图像滤波算法的研究

概述

由于成像系统、传输介质和记录设备等的不完善，数字图像在其形成、传输记录过程中往往会受到多种噪声的污染。另外，在图像处理的某些环节当输入的像对象并不如预想时也会在结果图像中引入噪声。这些噪声在图像上常表现为一引起较强视觉效果的孤立像素点或像素块。一般，噪声信号与要研究的对象不相关它以无用的信息形式出现，扰乱图像的可观测信息。对于数字图像信号，噪声表为或大或小的极值，这些极值通过加减作用于图像像素的真实灰度值上，对图像造成亮、暗点干扰，极大降低了图像质量，影响图像复原、分割、特征提取、图像识别等后继工作的进行。要构造一种有效抑制噪声的滤波器必须考虑两个基本问题：能有效地去除目标和背景中的噪声;同时，能很好地保护图像目标的形状、大小及特定的几何和拓扑结构特征。

而针对图像处理而言一般有对二维图像处理和三维点云图像处理两大类，本文将针对这两大类算法展开具体的研究。

一 传统图像降噪方法

1.0 图像降噪算法分类

虽然各种图像降噪算法犹如雨后春笋般不断新增，然而很多方法都存在一个通用的缺点，就是在降噪的同时往往会丢失图像的细节或边缘信息。一般的图像处理，微小的细节对图像降噪的后续处理程序影响不太明显，但是当处理对象为医学图像时，这样的小失误是不被允许的，因为在医疗诊断或治疗中，每一个微小的失误都会影响医师的治疗方法甚至威胁到患者的生命。这就要求更多的研究者来投入时间和精力研究新的降噪技术，以达到降噪并同时仍能保留足够细节信息的目的。

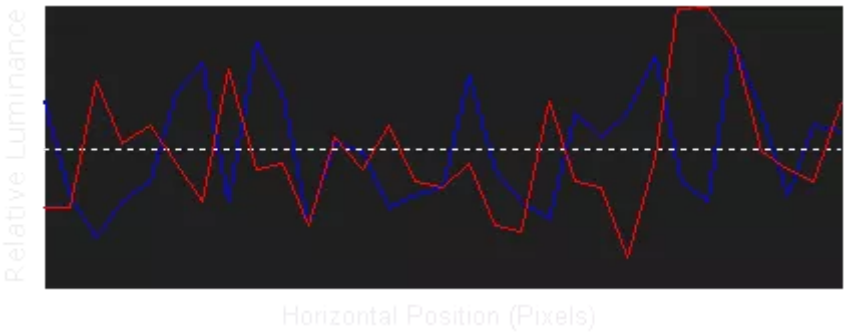
目前常用的图像去噪算法大体上可非为两类，即**空域像素特征去噪算法和变换域去噪算法。**前者是直接地在图像空间中进行的处理，后者是间接地在图像变换域中进行处理。

1.1 空域像素特征去噪算法

首先说明一点就是在信号处理教科书中，虽然介绍过很多经典的图像去噪方法，但主要都是针对随机噪声的，对于sensor缺陷导致的一些脉冲噪声（impulse noise）这里我们不考虑。

那么什么是随机噪声呢？相比于图像的真实信号来说随机噪声就是一种或高或低呈现出不确定变化的一种信号，如下图所示虚线代表真实信号，红蓝线表示的就是随机噪声信号，所有的随机噪声信号求和后结果为0。

由于这个零和特点，目前几乎所有的空域降噪算法都是基于这个理论为出发点来进行降噪处理的。



基于空域像素特征的方法，是通过**分析在一定大小的窗口内，中心像素与其他相邻像素之间在灰度空间的直接联系**，来获取新的中心像素值的方法，因此往往都会存在一个典型的输入参数，即滤波半径r。此滤波半径可能被用于在该局部窗口内计算像素的相似性，也可能是一些高斯或拉普拉斯算子的计算窗口。在邻域滤波方法里面，最具有代表性的滤波方法有以下几种：

(1) 算术均值滤波与高斯滤波

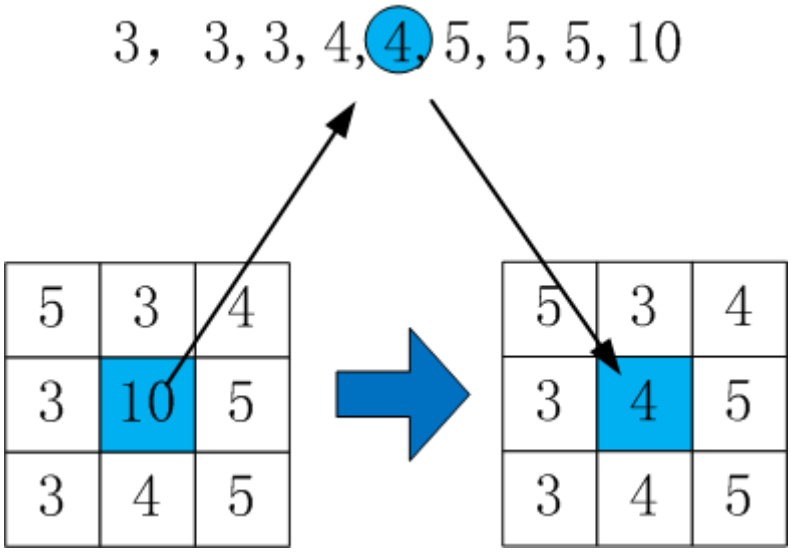
算术均值滤波用**像素邻域的平均灰度**来代替像素值，适用于脉冲噪声，因为脉冲噪声的灰度级一般与周围像素的灰度级不相关，而且亮度高出其他像素许多。

$$A'(i,j)=\frac{1}{(2L+1)^2}\sum_{k=-L}^{L}\sum_{l=-L}^{L}A(i+k,j+l)$$

- 均值滤波结果A'(i,j)随着L（滤波半径）取值的增大而变得越来越模糊，图像对比度越来越小。经过均值处理之后，噪声部分被弱化到周围像素点上，所得到的结果是噪声幅度减小，但是噪声点的颗粒面积同时变大，所以污染面积反而增大。为了解决这个问题，可以通过设定阈值，比较噪声和邻域像素灰度，只有当差值超过一定阈值时，才被认为是噪声。不过阈值的设置需要考虑图像的总体特性和噪声特性，进行统计分析。自适应均值滤波算法通过方向差分来寻找噪声像素，从而赋予噪声像素与非噪声像素不同的权重，并自适应地寻找最优窗口大小，优于一般的均值滤波方法。
- 高斯滤波矩阵的权值，随着与中心像素点的距离增加，而呈现高斯衰减的变换特性。这样的好处在于，离算子中心很远的像素点的作用很小，从而能在一定程度上保持图像的边缘特征。通过调节高斯平滑参数，可以在图像特征过分模糊和欠平滑之间取得折中。与均值滤波一样，高斯平滑滤波的尺度因子越大，结果越平滑，但由于其权重考虑了与中心像素的距离，因此是更优的对邻域像素进行加权的滤波算法。

(2) 统计中值滤波

中值滤波首先确定一个滤波窗口及位置（通常含有奇数个像素），然后将窗口内的像素值按**灰度大小进行排序**，最后取其中位数代替原窗口中心的像素值（如下图）。



但当噪声像素个数大于窗口像素总数的一半时，由于灰度排序的中间值仍为噪声像素灰度值，因为滤波效果很差。此时如果增加窗口尺寸，会使得原边缘像素被其他区域像素代替的几率增加，图像更容易变模糊，并且运算量也大大增加。

无论是中值滤波还是加权滤波，两者受窗口的尺寸大小影响非常大。一种对中值滤波的改进是自适应中值滤波，它首先判断窗口内部的中心像素是否是一个脉冲，如果不是，则输出标准中值滤波的结果；如果是，则通过继续增大窗口滤波尺寸来寻找非脉冲的中值，因此该方法相比较原始的统计中值滤波器，在保持清晰度和细节方面更优。

(3) 双边滤波

这是一种非线性的保边滤波方法，是**结合图像的空间邻近度和像素值相似度**的一种折中处理，同时考虑空域信息和灰度相似性，达到保边去噪的目的。具有简单、非迭代、局部的特点。双边滤波器之所以可以达到保边去噪的效果，是因为滤波器是由两个函数构成。一个函数是由几何空间距离决定滤波器系数。另一个由像素差值决定滤波器系数。双边滤波器中，输出像素的值g(i,j)依赖于邻域像素的值的加权组合：

$$g(i,j)=\frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$\omega(i,j,k,l)=\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2}-\frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right)$$

空域　　值域

上图中权重系数w(i,j)取决于空域核和值域核的乘积。其中空域滤波器对空间上邻近的点进行加权平均，加权系数随着距离的增加而减少。值域滤波器则是对像素值相近的点进行加权平均，加权系数随着值差的增大而减少。

(4) 引导滤波（guided filter）

高斯滤波等线性滤波算法所用的核函数相对于待处理的图像是独立无关的，这里的独立无关也就意味着，对任意图像都是采用相同的操作。

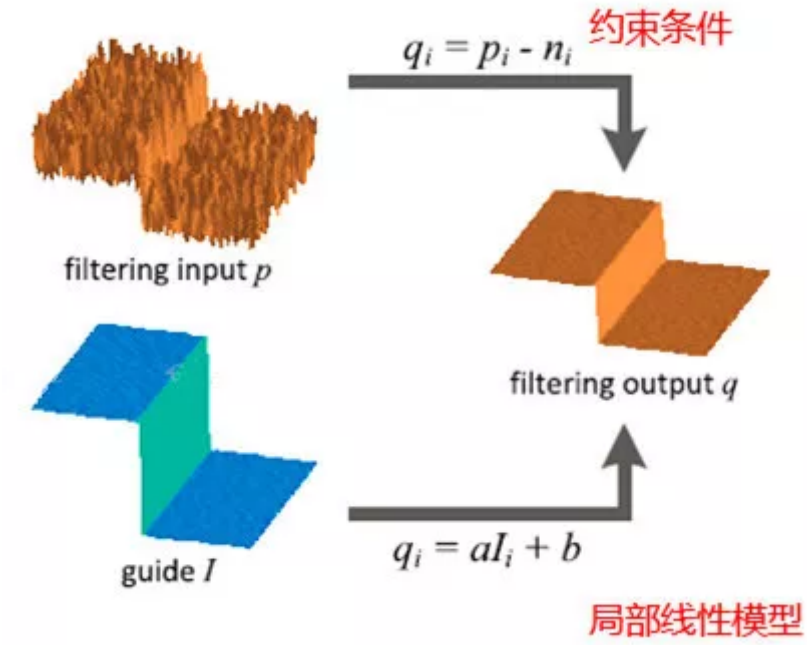
引导滤波就是在**滤波过程中加入引导图像中的信息**，这里的引导图可以是单独的图像也可以是输入图像，当引导图为输入图像时，引导滤波就成为了一个可以保持边缘的去噪滤波操作。我们来看一下具体算法原理：

第一步：假设该引导滤波函数的输出与输入在一个二维窗口内满足线性关系，如下：

$$q_i = a_k I_i + b_k, \forall i \in \omega_k,$$

$$q_i = p_i - n_i$$

其中，q是输出像素的值，即p去除噪声或者纹理之后的图像，ni表示噪声，I是输入图像的值，i和k是像素索引，a和b是当窗口中心位于k时该线性函数的系数。（当引导图为输入图像时，引导滤波就成为一个保持边缘的滤波操作，即 I= p，对上示两边取梯度可得q'=aI'，即当输入图I有梯度时，输出q也有类似的梯度，这也就可以解释为什么引导滤波有边缘保持特性了。



第二步求出线性函数的系数，也就是线性回归，即希望拟合函数的输出值q与真实值p之间的差距最小，转化为下面但最优化问题，也就是让下式最小：

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left(\left(a_k I_i + b_k - p_i \right)^2 + \epsilon a_k^2 \right).$$

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon}$$

$$b_k = \bar{p}_k - a_k \mu_k.$$

在这里，μk和σk^2表示I在局部窗口wk中的均值和方差。 |ω|是窗口内的所有像素数，pk表示p在窗口wk中的均值，ϵ就是规整化参数，当I=p时，上面第二个公式即可简化为：

$$a_k = \frac{\sigma_k^2}{\sigma_k^2 + \epsilon}$$

$$\mathrm{b_k} = (1 - a_k)\mu_\mathrm{k}$$

- 如果ϵ=0，显然a=1, b=0是E(a,b)为最小值的解，从上式可以看出，这时的滤波器没有任何作用，将输入原封不动的输出。
- 如果ϵ>0，在像素强度变化小的区域（方差不大），即图像I在窗口wk中基本保持固定，此时有σ2k<<ϵ，于是有ak≈0和bk≈μk，即做了一个加权均值滤波，而在高方差区域，即表示图像I在窗口wk中变化比较大，此时我们有σ2k>>ϵ，于是有ak≈1和bk≈0，对图像的滤波效果很弱，有助于保持边缘。
- 在窗口大小不变的情况下，随着ϵ的增大，滤波效果越明显。

第三步：在计算每个窗口的线性系数时，我们可以发现一个像素会被多个窗口包含，也就是说，每个像素都由多个线性函数所描述。因此，如之前所说，要具体求某一点的输出值qi时，只需将所有包含该点的线性函数值平均即可，如下：

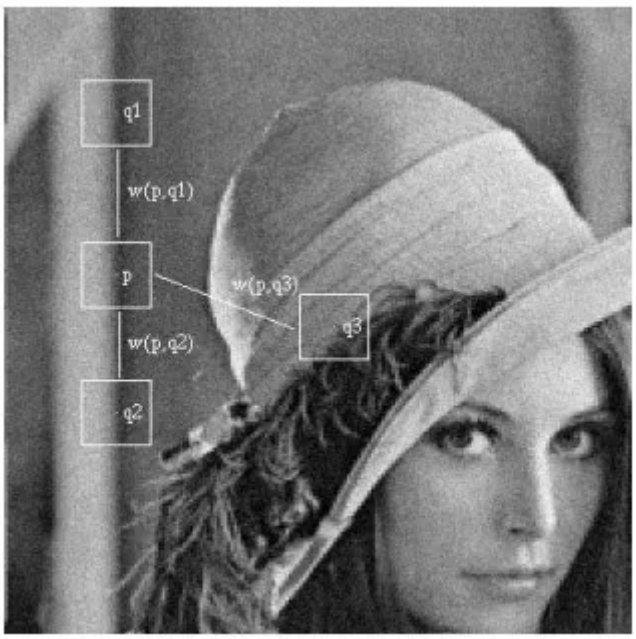
$$q_i = \frac{1}{|\omega|} \sum_{k:i \in \omega_k} (a_k I_i + b_k)$$
$$= \bar{a}_i I_i + \bar{b}_i$$

其中，输出值q又与两个均值有关，分别为a和b在窗口w中的均值，我们将上一步得到两个图像ak和bk都进行盒式滤波，得到两个新图：ai'和bi'。然后用ai'乘以引导图像Ii，再加上bi'，即得最终滤波之后的输出图像q。

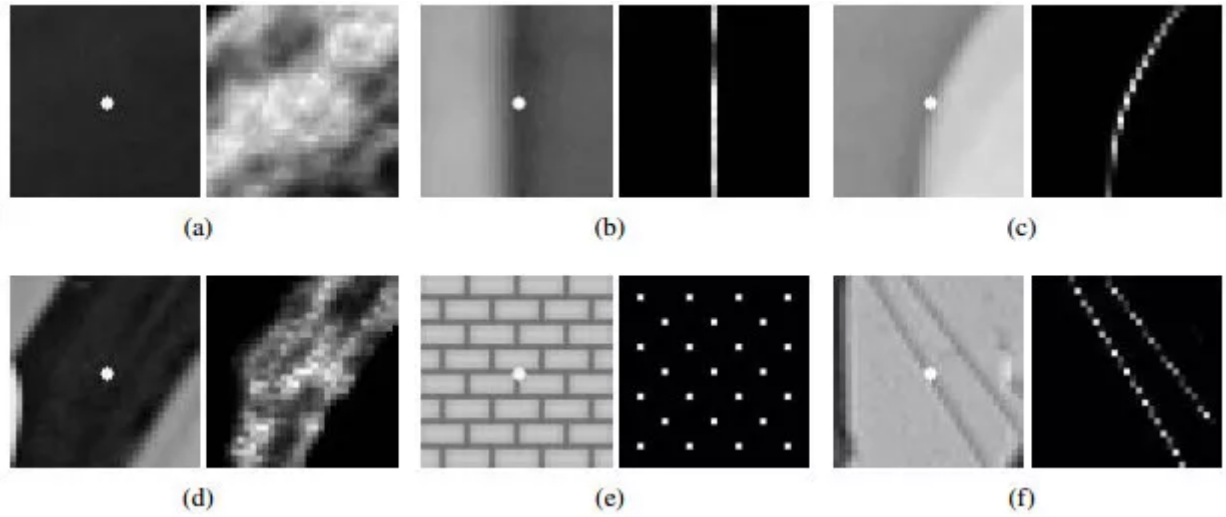
(5) NLM（Non-Local means）算法

前面基于邻域像素的滤波方法，基本上只考虑了有限窗口范围内的像素灰度值信息，没有考虑该窗口范围内像素的统计信息如方差，也没有考虑整个图像的像素分布特性，和噪声的先验知识。

针对其局限性，NLM算法被提出，该算法使用自然图像中普遍存在的冗余信息来去噪声。与常用的双线性滤波、中值滤波等利用图像局部信息来滤波不同的是，它**利用了整幅图像来进行去噪**，以图像块为单位在图像中寻找相似区域，再对这些区域求平均，能够比较好地去掉图像中存在的高斯噪声。这里我直接拿图来说可能会更能说明问题：



如上图所示，其中p为去噪的点，从图中看出q1和q2的邻域与p相似，所以权重和比较大，而q3因为与q邻域相差比较大所以赋予的权重值就很小。NLM就是将一幅图像中所有点的权重都表示出来，那就得到下面这些权重图：



上面权值图像中，左边是原图，中心的白色色块代表了像素 块邻域，右边是计算出来的权重图，权重范围从0（黑色）到1（白色）。

这个块邻域在整幅图像中移动，计算图像中其他区域跟这个块的相似度，相似度越高，得到的权重越大。最后将这些相似的像素值根据归一化之后的权重加权求和，得到的就是去噪之后的图像了。

由于原始NLM方法需要用图像中所有的像素来估计每一个像素的值，因此计算量非常大，研究者不断对该方法进行了几点改进。(a) 采用一定的搜索窗口代替所有的像素，使用相似度阈值，对于相似度低于某一阈值的像素，不加入到权重的计算(即不考虑其相对影响，这些都可以降低计算复杂度。(b)使用块之间的显著特征，如纹理特征等代替灰度值的欧氏距离来计算相似度，在计算上更加有优势，应用上也更加灵活。

除了上面所说的方法，还有如加权最小二乘法(WLS)，变分法(TV)等滤波算法，并且上面的这些算法都产生出了非常多的变种，篇幅有限不再一一详述，可以参考文献【1】。

1.2 变换域去噪算法

空域去噪都是从空间的角度去思考如何去噪，也就是所谓的spatial noise reduction，这条路子能想的方法也都做得差不多了，于是有人就换个角度想问题，就有了变换域做去噪的方法。通过数学变换，在变换域上把信号和噪声分离，然后把噪声过滤掉，剩下的就是信号。如下图没有噪声的信号就比较顺滑没有杂质。

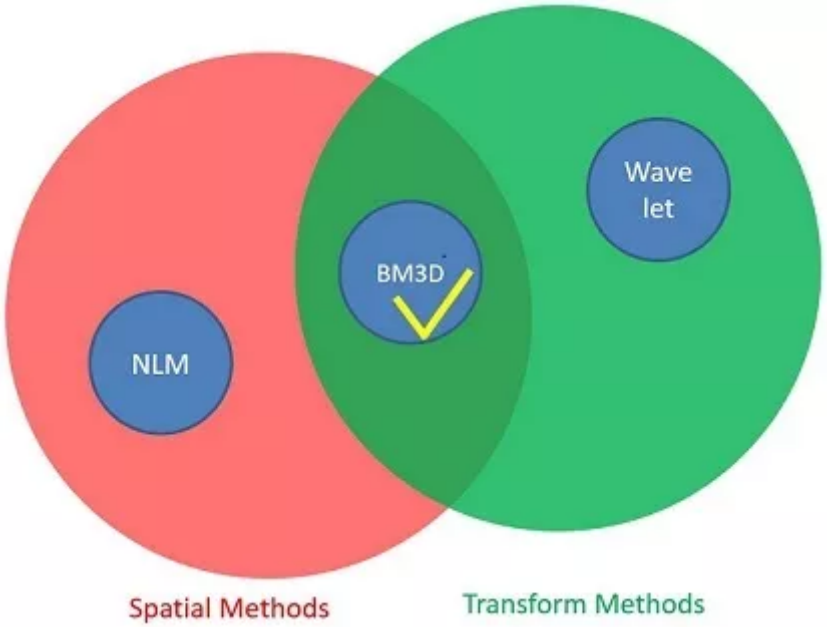
而下图中含有噪声的信号就会显得参差不齐，毛刺较多。而如果我们可以将噪声变换一个域后设定一个阈值将高于阈值的部分去掉，再反变换后剩下的就是干净的信号了。

因此图像变换域去噪算法的基本思想其实就是**首先进行某种变换，将图像从空间域转换到变换域**，然后从频率上把噪声分为高中低频噪声，用这种变换域的方法就可以把不同频率的噪声分离，之后进行反变换将图像从变换域转换到原始空间域，最终达到去除图像噪声的目的。

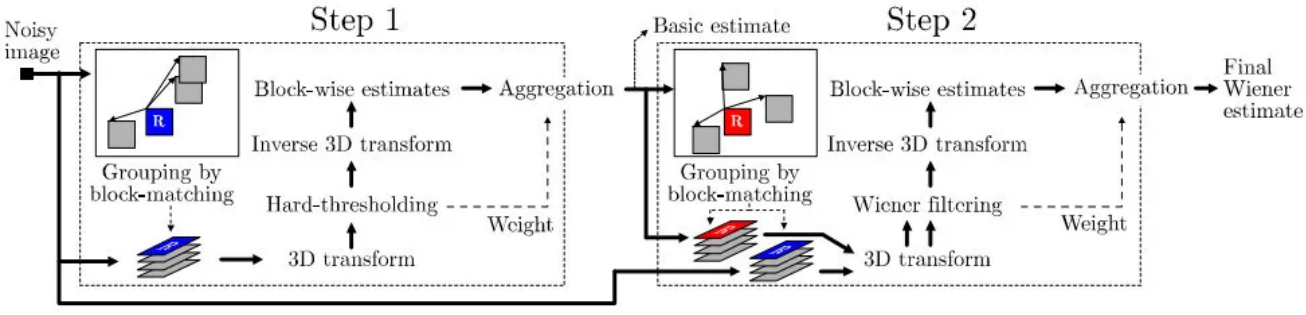
图像从空间域转换到变换域的方法很多，其中最具代表性的有傅里叶变换、离散余弦变换、小波变换以及多尺度几何分析方法等。

其中基于小波萎缩法是目前研究最为广泛的方法，小波萎缩法又分成如下两类：第1类是阈值萎缩，由于阈值萎缩主要基于如下事实，即比较大的小波系数一般都是以实际信号为主，而比较小的系数则很大程度是噪声。因此可通过设定合适的阈值，首先将小于阈值的系数置零，而保留大于阈值的小波系数；然后经过阈值函数映射得到估计系数；最后对估计系数进行逆变换，就可以实现去噪和重建；而另外一种萎缩方法则不同，它是通过判断系数被噪声污染的程度，并为这种程度引入各种度量方法(例如概率和隶属度等)，进而确定萎缩的比例，所以这种萎缩方法又被称为比例萎缩。

1.3 BM3D去噪算法



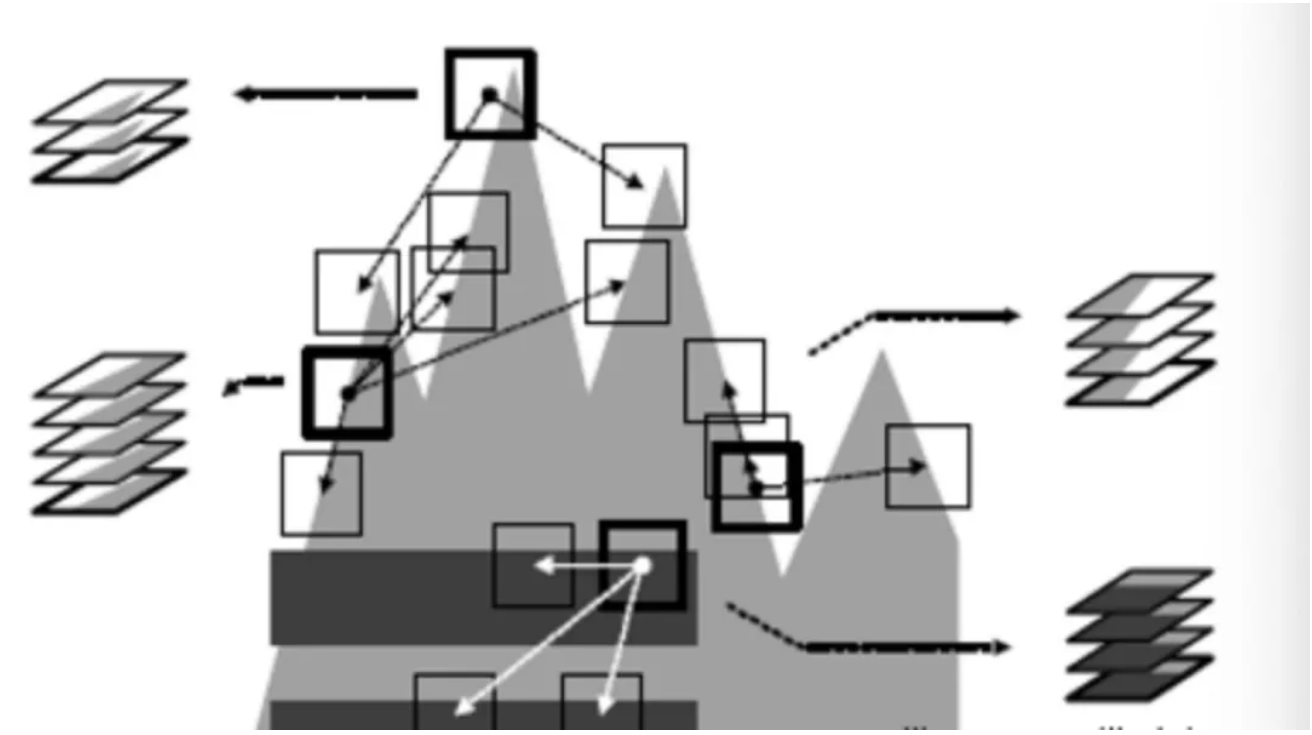
空域中NLM算法和变换域中小波萎缩法效果都很好，一个很自然的想法就是是否可以将两者相结合呢？是的，BM3D就是融合了spatial denoise和tranform denoise，从而可以得到最高的峰值信噪比。它先吸取了NLM中的计算相似块的方法，然后又融合了小波变换域去噪的方法。我们来看一下具体算法流程如下图：



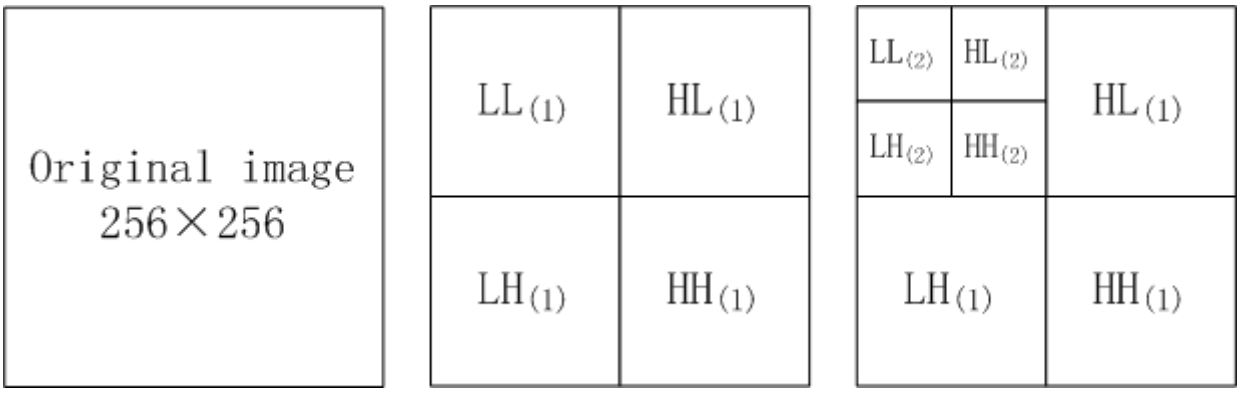
BM3D算法总共有两大步骤，分为基础估计（Step1）和最终估计（Step2）。在这两大步中，分别又有三小步：相似块分组，协同滤波和聚合。

Stpe1：基础估计

(1) 相似块分组：首先在噪声图像中选择一些大小的参照块（考虑到算法复杂度，不用每个像素点都选参照块，通常隔3个像素为一个步长进行选取，复杂度降到1/9），在参照块的周围适当大小区域内进行搜索，寻找若干个差异度最小的块，并把这些块整合成一个3维的矩阵。



(2) 协同滤波：形成若干个三维的矩阵之后，首先将每个三维矩阵中的二维的块（即噪声图中的某个块）进行二维变换，可采用小波变换或DCT变换等。二维变换结束后，在矩阵的第三个维度进行一维变换，变换完成后对三维矩阵进行硬阈值处理，将小于阈值的系数置0，然后通过在第三维的一维反变换和二维反变换得到处理后的图像块。



(3) 聚合：此时，每个二维块都是对去噪图像的估计。这一步分别将这些块融合到原来的位置，每个像素的灰度值通过每个对应位置的块的值加权平均，权重取决于置0的个数和噪声强度。

Step2：最终估计

具体的步骤从流程图可看出和Step1基本一样，不同的有两处：

一处是聚合过程将会得到两个三维数组：噪声图形成的三维矩阵和基础估计结果的三维矩阵。

另一处是协同滤波中用维纳滤波（Wiener Filtering）代替了硬阈值处理。

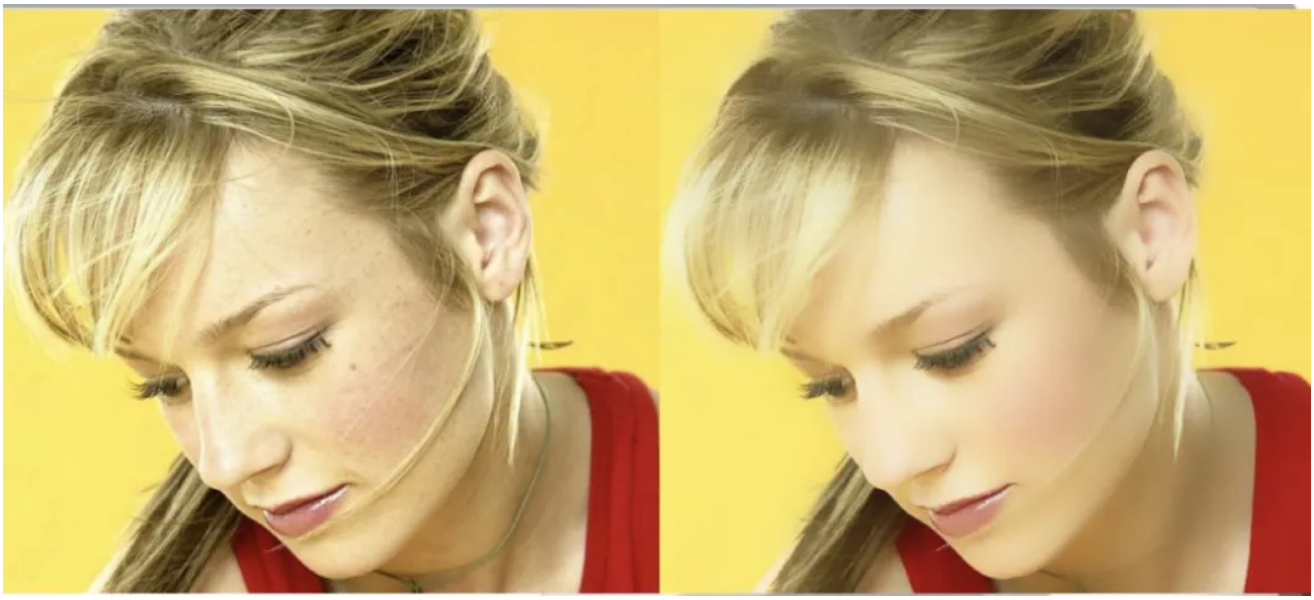
滤波器抑制噪声比较

对图像进行滤波去噪的算法其实就是一个加权平均的运算过程，滤波后图像中的每个像素点都是由其原图像中该点邻域内多个像素点值得加权平均，不同的滤波器最根本的差异就是权值不同。另外根据噪声的不同，滤波效果也各有不同。

- 均值滤波处理会噪声部分被弱化到周围像素点上，所得到的结果是噪声幅度减小，但是噪声点的颗粒面积同时变大，所以污染面积反而增大。
- 高斯滤波器用像素邻域的加权均值来代替该点的像素值，而每一邻域像素点权值是随该点与中心点的距离单调增减的．这一性质是很重要的，因为边缘是一种图像局部特征，如果平滑运算对离算子中心很远的像素点仍然有很大作用，则平滑运算会使图像失真，因此缺点是权重完全取决于图像像素之间欧氏距离，与图像的内容没有关系。

- 中值滤波适用于椒盐噪声和脉冲噪声。因为对于受脉冲噪声和椒盐噪声污染的图像，相应位置的图像灰度发生了跳变，是不连续的，而此处的中值滤波正是一种非线性滤波方法，对这些类型的随机噪声，它比相同尺寸的线性平滑滤波器引起的模糊更少，能较好的保持边缘，但会使图像中的小目标丢失，因此对点、线和尖顶多的图像不宜采用中值滤波。

- 双边滤波器的好处是可以做边缘保存，一般过去用的维纳滤波或者高斯滤波去降噪，都会较明显地模糊边缘，对于高频细节的保护效果并不明显。双边滤波顾名思义比高斯滤波多了一个高斯方差，它是基于空间分布的高斯滤波函数，所以在边缘附近，离的较远的像素不会太多的影响到边缘上的像素值，这样就保证了边缘附近像素值的保存。但是由于保存了过多的高频信息，对于彩色图像里的高频噪声，双边滤波器不能够干净的滤掉，只能够对于低频信息进行较好的滤波因此，双边滤波器即平滑滤波了图像，又保持的图像边缘。虽然去噪效果很明显，但很多细节被去除，只有整体形状被保留，不过美颜相机磨皮恰恰就需要这种算法（如下图美女磨皮后的效果）。



- 引导滤波不像高斯滤波等线性滤波算法所用的核函数相对于待处理的图像是独立无关的，而是在滤波过程中加入了引导图像中（去噪时用的就是图像本身）的信息，所以引导滤波本质上就是通过一张引导图I，对初始图像p（输入图像）进行滤波处理，使得最后的输出图像大体上与初始图像P相似，但是纹理部分与引导图I相似。在滤波效果上，引导滤波和双边滤波差不多，在一些细节上，引导滤波较好。引导滤波最大的优势在于能够保持线性复杂度，每个像素虽然由多个窗口包含，求某一点像素值的具体输出值时，只需将包含该点所有的线性函数值平均即可，而双边滤波不是线性复杂度在于他考虑了每个点的几何差距与强度差距两个因素，当处理图像较大时，运算量很明显会增大很多。

- 非局部算法获得的信噪比比双边滤波略高，有时候还不如双边滤波。但是，非局部滤波是一种基于快的匹配度来计算滤波权值的，所以能获得比较好的视觉效果。然而，它的计算复杂度实在是太高了。最原始非局部均值算法是在整个图片中进行块搜索，根据块的匹配度来计算权值。实际执行过程，都会把搜索区域限定在一个局部的搜索窗口中。

- **BM3D算法是目前传统算法中效果最好的去噪算法**，相比于NLM噪声更少，图像细节恢复更多，但算法复杂度实在太高，除非解决计算性能问题，不然至少工业界是无法容忍几分钟的处理时间进行降噪处理。

1.4 总结

图像去噪难点在于区别高频信号（如纹理、边缘）和噪声，去噪常用思想是利用图像的相似性。空域去噪是认为相近的点相似，通过平滑可以降低随机性的噪声，效果较好的去噪方法大多是多种方法结合，既能很好地保持边缘信息，又能去除图像中的噪声，比如将中值滤波和小波滤波结合起来进行滤波。

基本上传统的去噪算法都是从噪音图像中找出规律后再进行相对应的去噪处理。那么如果从有噪音的图片本身无法找到规律，我们是否也可以借助其他类似但又没有噪音的图片，来总结图片具有的固有属性呢？

深度学习方法是数据驱动的方法，在仿真图像去噪上，数据（也就是干净图像）是非常充足的，所以当前深度学习方法在高斯白噪声假设条件下的滤波问题中已经达到甚至超过BM3D算法。

二 三维点云降噪方法

2.1 传统点云降噪算法

点云滤波是点云处理的基本步骤，也是进行 high level 三维图像处理之前必须要进行的预处理。其作用类似于信号处理中的滤波，但实现手段却和信号处理不一样，我认为原因有以下几个方面：

1. 点云不是函数，对于复杂三维外形其x,y,z之间并非以某种规律或某种数值关系定义。所以点云无法建立横纵坐标之间的联系。
2. 点云在空间中是离散的。和图像，信号不一样，并不定义在某个区域上，无法以某种模板的形式对其进行滤波。换言之，点云没有图像与信号那么明显的定义域。
3. 点云在空间中分布很广泛。历整个点云中的每个点，并建立点与点之间相互位置关系成了最大难点。不像图像与信号，可以有迹可循。
4. 点云滤波依赖于几何信息，而不是数值关系。

综上所述，点云滤波只在抽象意义上与信号，图像滤波类似。因为滤波的功能都是突出需要的信息。

PCL常规滤波手段均进行了很好的封装。对点云的滤波通过调用各个滤波器对象来完成。主要的滤波器有*直通滤波器，体素格滤波器，统计滤波器，半径滤波器* 等。不同特性的滤波器构成了较为完整的点云前处理族，并组合使用完成任务。实际上，滤波手段的选择和采集方式是密不可分的。

1. 如果使用线结构光扫描的方式采集点云，必然物体沿z向分布较广，但x,y向的分布处于有限范围内。此时可使用直通滤波器，确定点云在x或y方向上的范围，可较快剪除离群点，达到第一步粗处理的目的。
2. 如果使用高分辨率相机等设备对点云进行采集，往往点云会较为密集。过多的点云数量会对后续分割工作带来困难。体素格滤波器可以达到向下采样同时不破坏点云本身几何结构的功能。点云几何结构不仅是宏观的几何外形，也包括其微观的排列方式，比如横向相似的尺寸，纵向相同的距离。随机下采样虽然效率比体素滤波器高，但会破坏点云微观结构。
3. 统计滤波器用于去除明显离群点（离群点往往由测量噪声引入）。其特征是在空间中分布稀疏，可以理解为：每个点都表达一定信息量，某个区域点越密集则可能信息量越大。噪声信息属于无用信息，信息量较小。所以离群点表达的信息可以忽略不计。考虑到离群点的特征，则可以定义某处点云小于某个密度，既点云无效。计算每个点到其最近的k个点平均距离。则点云中所有点的距离应构成高斯分布。给定均值与方差，可剔除3∑之外的点。
4. 半径滤波器与统计滤波器相比更加简单粗暴。以某点为中心画一个圆计算落在该圆中点的数量，当数量大于给定值时，则保留该点，数量小于给定值则剔除该点。此算法运行速度快，依序迭代留下的点一定是最密集的，但是圆的半径和圆内点的数目都需要人工指定。

实际上点云滤波的手段和传统的信号滤波与图像滤波在自动化程度，滤波效果上还有很大的差距。学者大多关注图像识别与配准算法在点云处理方面的移植，而对滤波算法关注较少。其实点云前处理对测量精度与识别速度都有很大影响。

2.2 形态学滤波

1. 基本形态学操作

首先来介绍一下机载点云滤波过程中用到的基本形态学操作：

- 侵蚀：将邻域内高程最低点作为计算点的高程点；
- 扩张：将领域内高程最高点作为计算点的高程点；
- 开运算：先进行侵蚀操作再进行扩张操作；
- 闭运算：先进行扩张操作再进行侵蚀操作；

开运算的特性，能够将邻域窗口内突出的物体剔除；计算开运算后高程值和初始高程值的差，将高程差小于阈值的点归类为地面点，否则被归类为地物点。

2.栅格化和空缺数据填充

将点云数据按照一定网格尺寸进行栅格化，当有多个点落入同一网格时，取其中最低点的高程作为网格的高程，将该栅格数据表示为“g_min”；

对于没有点的栅格需要进行填充处理，具体填充方法为：

2.1定位空缺数据的区域

a) 对栅格数据进行二值化，1代表存在数据的栅格 ，0代表不存在数据的栅格；

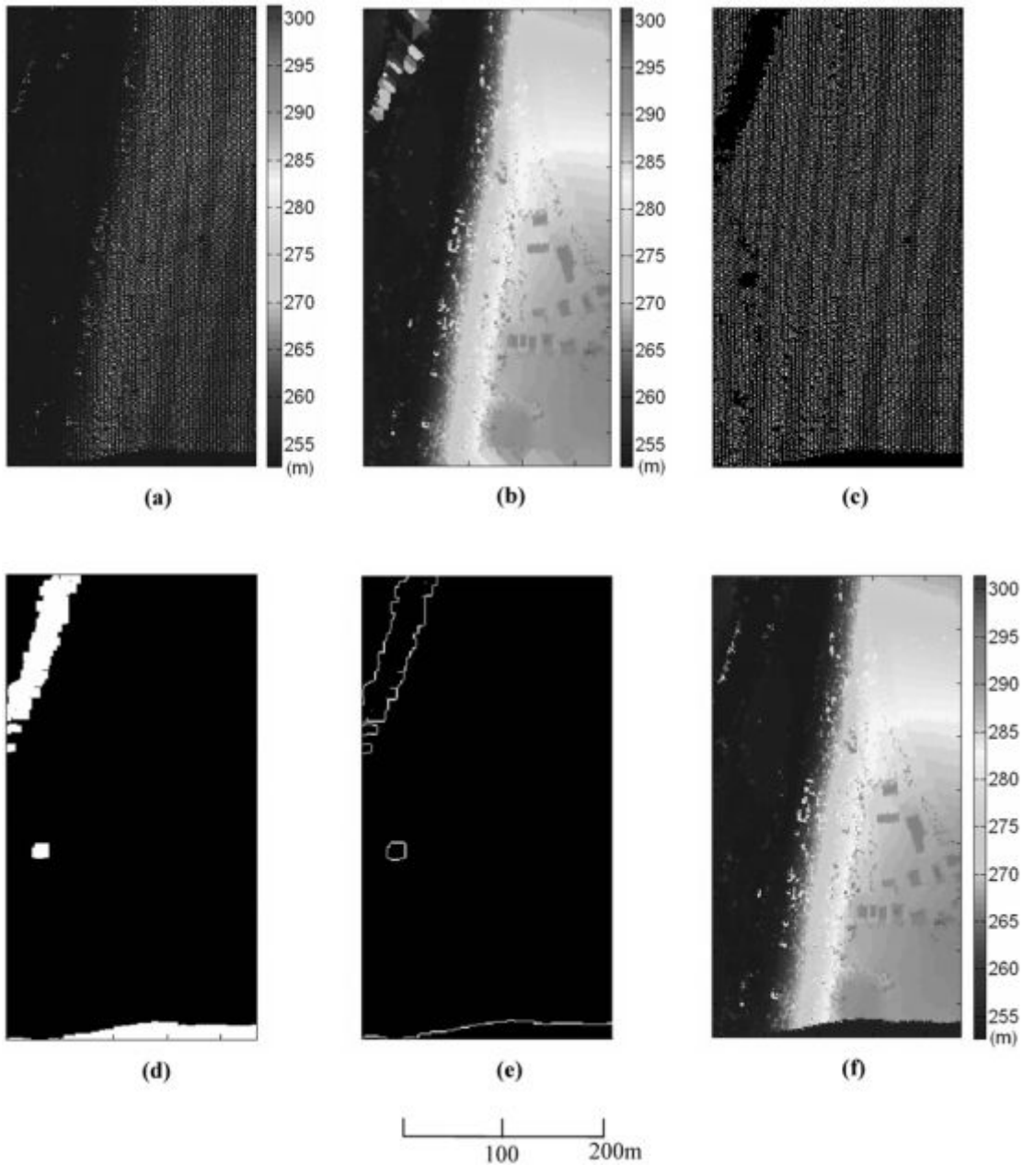
b) 对二进制图像进行闭运算操作，设置参数为r；

$$r = \left(\frac{1}{d}\right)^{0.5} \cdot \frac{1}{c}$$

式中：d表示点云密度（每平方米中点的个数），由原始数据计算得到；c表示网格尺寸，单位为米；在闭运算操作之后，稀疏数据之间的间隙消失了，并且数据空缺部分区域将显现出来。

2.2 对于每片区域，将网格中的“g_min”值替换为边界内最低网格的高程值。每片区域的边界是通过扩张后的数据减去原始数据得到的。设置扩张时的参数r为1；

2.3 将修改后的“g_min”进行替换操作，迭代的将网格中没有数据的网格的值替换为最近的有数据的网格的值。图1b和图1f展示了空缺数据填充后的效果。填充后的网格用“g_fmin”表示。



图对原始数据进行栅格化，并对缺失数据进行填充；b).缺失区域填充前，采用最近高程点填充网格；c).二进制栅格，用来表明栅格中是否存在点云数据；d).形态学操作处理后得到的空缺数据的区域；e).表示d)中每片区域中的边界；f).缺失区域填充后，采用最近高程点填充网格。

3.对树木区域进行过滤处理

对网格进行形态学开运算操作，邻域网格尺寸至少包含一个地形点数据；但是很难判别邻域网格尺寸中是否存在至少一个地形点数据；因此在实际操作过程中使用不同的邻域网格尺寸来对“g_fmin”操作，最终选择一个视觉上地形相对平滑，尺寸最小的邻域网格；其中，网格的尺寸用“d_min”来表示；开运算操作后的网格用“g_o(fmin)”表示；

在对树木进行过滤时，使用的邻域窗口尺寸为：“d_min”，数据中最大的建筑物尺寸为“d_max”，因此在进行建筑物滤波时，所使用的邻域窗体尺寸“wi”为：

$$w_i = \begin{Bmatrix} d_{\min} + 2^i & if\left(w_i < d_{\max}\right) \\ d_{\max} & if\left(w_i \le d_{\max}\right) \end{Bmatrix}$$

4.对建筑物区域进行过滤处理

本文提出了一种新方法来滤除建筑物点云，而保留地形点云数据。将滤除树木后的点云数据称为originalOpen，用一个较大的窗口尺寸进行开运算处理，得到newOpen；利用newOpen和originalOpen做差将得到滤除部分点云，判断滤除部分点云边缘点高程大小，如果较大则为建筑物点云，较小则为地形点云（示意图如图所示）。

图2 建筑物滤除示意图，其中箭头所指向的点为边缘点，建筑物的边缘点高差较大；地形的边缘点高差较小。

进行建筑物点云过滤时，具体的伪代码如下图所示：

1. *buildingMask* = 0 # binary image indicating it is a building or not. 1 for buildings and 0 for else
2. *originalOpen* = $g_{o(f\,min)}$
3. for $w_i = w_1$ to w_n
4. *newOpen* = imopen(*originalOpen*, w_i) # open original Open with window size w_i
5. *diff* = *originalOpen-newOpen*
6. create cut areas m_j which satisfy *diff* > 1m
7. remove m_j that are smaller than d_{min}*d_{min} m^2. # the following loop checks whether each binary area m_j is truly a building area or not.
8. for each m_j
9. get the list of *diff* along boundary of m_j, that is, {*diff*($m_{j,b}$)}
10. if min({*diff* ($m_{j,b}$)}) > p_min (condition 1) OR prctile5({*diff* ($m_{j,b}$)}) > p_prctile5 (condition 2) OR prctile20({*diff* ($m_{j,b}$)}) > p_prctile20 (condition 3) OR (prctile80({*diff* ($m_{j,b}$)}) > p_prctile80 AND prctile40 ({*diff* ($m_{j,b}$)}) > p_prctile40) (condition 4), then
11. mark the m_j as a building area
12. end
13. end
14. replace the elevation of *originalOpen* with the value of *newOpen* where those m_j are marked as building areas
15. change the value of *buildingMask* to 1 where those m_j are marked as building areas
16. end

需要注意的是，判断一组点集是否为建筑物点云并不是通过设置一个限差来实现的，而是通过一组条件的组合来区分（code line 10）。因为在单个集合mj中的点云并非都属于同一类，可能同时包含了地面点和地物点，

条件1：中所有高程值都大于p_min，其中p_min可以根据先验知识来得到；

条件2：95%的网格高程大于p_partile5；

条件3：如果80%的网格高程大于p_partile20；

条件4：20%的网格高程大于p_partile80且40%的网格高程大于p_partile40。

这4个条件只需要有一个满足即可断定为建筑物。（原文中伪码10行与后面的解释存在数值上的差异）

实验结果对比：

TABLE 4. COMPARISON OF TOTAL ERRORS FOR ALL SAMPLES

Samples	Elmqvist (%)	Sohn (%)	Axelsson (%)	Pfeifer (%)	Brovelli (%)	Roggero (%)	Wack (%)	Sithole (%)	We (%)	Mean (%)	Min (%)	Max (%)
1(Sample11)	22.40	20.49	*10.76*	17.35	36.96	20.80	24.02	23.25	13.92	21.11	10.76	36.96
2(Sample 12)	8.18	8.39	3.25	4.50	16.28	6.61	6.61	10.21	3.61	7.52	3.25	16.28
3(Sample 21)	8.53	8.80	4.25	2.57	9.30	9.84	4.55	7.76	2.28	6.43	2.28	9.84
4(Sample 22)	8.93	7.54	3.63	6.71	22.28	23.78	7.51	20.86	3.61	11.65	3.61	23.78
5(Sample 23)	12.28	9.84	4.00	8.22	27.80	23.20	10.97	22.71	9.05	14.23	4.00	27.80
6(Sample 24)	13.83	13.33	4.42	8.64	36.06	23.25	11.53	25.28	3.61	15.55	3.61	36.06
7(Sample 31)	5.34	6.39	4.78	1.80	12.92	2.14	2.21	3.15	1.27	4.44	1.27	12.92
8(Sample 41)	8.76	11.27	13.91	10.75	17.03	12.21	9.01	23.67	34.03	15.63	8.76	34.03
9(Sample 42)	3.68	1.78	1.62	2.64	6.38	4.30	3.54	3.85	2.20	3.33	1.62	6.38
10(Sample 51)	23.31	9.31	2.72	3.71	22.81	3.01	11.45	7.02	2.24	9.51	2.24	23.31
11(Sample 52)	57.95	12.04	3.07	19.64	45.56	9.78	23.83	27.53	11.52	23.44	3.07	57.95
12(Sample 53)	48.45	20.19	8.91	12.60	52.81	17.29	27.24	37.07	13.09	26.41	8.91	52.81
13(Sample 54)	21.26	5.68	3.23	5.47	23.89	4.96	7.63	6.33	2.91	9.04	2.91	23.89
14(Sample 61)	35.87	2.99	2.08	6.91	21.68	18.99	13.47	21.63	2.01	13.96	2.01	35.87
15(Sample 71)	34.22	2.20	1.63	8.85	34.98	5.11	16.97	21.83	3.04	14.31	1.63	34.98%

Note: The algorithms with the lowest total error are underlined for each sample.

图3 不同方法对所有样例数据总误差的对比

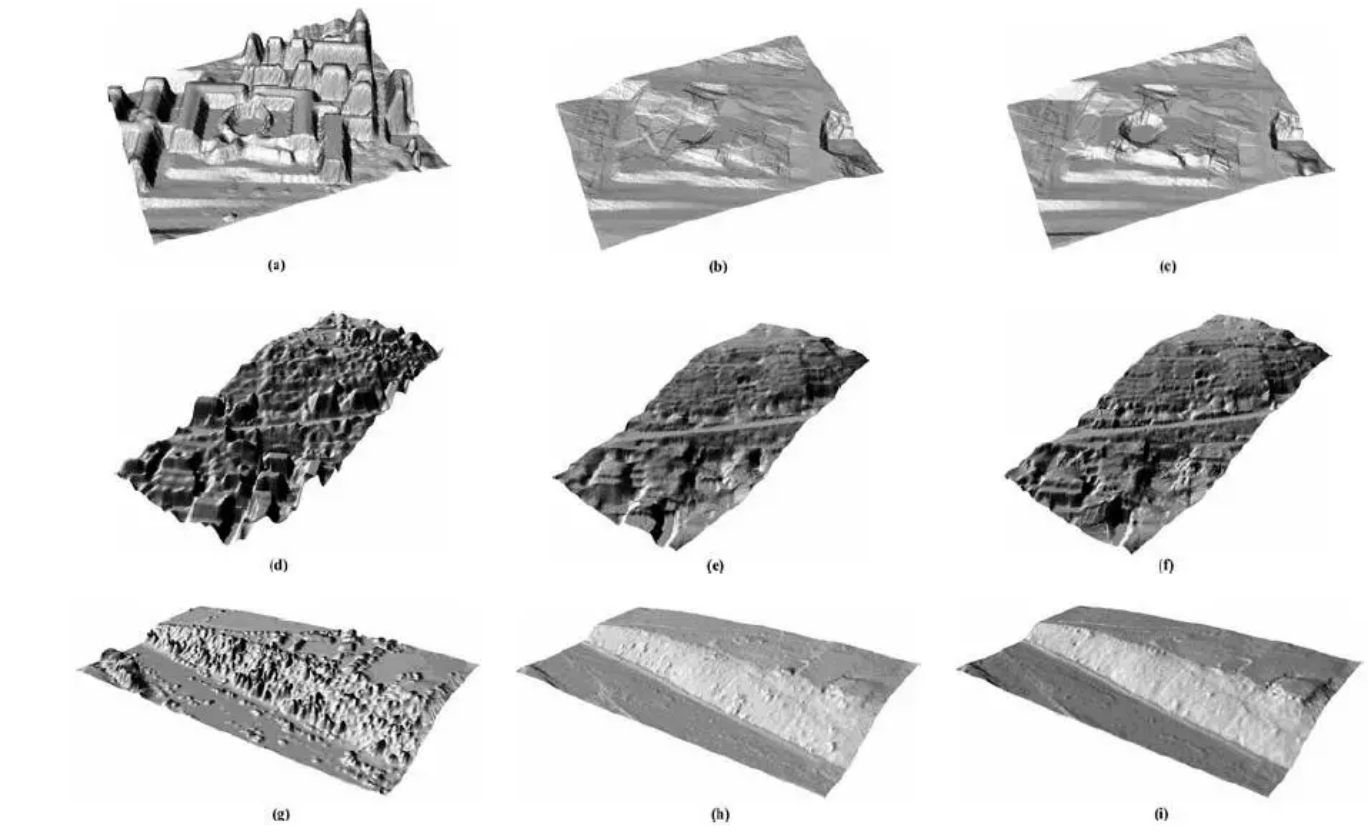


图4 a)到c)为DSM，d)到f)为滤波处理后的DEM，f)到i)为真实的DEM

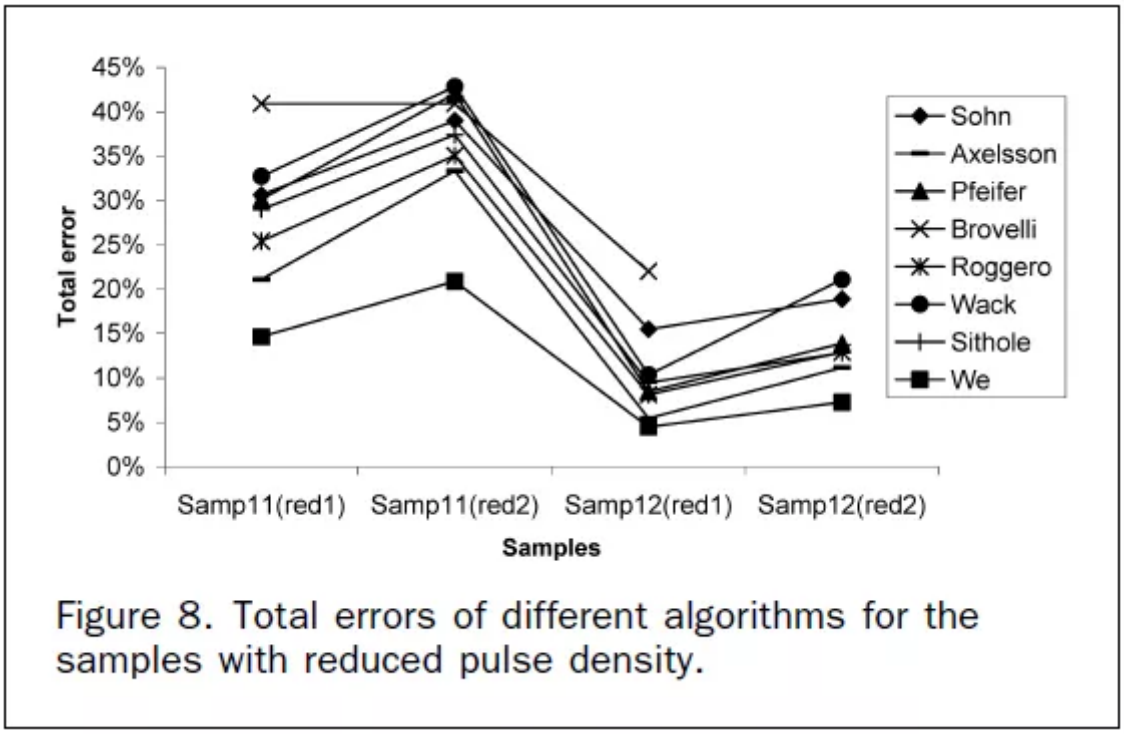


Figure 8. Total errors of different algorithms for the samples with reduced pulse density.

图5 不同算法对低密度样本数据的总误差对比图

2.3 机载激光雷达滤波

机载激光雷达系统经过多年的研究发展，硬件和系统技术已经很成熟，数据获取的精度也在不断提高，但数据后处理相对来说还是处于发展停滞状态，还有很多问题没有解决。国内外众多学者提出了多种滤波算法，目前机载LiDAR数据滤波算法主要有：形态学滤波算法、线性迭代最小二乘滤波算法、基于地形坡度滤波算法、三角网迭代滤波算法、曲面平面滤波算法以及基于数据分割滤波算法等几种方法。

基于数学形态学滤波算法

德国斯图加特大学研究者,indenberger利用数学形态学的滤波算子开运算，对测试用机载Lidar点云数据进行了处理，对于测试用机载LIDAR点云数据进行了处理，对于运算实验得到的结果，采取自回归的过程对其加以改进。但是这一改进算法只能在数据点排列非常非常有序的情况下进行，故只能适用于类型为剖面式的激光脚点数据。多数图像处理软件中的形态学算法均采用规则网格形式存储的数据结构，但这必将丢失数目庞大的换件地形要素的信息，某些情况下，当地面的数据点和地标物体的数据点间产生内插操作时，由于进行了内插处理，使得高程数据十分平滑，会破坏原始高程变化的特征，，反而会给分类和过滤带来困难。为保证原始信息的完整性，处理是应尽可能利用原始离散点云数据，只是该算法的实现十分耗时。1996年，Kilian等人利用形态学开运算，以移动窗口为平台，认定地面数据点，即为在窗口中拥有最小高程值的点，即使高程大于此点，只要在规定阈值内，仍被当作地面数据点，此时，参考不同大小的移动窗口，可以设定相应的权，提出的算法通常用来在多个窗口中反复处理，这些窗口大小不一，最终根据不同的权，用内插的方法，生成DEM。

基于线性迭代最小二乘滤波算法

维也纳大学的Kraus和Dfeifer教授首次提出了将LiDAR点云数据进行迭代最小二乘线性内插的滤波算法。其中心思想是使用低维的多项式曲线实现对地形起伏不大的扫描区域内数据的滤波处理。该算法的基本出发点是位于地物的激光脚点的高程比其对应的地面点的高程高，对激光焦点进行线性最小二乘内插后，拟合一个高程拟合面，激光脚点的高程与该点在拟合面上的拟合高程之差不服从正态分布。

基于坡度滤波算法

这种方法类似于数学的形态算法内常用的腐蚀算子。由地形坡度的变化决定最优滤波函数，对滤波窗口的大小作适当调整以保证倾斜地形信息的完整性，同时多提供几个可以用来筛选的阈值，选择性多的情况下可以确保真正存在于地标的激光数据点不会有被过滤掉的危险。关于如何设置最恰当的参数依然岁地形的变化而变化。

其基本思想是：当两个邻近的激光脚点间具有较大的高程差异时，有地形急剧变化导致这一现象的情况比较少。即相邻点间高度差的取值有时候会超过所规定的阈值，这种前提下，点与点之间的距离的高程，产生这一现象的具体原因一般为两数据点钟，一个在地表；一个在植被上，有时也会在其他地物的各种部位。描述的算法是通过比较两点间的高度差来决定是否采纳所选定的数据点，距离函数可有两点之间的高度差值定义。点云密度越小，分类误差就越大，滤波效果也就越差；反之，密度越高，误差会变小，则滤波效果也会增强。此方法难以实现矮小地面植被被激光脚点的过滤。

基于曲面平面滤波算法

基于平面/曲面的滤波算法，其核心在于参数曲面的选取。经典的方法使用最小二乘用于表面拟合。Kraus和pfeifer，pfeifer等应用具有最小二乘的线性预测方法提出了一种分等级的内插方法。该方法是通过减少地面之上的点和奇异点的权值，来确定最佳的参数曲面。Hu和Tao分析了来自道路和植被的多次回波信息特征，以有粗到细的方式，预先提取出地面点。由于这些方法的假设是基于一个连续的表面模型，要保持在断裂线上的地面点很困难，在邻近断裂线附近会产生较大误差。

有些方法使用TIN模型作为参数曲面，但是这些方法用于非连续的地形表面。TIN模型中的地面点选取，常用的方法是对TIN模型进行分等级加密，逐步增加TIN的精度和选取地面点的数目，以得到最终的DEM。由于这些方法均假定地形表面是局部平坦的，处理结果有时会导致山地和丘陵地区的DEM比真实地形平坦。

基于布料模拟滤波算法

布料滤波算法是模拟布料自然下落的物理过程，假设一个柔软的布料在重地作用下落在地形表面上并与之紧紧贴合，则布料最终的形状便反映了DSM形状，若最初先将地形进行上下颠倒，则布料最终反映的是DEM的形状，该算法运用到点云滤波领域，大致过程是先将点云上下翻转，构建质点弹簧模型并将其覆盖在翻转后的点云表面上。根据质点所受的重力和弹力来调整质点的位置从而模拟颠倒的地形表面。最后通过分析激光点与模拟的地形表面之间的距离，区分地面点与非地面点。

2.4 总结

由上文分析可以得出结论：点云处理算法的分类有很多，其中也不乏重复的内容，但都各有其优缺点和适应范围，在具体应用中应该综合分析图像特点采用不同的算法或者改进算法。

【参考文献】

[1] Vollmer,J.and Mencl,R.and Mueller, H.Improved Laplacian smoothing of noisy surface meshes[J].Wiley Online Library,1999,10.

[2] 罗大兵,高明,王培俊.逆向工程中数字 化测量与点云数据处理[J].机械设计与 制 造 ,2005,4.

[3] 董明晓,郑康平.一种点云数据噪声点 的随机滤波处理方法[J].中国图象图形 学 报 :A辑 ,2004,11.

[4] 龙鹏. MRI医学图像增强与分割新方法[D]. 中国科学院大学, 2015.