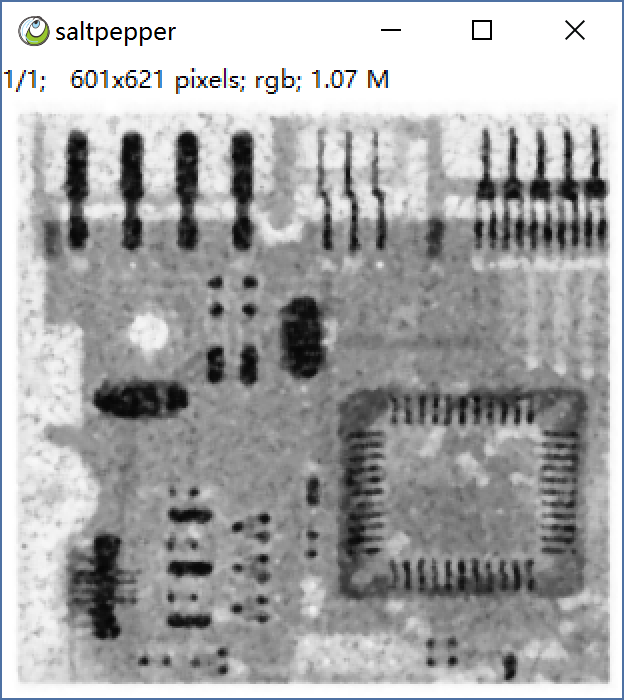
前面介绍的全部滤波器都属于线性滤波器，也就是说可以用一个卷积来表示，但滤波器不仅限于线性，也可以是非线性的，比如本章我们介绍的排序滤波器。

**中值滤波器**

**$ Process > Filter > Median**

在指定窗口内对像素值进行排序，选用排在中间的一个作为响应结果。

**$ IBook > Chapter4 Digital-Filter > Show Median Filter**

椒盐噪声的电路板 r=5 高斯滤波 r=5 中值滤波 r=5

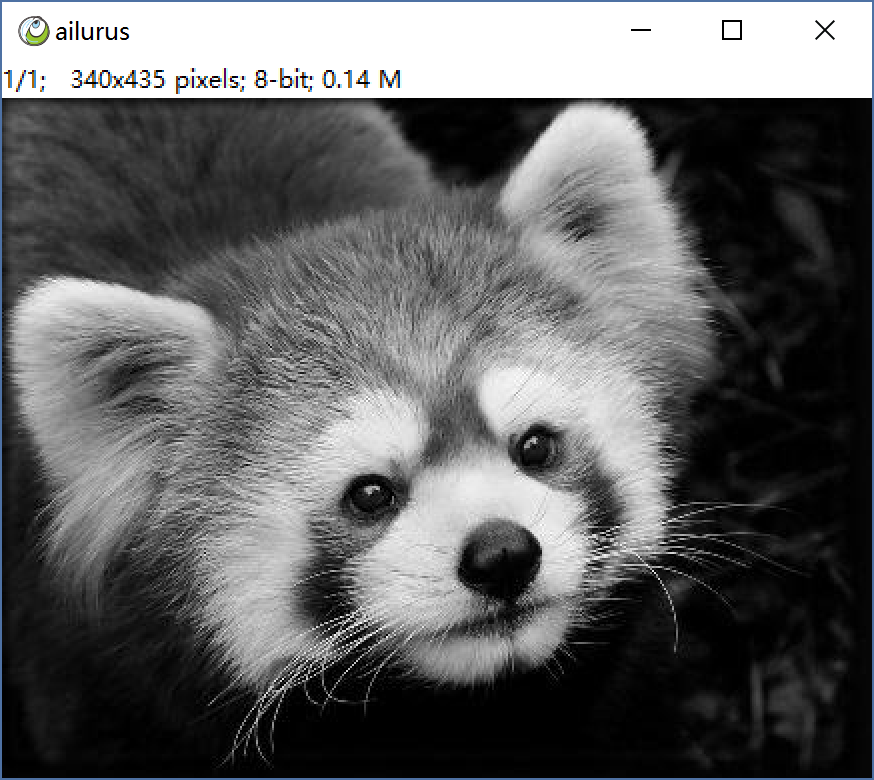
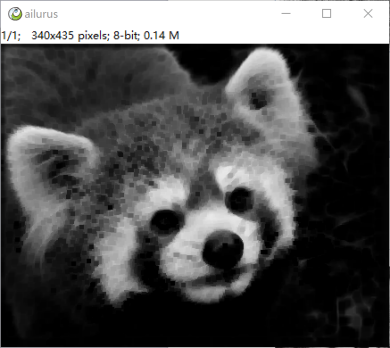
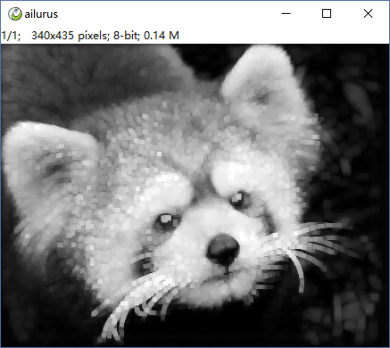
**观察描述：**椒盐噪声是一种特殊的噪声，由孤立的黑点，白点构成，看起来像在图片上撒下一把胡椒和盐，他们的特征是孤立的出现，并且与周围像素差异较大，由于差异过大，用小尺度sigma的高斯模糊难以抑制，而大尺度的sigma又会破坏图像其他部分的信息，尤其对于电路板，布线等信息非常重要，因此高斯滤波很难完美解决这类问题。而中值滤波对椒盐噪声有很好的抑制效果，因为孤立亮点与暗点在排序中会被排在前面或后面，不会被滤波器响应。

**最大值，最小值滤波器**

**$ Process > Filter > Maximum $ Process > Filter > Minimum**

与中值滤波类似，我们可以对排序在最前或最后的值做出响应，就得到了最大值，最小值滤波器。

**$ IBook > Chapter4 Digital-Filter > Show Max And Min Filter**

原图 r=5 最小值滤波 r=5 最大值滤波 r=5

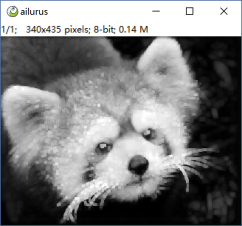
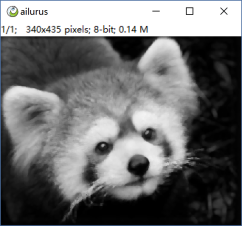
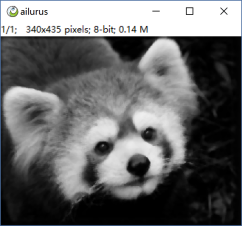
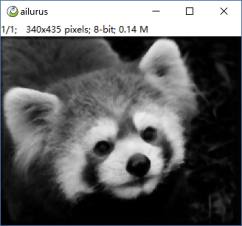
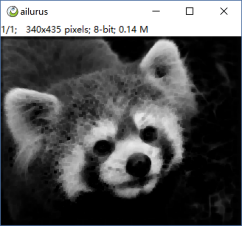
**观察描述：**原图是黑白分布比较密集的，尤其是毛发，我们用最小值滤波器滤波之后，看到胡须消失了，眼球内的高光小时了，眼珠扩张变大，整个皮毛都变深了。而最大值滤波器恰好相反。胡须变粗，黑眼球变小，内部的高光变大，整个毛发变量，如果对于椒盐噪声来说，最小值滤波器是抑制盐，增大胡椒，而最大值滤波器是抑制胡椒，增大盐。

**通用排序滤波器**

**$ Process > Filter > Percent**

前面我们学习了中值滤波，和最大值，最小值滤波，这里我们给出更通用的形式。

**$ IBook > Chapter4 Digital-Filter > Show Percent Filter**



per=0% per=25% per=50% per=75% per=100%

这里我们用一个邻域为5的窗口，对内部像素进行排序，然后选取排序在特定位置像素值作为响应。上图分别是同一张图对于不同比例的排序滤波器的滤波结果。可以看到是一个渐变过程，0%的时候相当于最小值滤波器，100%时相当于最大值滤波器，而50%相当于中值滤波器。

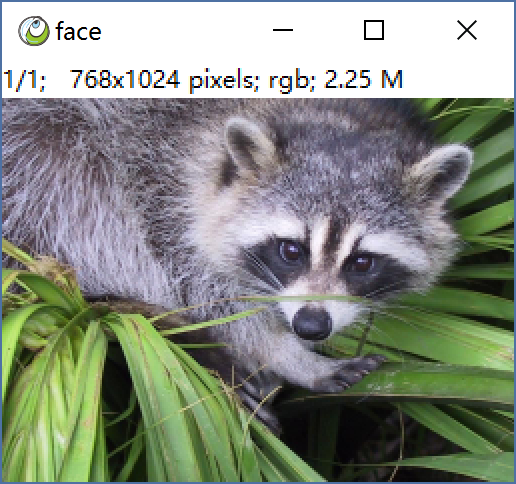
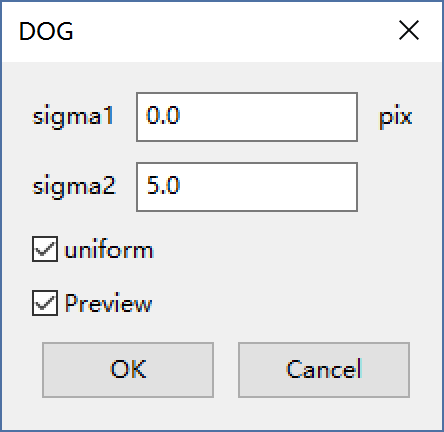
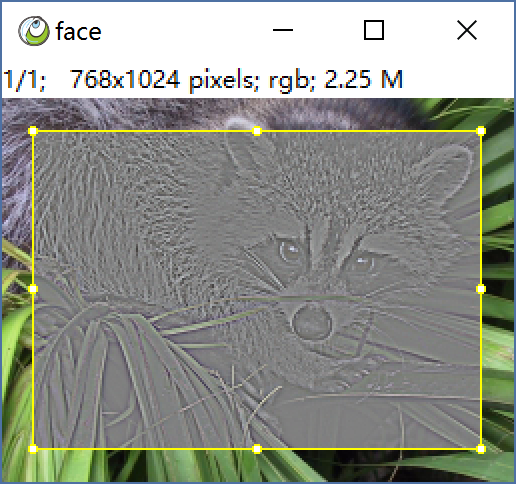
**对于线性，非线性的理解**

我们知道卷积是一系列的相乘与相加，如果把确定的模板当作常数，图像当作变量，那么卷积可以表达成若干像素的数乘与相加，因而叫做线性滤波器。而排序滤波器，必须对像素进行排序，然后取得相应位置的值，不能用卷积形式表达，因而属于非线性滤波器。

**ImagePy 中的滤波器运算**

**ImagePy中滤波器主要在Process > Filter菜单下，下面以差分高斯滤波为例。**

**$ File > Samples Local > face $ Process > Filter > DOG**

原图 DOG 参数对话框 作用于选区

多数的滤波器需要若干参数，ImagePy会弹出相应的对话框，勾上Preview选项可以实时看到处理效果。多数滤波器支持彩色图像，本质是依次处理每个通道。多数滤波器支持选区操作，既只作用于选区内部的像素。

**本章小结**

本章介绍了滤波器的基础概念和卷积运算，并且介绍了很多常用的低通，高通，锐化滤波器，以及他们之间的关系，并结合一个文字提取的例子，展示了滤波器的用法，最后介绍了非线性排序滤波器。滤波器的作用简单说就是增强某些信息，直接作用是图像增强，而这些增强可以是为了视觉效果，而更多的图像处理工作中，滤波器是必不可少的预处理过程，为更进一步提取其他信息做准备。滤波器是图像处理中非常重要的概念，希望读者通过本章学习，掌握卷积原理，并对高通，低通，锐化形成一定的认识。我们后续章节以及实验部分都会继续不断加强对各类滤波器的认识，掌握更多应用技巧。