《微波遥感》课程集中实习

实习一 SAR 图像辐射处理及斑点噪声滤波

实习报告

目录

一、实习目的

本程序的主要编写目的为：使用 GDAL 库编程，实现多种 SAR 影像的滤波方法。

具体步骤为，通过对 SLC 影像进行波段运算，首先实现幅度图生成，然后根据幅度图编写函数实现多种对 SAR 图像斑点噪声的滤波方法。

本程序报告主要囊括四部分：

1.多种斑点噪声滤波算法原理

利用所提供的 SAR 图像滤波算法相关资料及课堂教学内容，深入学习滤波算法实现过程。

2. 数据预处理

GDAL 库为一套开源遥感和 GIS 数据输入输出库，支持几乎所有主流的遥感和 GIS 数据读写，后续实习也将使用该库进行数据读写，根据所提供的资料，熟悉 GDAL 库的遥感图像的输入和输出实现过程。

3. 滤波算法的编程实现

根据 1）和 2）的基础，完成不少于三种滤波算法，本次实习实现了中值滤波、均值滤波、 Sigma 滤波、Lee 滤波及其增强算法。

4. 算法评价

选择典型地物对实现的算法进行定性和定量评价。

二、实习原理

2.1 SAR 影像斑点噪声

2.1.1 斑点噪声产生的机理

SAR 成像系统是基于相干原理的，而这一理论基础存在着原理性缺陷，这个缺陷表现为：在雷达回波信号中，相邻像素点的灰度值会由于相干性而产生一些随机的变化，并且这种随机变化是围绕着某一均值而进行的，这样就在图像中产生了斑点噪声。而斑点噪声的产生是由于 SAR 成像所基于的相干原理所造成的缺陷，因此是不可避免的。

从产生机理上讲，SAR 图像中的斑点噪声是由于雷达目标回波信号的衰落现象所引起的。而信号的衰落过程是这样产生的：同时被照射的有多个散射体，当雷达目标和雷达站之间具有相对运动时，这多个散射体与雷达之间具有不同的路程长和不同的径向速度，这使得雷达接收机接收到的信号产生一定的随机起伏，从而使 SAR 对目标散射系数的测量产生很大的偏差。最终表现在图像上，就产生了不可避免的斑点噪声现象。因此，斑点噪声的不可避免性决定了要想得到高质量的 SAR 图像，如何有效地抑制斑点噪声是关键所在。

2.1.2 斑点噪声的乘性模型

在 SAR 图像中，斑点噪声是由于信号的衰落引起的，而且通过对 SAR 图像的观察，人们发现该图像具有这样的特点:在均匀区域，被斑点污染得越厉害的区域，在图像上表现得越亮，因此，人们设想斑点噪声的模型为乘性的。后来通过对 SAR 图像的统计，对斑班点完全发育的 SAR 图像建立了乘性噪声模型这个模型成为人们研究 SAR 的基础。

完全发育的斑点噪声的概念是由 Goodman 提出的，斑点噪声只有在每个分辨单元内，必须同时满足下列的三个条件才是完全发育的斑点噪声，而这个三个条件依次为:

(1) 有大量的散射体，且相位和幅度都统计独立的;

(2)不同散射体的幅度服从同一的统计分布;

(3)它们的相位是上的均匀分布。

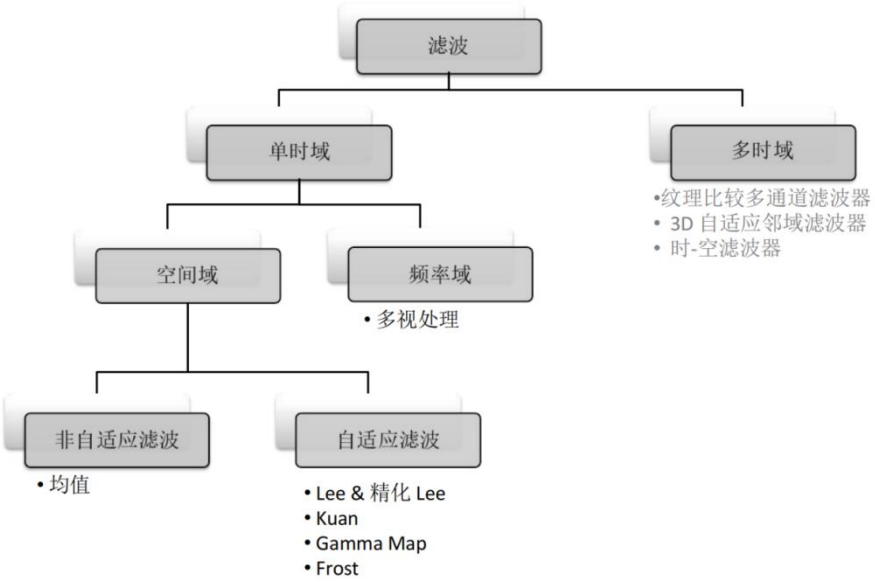
斑点完全发育的区域，表现在图像上，为均匀区域或者是弱纹理区域。

2.2 SAR 影像滤波算法

2.2.1 滤波算法概述

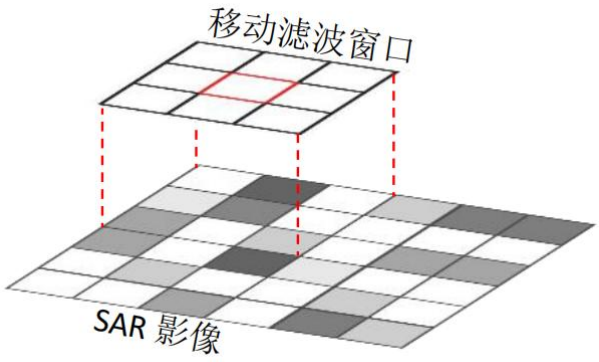
目前已有大量的雷达相干斑抑制算法，这些算法可分为成像前的多视平滑预处理和成像后的滤波两大类。而成像后的滤波又包括空域滤波和频域滤波两种。

为了减少相干斑噪声，早期的方法是在 SAR 成像处理中，通过降低处理器带宽形成多视图子图像，然后对多视子图像进行非相干叠加来降低相干斑噪声。这种非相干叠加来降低斑点噪声的方法称为多视处理。多视处理通过牺牲 SAR 图像的空间分辨率为代价来对相干斑进行抑制，已不能满足空间高分辨率的要求。空域滤波方法是利用图像像素的空间相关性对相干斑进行滤波，一般是利用一个滑动窗口，然后对窗口内的像素进行加权得到窗口中心点的像素值。频域的方法主要是利用小波变换，比较著名的有小波软阈值方法，基于小波变换和多尺度分析的滤波方法。



2.2.2 传统滤波方法

传统滤波算法包括均值滤波、中值滤波等。这类算法的特点是直接对图像进行处理，没有考虑任何噪声模型，也没有考虑噪声的统计特性。这些算法实现起来比较简单，但效果不太理想。它们计算简单，速度快，均匀区域的斑点噪声去除效果较好。缺点是细节保持得不好，图像边缘变模糊，点目标损失大，随着处理窗口的增大，图像的整体模糊和分辨率下降更严重。正是由于这两种传统滤波算法不适合相干斑噪声的乘性特点，实际中较少采用。



（1）均值滤波：

①原理：

均值滤波是将平滑窗口内所有像元的灰度值进行平均计算，然后赋给平滑窗口的中心像元。

②数学表达式：

$$R_{i,j}=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}DN_{i,j}$$

式中，$R_{i,j}$为滤波后中心元素灰度值，$DN_{i,j}$为滤波窗口内各个像元的灰度值，窗口大小为$n\times n$。

（2）中值滤波：

①原理：

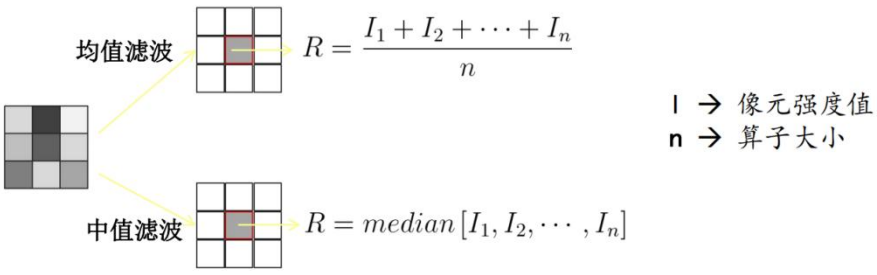
中值滤波是一种非线性信号处理技术。它假设信号有极端的数值，即认为在平滑窗口内噪声是极大值或极小值。中值滤波将平滑区域内所有像素的中值作为平滑区域中心像元值。

②数学表达式：

设$DN_{i,j}$为奇数项离散系列（i=1，…2n-1，j=1,…2n-1），$DN_{i,j}^{'}$为$DN_{i,j}$按大小重新排列的奇数项离散系列，则中值滤波的数学表达式为：

$$R_{i,j}=DN_{n,n}^{'}$$

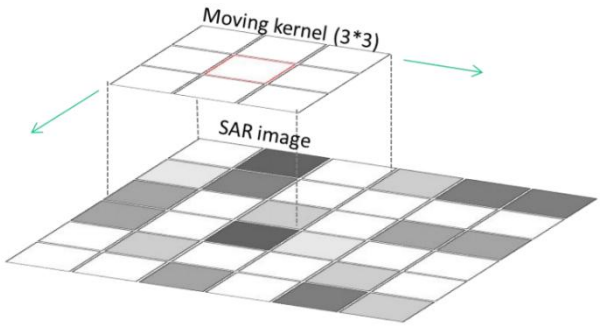
式中，$R_{i,j}$为滤波后的中心像元灰度值，$DN_{i,j}$为滤波前平滑模板内各个像元的原始灰度值，$DN_{i,j}^{'}$为平滑模板内各个像元重新排列后的灰度值，窗口大小为$n\times n$。



2.2.3 基于局域统计特性的自适应滤波算法

自适应滤波是近 30 年以来发展起来的一种最佳滤波方法。它是在维纳滤波、Kalman 滤波等线性滤波基础上发展起来的一种最佳滤波方法。由于它具有更强的适应性和更优的滤波性能，从而在工程实际中，尤其在信息处理技术中得到了广泛的应用。自适应滤波的研究对象是具有不确定的系统或信息过程。这里的“不确定性”是指所研究的处理信息过程及其环境的数学模型不是完全确定的。其中包含一些未知因素和随机因素。

- 自适应滤波器在降低斑点噪声的同时，保持边缘信息
- 自适应滤波器对移动窗口内的相邻像元的灰度信息做统计分析



（1）Sigma 滤波：

①原理：

该算法建立在 SAR 图像的乘性噪声模型上，假设斑点噪声的分布为高斯分布，窗口内的像素灰度值与其中心像素的灰度值比较接近。其基本原理为：Sigma 滤波器将 2σ 范围内的像素进行平均，即可去除差别过大的象素的影响。我们知道，对于一维高斯分布，采样点落在 2σ 区间的概率是 93.5%。在窗口滤波过程中，只选取窗口内像素灰度值落在 2σ 范围内的点，将它们的平均值作为中心像素灰度的估计，而其它变化显著的像素则被视作边缘而不做滤波处理。

②数学表达式：

$$R=\frac{\sum_{k=i-m}^{i+m}\sum_{l=j-n}^{j+n}\delta_{kl}g_{kl}}{\sum_{k=i-m}^{i+m}\sum_{l=j-n}^{j+n}\delta_{kl}}$$

$$\delta_{kl}=\begin{cases}1 & \bar{g}_{ij}(1-2\sigma_F)\le g_{kl}\le \bar{g}_{ij}(1+2\sigma_F)\\ 0 & otherwise\end{cases}$$

$$\sigma_F=\sqrt{\sigma}/\bar{g}$$

其中，滤波窗口内各像元灰度的平均值 $\bar{g}_{ij}$ 作为滤波中心像元 (i,j) 的平均值；窗口内标准差 σ_{ij} 作为滤波中心像元点 (i,j) 的标准差。

（2）Lee 滤波及其增强算法：

①原理：

Lee 滤波基于完全发育的斑点乘性噪声模型，假定先验均值和方差可由均质区内计算局域的均值和方差来得到，它是使用滤波窗口内样本均值和方差的自适应滤波算法。该方法是以 MMSE（最小均方误差）准则作为基础，是固定窗口中观察强度 g 和局部平均强度 $\bar{g}_{ij}$ 的线性组合，是一个优化的线性滤波器。该方法是在图像上对每个像元逐个滤波移动的过程，局部统计量随着空间位置的改变而改变。

②数学表达式：

$$g_{ij}^{'}=\bar{g}_{ij}w+g(1-w)$$

其中：$w=1-C_u^2/C_I^2$ 是 Lee 滤波的权函数；

$c_u, c_{\max}$ 为阈值；$C_I=\sigma/\overline{g_{ij}}$；

$$C_u=0.5227/\sqrt{L}$$；

$C_{\max}=\sqrt{3}C_u$；L 为成像视数。

③增强算法：

Lee 滤波算法是在均质区域的基础上推导得到的，但这一点事实上在真实的 SAR 图像中是不成立的。因此，Lee 滤波方法对于在保持边缘等细节信息方面不是十分理想，但同质区则比较有效。针对 Lee 算法的缺陷，A. Lopes 提出根据图像不同区域采用不同滤波器的方法。A. Lopes 把一个图像分为三类区域：第一类是均匀区域，其中的相干斑噪声可以简单地用均值滤波平滑掉；第二类是不均匀区域，在去除噪声时应保留纹理信息，应用 Lee 滤波；第三类是包含分离点目标的区域，滤波器应尽可能地保留原始值。

增强的 Lee 滤波表达式为：

$$\begin{cases} g_{ij}^{'}=\bar{g}_{ij} & C_I \le C_u \\ g_{ij}^{'}=\bar{g}_{ij}w+g(1-w) & C_u \le C_I \le C_{\max} \\ g_{ij}^{'}=g_{ij} & C_I \ge C_{\max} \end{cases}$$

（3）Kuan 滤波及其增强算法：

①原理：

Kuan 滤波算法假设噪声为与信号相关的加法噪声，然后运用最小方差估计获得固定窗口中观察强度 g 和局部平均强度 $\overline{g}_{ij}$ 的线性组合。Kuan 滤波器与 Lee 滤波器的区别在于用一个信号加上一个依赖于信号的噪声来表示乘性模型的相干斑噪声。该方法是在图像上对每个像元逐个滤波移动的过程，局部统计量随着空间位置的改变而改变。

②数学表达式：

$$g_{ij}^{'}=\bar{g}_{ij}w+g(1-w)$$

其中：g'_{ij} 为平滑处理后的像元灰度值；g_{ij} 为平滑窗口中各像元的原始灰度值；

$\overline{g}_{ij}$ 为窗口内像元灰度平均值；

$w=-\dfrac{1-C_u^2/C_I^2}{1+C_u^2}$ 是 Kuan 滤波的权函数；

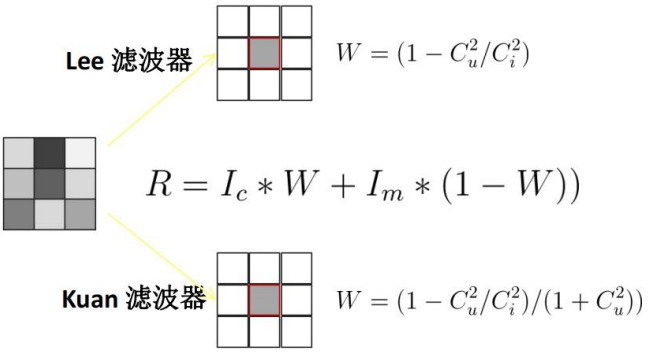
$c_u, c_{\max}$ 为阈值；$C_I = \sigma / g_{ij}$；σ_I 为局部标准差；

$C_u = 1/\sqrt{L}$；$C_{\max} = \sqrt{1+2/L}$；L 为成像视数。

③增强算法：

Kuan 滤波算法与 Lee 滤波算法一样，存在着保持边缘等细节信息不佳的问题。因此，它也有对应的增强算法。A. Lopes 提出的增强的 Kuan 滤波表达式为：

$$\begin{cases} g_{ij}' = \bar{g}_{ij} & C_I \le C_u \\ g_{ij}' = \bar{g}_{ij} w + g(1-w) & C_u \le C_I \le C_{\max} \\ g_{ij}' = g_{ij} & C_I \ge C_{\max} \end{cases}$$



（4）Frost 滤波及其增强算法：

①原理：

Frost 滤波算法假定斑点噪声是乘性噪声的前提下，并假设 SAR 影像是平稳过程，对影像进行滤波。Frost 滤波器的冲激响应为一双边指数函数，近似为低通滤波器，其滤波器参数由图像局域方差系数决定。冲激响应的衰减快慢取决于局域方差系数的大小，与其成正比关系。Frost 自适应滤波器是以权重 M 值为自适应调节参数的环形对称滤波器。

②数学表达式：

$$g_{ij}' = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij} \times M_{ij} \Big/ \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}$$

$$M_{ij} = \exp(-A_{ij} \times T_{ij})$$

$$A_{ij} = \sigma_{ij} / \bar{g}_{ij}^{2}$$

其中：g_{ij}' 为平滑处理后的像元灰度值；

g_{ij} 为平滑窗口中各像元的原始灰度值；

$\overline{g}_{ij}$ 为窗口内像元灰度平均值；

M_{ij} 为平滑窗口中各个对应像元的权重指数；

T_{ij} 为平滑窗口内中心像元到其邻像元的绝对距离；

σ_{ij} 为平滑窗口中像元值的方差；n^2 是平滑窗口的大小；

$C_I = \sigma_{ij} / g_{ij}$；$C_u = 1/\sqrt{L}$；$C_{\max} = \sqrt{1+2/L}$；$L$ 为成像视数。

③增强算法：

A. Lopes 提出的增强的 Frost 滤波表达式为：

$$C_I < C_u \text{ 时} \qquad g'_{ij} = \overline{g_{ij}};$$

$$C_u \le C_I \le C_{\max} \text{ 时} \qquad g'_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij} \times M_{ij} \Big/ \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij};$$

$$C_I > C_{\max} \text{ 时} \qquad g'_{ij} = g_{ij};$$

$$R = \frac{(P_1 * M_1 + P_2 * M_2 + \cdots + P_n * M_n)}{(M_1 + M_2 + \cdots + M_n)}$$

（5）GAMMA-MAP 滤波算法：

①原理：

最大后验概率（MAP）滤波法是假设相干斑为乘性 gamma 分布，所以又称 Gamma MAP 滤波器。在知道 σ 的概率密度函数（Probability Density Function，PDF）先验知识情况下，就能获取更多的信息。这就是根据先验分布和似然函数的 MAP 滤波方法。

②数学表达式：

$$C_I < C_u \text{ 时} \qquad g'_{ij} = \overline{g_{ij}};$$

$$C_I > C_{\max} \text{ 时} \qquad g'_{ij} = g_{ij};$$

$$C_u \le C_I \le C_{\max} \text{ 时}$$

$$g'_{ij} = \frac{\overline{g_{ij}}(\alpha - L - 1) + \sqrt{\overline{g_{ij}}^2(\alpha - L - 1)^2 + 4\alpha L \overline{g_{ij}} g_{ij}}}{2\alpha} \quad (\text{L} \neq 1)$$

$$g'_{ij}=\frac{\overline{g_{ij}}(\alpha-2)+\sqrt{\overline{g_{ij}}^{2}(\alpha-2)^{2}+8\alpha\overline{g_{ij}}g_{ij}}}{2\alpha}$$（L=1）。

其中：g'_{ij} 为平滑处理后的像元灰度值；

g_{ij} 为平滑窗口中各像元的原始灰度值；

$\overline{g}_{ij}$ 为窗口内像元灰度平均值；

$\alpha=(1+C_u^2)\big/(C_I^2-C_u^2)$；

σ_{ij} 为平滑窗口中像元值的方差；

n^2 是平滑窗口的大小；

$C_I=\sigma_{ij}\big/\overline{g_{ij}}$；$C_u=1\big/\sqrt{L}$；$C_{\max}=\sqrt{1+2/L}$；$L$ 为成像视数。

$$R_f=(B*I_m+\sqrt{D})/(2*A)$$

2.3 滤波算法评价

（1）均值（Mean）和方差（Standard Deviation ， STD）

图像均值是整个图像的平均强度，它反映了图像的平均灰度，即图像所包含目标的平均后向散射系数；图像方差代表了图像区域中所有点偏离均值的程度，反映了图像的不均匀性。图像的均值和方差是反映图像整体特征的指标，一般情况下，如果地形、含水量（复介电常数）和表面粗糙程度不同，则会有不同的后向散射系数，反映到 SAR 图像中有不同的图像均值。图像区域中的地形差异大，人工目标多，图像的灰值变化大，对应的图像的方差变化也就越大。所以应当尽量保持图像的均值，同时减少图像的方差。

若图像区域大小为 $M\times N$，图像在 (i,j) 处的像素灰度值为 $I_{i,j}$，则图像均值 μ 和图像方差 σ^2 分别定义如下：

$$\mu = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{i,j}$$

$$\sigma^2 = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} (I_{i,j} - \mu)^2$$

（2）等效视数（Equivalent Number of Looks，ENL）

等效视数是衡量一幅图像斑点噪声相对强度的一种指标，也是衡量滤波器滤波性能的一种指标，又称为有效视数。当均匀区域内等效视数越大，则滤波器的滤波效果越好；当纹理区域内等效视数越小，说明滤波器保持纹理信息的能力越好。ENL 定义为：

$$M_{ENL} = \frac{\mu^2}{\sigma^2}$$

式中 μ 和 σ^2 分别是 SAR 图像区域内的均值和方差。

（3）辐射分辨率（Radiation Resolution）

辐射分辨率表示区分 SAR 目标后向散射系数的能力，是衡量 SAR 系统区分相邻分布目标的能力的一种量度。它的好坏直接影响 SAR 图像判读和定量化应用。辐射分辨率的大小由消除相干斑噪声的多少决定，因此好的相干斑噪声抑制算法能提高辐射分辨率。它定义为一个分辨单元内反射信号相对于平均值的绝对偏差与平均值的比值。辐射分辨率 r 定义为：

$$r = 10\lg(\frac{1}{\sqrt{M_{ENL}}} + 1) = 10\lg(\frac{\sigma}{\mu} + 1)$$

其中，M_{ENL} 为等效视数。

三、实习数据与环境

3.1 SAR 数据

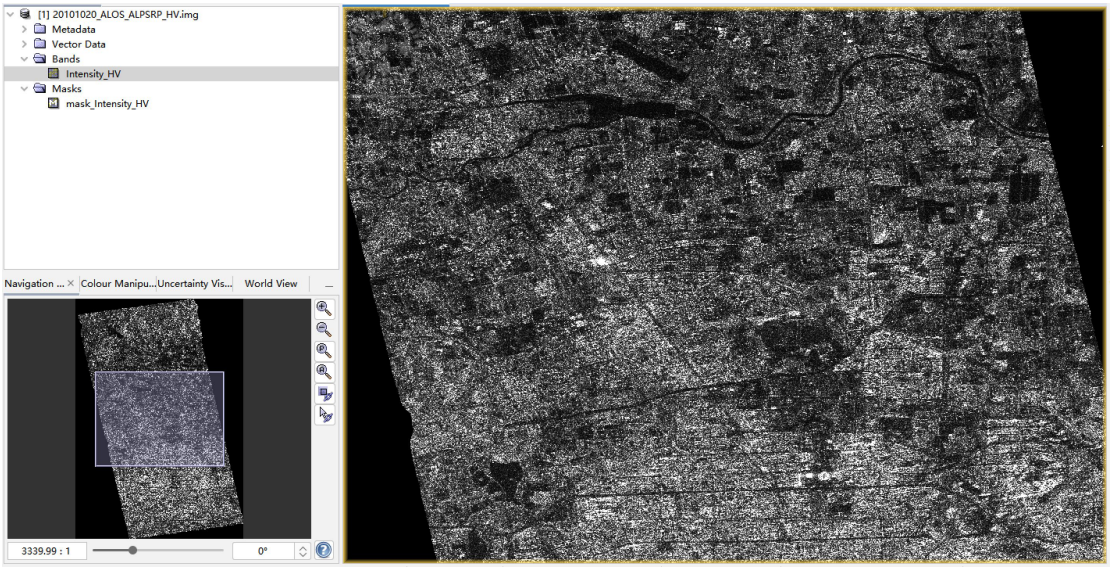
本次实习提供了三组数据，分别为 19990818_ERS、20101020_ALOS_ALPSRP 和 20161219_TerraSAR-X 三组不同时期不同传感器的 SAR 影像。

本程序所采用原始数据产品数据为 20101020_ALOS_ALPSRP 的 HV.Img 影像。

ALOS 是日本的对地观测卫星，ALOS 卫星载有三个传感器：全色遥感立体测绘仪（PRISM），主要用于数字高程测绘；先进可见光与近红外辐射计－2（AVNIR－2），用于精确陆地观测；相控阵型 L 波段合成孔径雷达（PALSAR），用于全天时全天候陆地观测。PALSAR 是一主动式微波传感器，它不受云层、天气和昼夜影响，可全天候对地观测，比 JERS-1 卫星所携带的图 4 SAR 传感器性能更优越。该传感器具有高分辨率、扫描式合成孔径雷达、极化三种观测模式，使之能获取比普通 SAR 更宽的地面幅宽。

20101020_ALOS_ALPSRP_HH.hdr	13/3/2022 下午 9:45	HDR 文件	1 KB
20101020_ALOS_ALPSRP_HH.img	13/3/2022 下午 9:45	光盘映像文件	43,059 KB
20101020_ALOS_ALPSRP_HV.hdr	13/3/2022 下午 9:45	HDR 文件	1 KB
20101020_ALOS_ALPSRP_HV.img	13/3/2022 下午 9:45	光盘映像文件	43,059 KB

本次实习提供了 SAR 影像，含有明显的斑点噪声，用以测试滤波算法的效果，如下所示。



3.2 GDAL 库

本次实习使用 C++语言，在 Visual Studio 2022 环境下编写，需要配置 GDAL 环境。

GDAL(Geospatial Data Abstraction Library)是由开源地理空间基金会发布的一个栅格和矢量的地理空间数据转换库。GDAL 库会创建一个栅格或矢量的抽象数据模型，通过这个抽象模型，我们可以借助程序来对栅格或矢量数据进行读取。

GDAL 是一个在 X/MIT 许可协议下的开源栅格空间数据转换库。它利用抽象数据模型来表达所支持的各种文件格式。它还有一系列命令行工具来进行数据转换和处理。OGR 是 GDAL 项目的一个分支，提供对矢量数据的支持。有很多著名的 GIS 类产品都使用了 GDAL/OGR 库，包括 ESRI 的 ARCGIS 9.3，Google Earth 和跨平台的 GRASS GIS 系统。利用 GDAL/OGR 库，可以使基于 Linux 的地理空间数据管理系统提供对矢量和栅格文件数据的支持。

GDAL 数据集是基于 OpenGIS 栅格数据的标准，一个完整的栅格数据集包含它的各个栅格波段以及对这些栅格波段的描述信息统一组成的，所有的波段具有相同的地理坐标、投影信息，可以进行地理坐标转换等。描述性信息包括数据地理坐标、投影信息以及一些元数据，

元数据是采用字符串的形式组织成字符串列表，表达形式为：“名称：值”。

GDAL 提供对多种栅格数据的支持，包括 Arc/Info ASCII Grid(asc)，GeoTiff (tiff)，Erdas Imagine Images(img)，ASCII DEM(dem) 等格式。GDAL 使用抽象数据模型(abstract data model)来解析它所支持的数据格式，抽象数据模型包括数据集(dataset)，坐标系统，仿射地理坐标转换(Affine Geo Transform)， 大地控制点(GCPs)， 元数据(Metadata)，栅格波段(Raster Band)，颜色表(Color Table)，子数据集域(Subdatasets Domain)，图像结构域(Image_Structure Domain)，XML 域(XML:Domains)。

GDALMajorObject 类：带有元数据的对象。

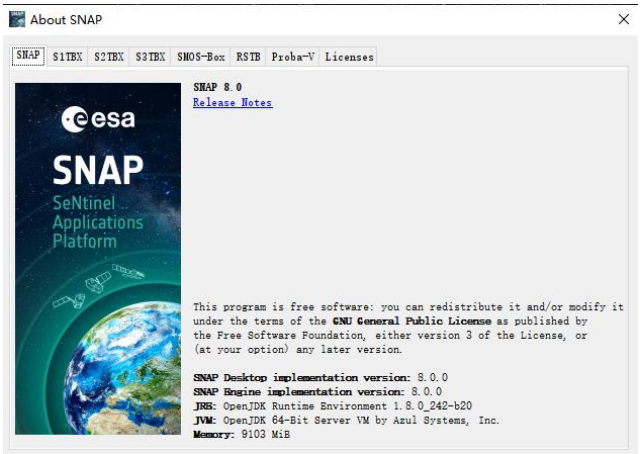
GDALDdataset 类：通常是从一个栅格文件中提取的相关联的栅格波段集合和这些波段的元数据;GDALDdataset 也负责所有栅格波段的地理坐标转换(georeferencing transform)和坐标系定义。

GDALDriver 类：文件格式驱动类，GDAL 会为每一个所支持的文件格式创建一个该类的实体，来管理该文件格式。

GDALDriverManager 类：文件格式驱动管理类，用来管理 GDALDriver 类。

3.3 SNAP 平台

ESA SNAP (Sentinel Application Platform) 哨兵数据应用平台，是欧空局针对哥白尼计划中哨兵数据任务开发的免费开源的软件。SNAP 软件中集成了多种算法的实现。



四、实习内容

4.1 使用 GDAL 库导入与输出影像

在编写滤波算法前，需要先熟悉使用 GDAL 库。使用 GDAL 导入影像，需要先注册驱动、定义输入数据集、最后定义输出数据集、输出影像。

（1）算法原理及流程

①注册 GDAL 驱动：使用 GDAL，需要首先通过 GDALAllRegister()命令注册驱动。

②打开 GDAL 数据集：定义一个 GDAL 数据集：GDALDataset，输入文件路径打开。

③获取影像参数：使用 GetRasterXSize()、GetRasterYSize()等命令，可以获取影像的宽度、高度、波段数、投影信息等。

④打开单波段数据：使用 RasterIO 函数，将影像一个波段的数据取出待用。

⑤数据处理：后续步骤，通过各种滤波方法对取出的数据进行处理。

⑥GTif 格式输出驱动：在输出结果前，定义一个输出格式为 GeoTiff 的驱动。

⑦写入输出数据集：使用 RasterIO 函数，将处理后的数据写入输出影像，然后释放 GDAL。

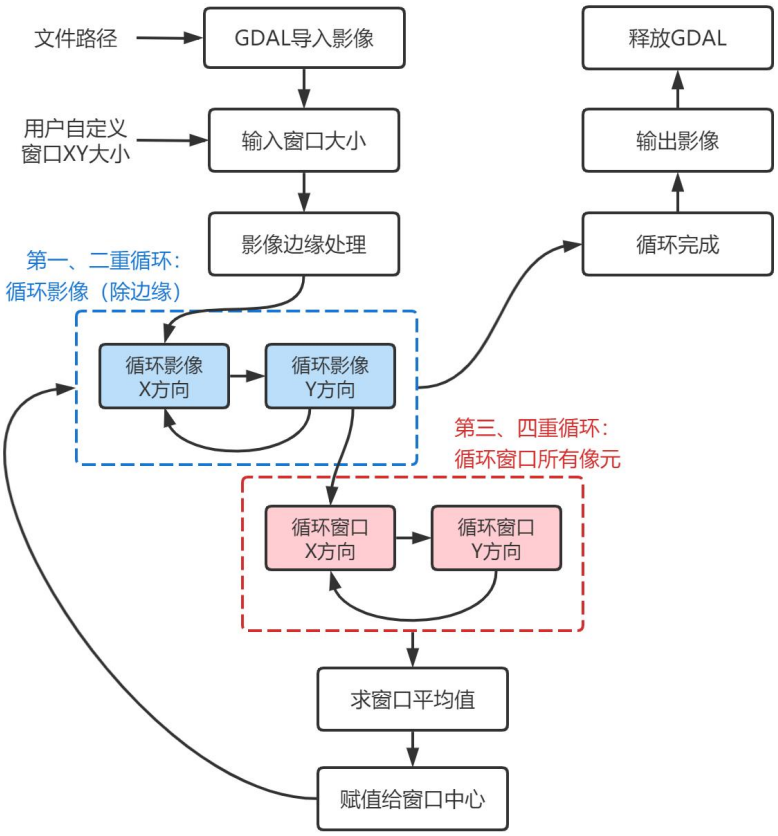
（2）代码实现

```
//注册 GDAL 库中的所有驱动程序
GDALAllRegister();
//支持中文路径
CPLSetConfigOption("GDAL_FILENAME_IS_UTF8", "NO");

//打开图像
const char* filepath = "2010_HV_cut.tif";
GDALDataset* image = (GDALDataset*)GDALOpen(filepath, GA_ReadOnly);
if (image == NULL)
{
	//AfxMessageBox("读取图像失败");
	cout << "读取图像失败";
}
else
	cout << "读取成功" << endl;
//定义图像的长宽
int W = image->GetRasterXSize();
int H = image->GetRasterYSize();
int C = image->GetRasterCount();
GDALDataType ImgType = image->GetRasterBand(1)->GetRasterDataType();	//数字图像类型

GDALRasterBand* band1 = image->GetRasterBand(1);	//第一个波段的影像读入到数据之中
unsigned char* bdata = new unsigned char[W * H];	//创建存放数据的内存
band1->RasterIO(GF_Read, 0, 0, W, H, bdata, W, H, ImgType, 0, 0);	//GDALRasterBand

//创建保存影像数据集
GDALDriver* imgDriver = GetGDALDriverManager()->GetDriverByName("GTiff");	//获取驱动
const char* outFilename1 = "MeanFilter.tif";
GDALDataset* outIMG1 = imgDriver->Create(outFilename1, W, H, 1, ImgType, NULL);

//保存影像
MeanFilter(bdata, newdata, W, H);
outIMG1->GetRasterBand(1)->RasterIO(GF_Write, 0, 0, W, H, newdata, W, H, ImgType, 0, 0);
GDALClose(image);
GDALClose(outIMG1);
delete []bdata;
delete []newdata;
```

4.2 均值滤波

（1）算法原理及流程

流程图如下所示：



（2）代码实现

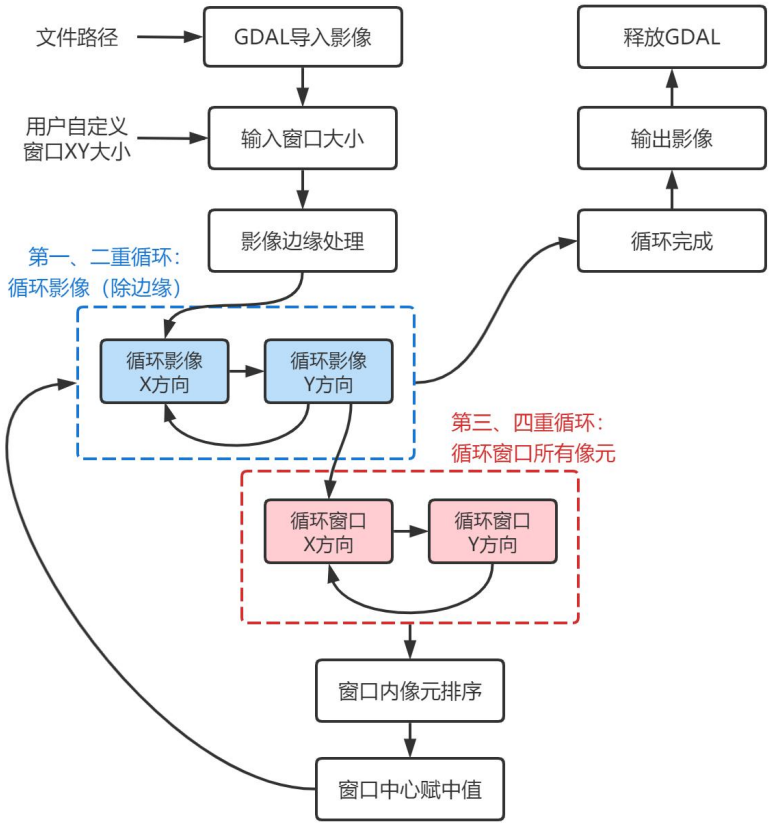
为节约篇幅，仅展示滤波算法中四重循环的核心部分，GDAL 导入与输出部分忽略：

```
    void MeanFilter(unsigned char* in, unsigned char* out, int W, int H){
    /*
    memcpy(void *dest,void *src,unsigned int count)
    由 src 所指内存区赋值 count 个字节到 dest 所指内存区
    */
    int sum = W * H * sizeof(unsigned char);     //图像所占容量
    memcpy((unsigned char*)out, (unsigned char*)in, sum);
    float sum1 = 0, count = 0, sum2 = 0;
    for (int j = 1; j < H - 1; j++)
    {
        for (int i = 1; i < W - 1; i++)
        {
            out[j * W + i] = (in[(j - 1) * W + (i - 1)] + in[(j - 1) * W + i] + in[(j - 1) * W + (i + 1)] +
                in[(j)*W + (i - 1)] + in[(j)*W + i] + in[(j)*W + (i + 1)] +
                in[(j + 1) * W + (i - 1)] + in[(j + 1) * W + i] + in[(j + 1) * W + (i + 1)]) / 9;
            sum1 += out[j * W + i];
            count++;
        }
    }
    cout << "使用<均值滤波>完成图像平滑" << endl;
}
```

4.3 中值滤波

（1）算法原理及流程

流程图如下所示：



（2）代码实现

为节约篇幅，仅展示滤波算法中四重循环的核心部分，GDAL 导入与输出部分忽略：

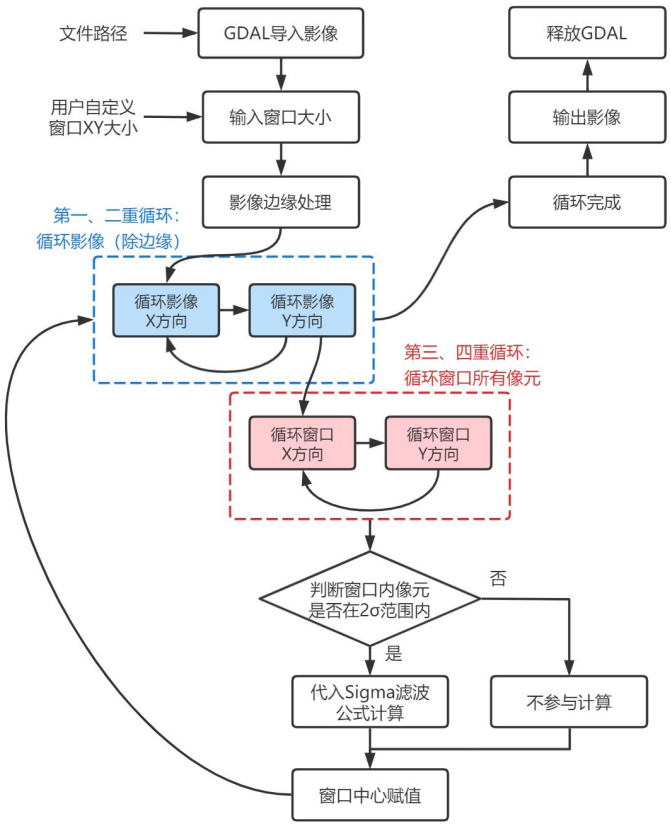
```
void MedianFilter(unsigned char* in, unsigned char* out, int W, int H)
{
    int sum = W * H * sizeof(unsigned char);    //图像所占容量
    memcpy((unsigned char*)out, (unsigned char*)in, sum);
    float sum1 = 0, count = 0, sum2 = 0;
    for (int j = 1; j < H - 1; j++)
    {
        for (int i = 1; i < W - 1; i++)
        {
            int k = 0;
            unsigned char win[9];
            for (int jj = j - 1; jj < j + 2; jj++)
            {
                for (int ii = i - 1; ii < i + 2; ii++)
                {
                    win[k++] = in[jj * W + ii];      //将窗口的九个元素记录到一个数组中
                }
                //排序函数，第 5 个为中值
```

```
                        for (int m = 0; m < 5; m++)
                        {
                            int min = m;
                            for (int n = m + 1; n < 9; n++)
                            {
                                if (win[n] < win[min])
                                    min = n;
                            }
                            unsigned char temp = win[m];
                            win[m] = win[min];
                            win[min] = temp;
                        }
                        out[j * W + i] = win[4];
                        sum1 += out[j * W + i];
                        count++;
                    }
                }
            }
            cout << "使用<中值滤波>完成图像平滑" << endl;
        }
```

4.4 Sigma 滤波

（1）算法原理及流程

流程图如下所示：



（2）代码实现

为节约篇幅，仅展示滤波算法中四重循环的核心部分，GDAL 导入与输出部分忽略：

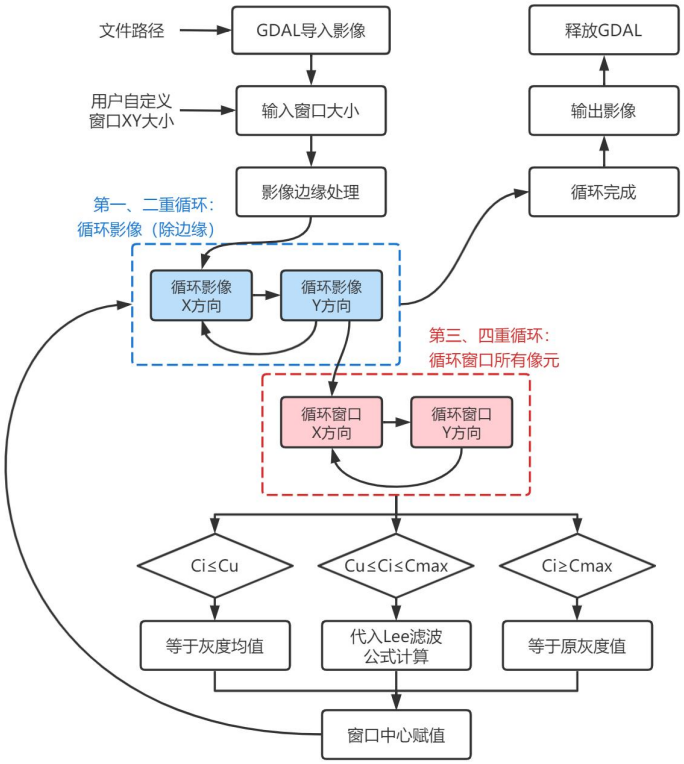
```
void SigmaFilter(unsigned char* in, unsigned char* out, int W, int H)
{
    int sum = W * H * sizeof(unsigned char);     //图像所占容量
    memcpy((unsigned char*)out, (unsigned char*)in, sum);
    float sum1 = 0, count = 0, sum2 = 0;
    for (int j = 1; j < H - 1; j++)
    {
        for (int i = 1; i < W - 1; i++)
        {
            int k = 0;
            unsigned char win[9];
            for (int jj = j - 1; jj < j + 2; jj++)
            {
                for (int ii = i - 1; ii < i + 2; ii++)
                {
                    win[k++] = in[jj * W + ii];      //将窗口的九个元素记录到一个数组中
                }
            }
            // 首先计算均值方差
            float sum = 0;
            for (int i = 0; i < k; i++)
                sum += win[i];
            float mean = sum / k; // 计算均值

            float sum2 = 0;
            for (int i = 0; i < k; i++)
                sum2 += (win[i] - mean) * (win[i] - mean);
            float variance = sum2 / k; // 计算方差

            float sigma_F = sqrt(variance) / mean;//计算 sigma_F
            float delta_kl = 0, R = 0;
            float sum3 = 0, sum4 = 0;

            for (int q = 0; q < k; q++)
            {
                if (win[q] >= mean * (1 - 2 * sigma_F) && win[q] <= mean * (1 + 2 * sigma_F))
                    delta_kl = 1;// 计算 delta_kl
                else delta_kl = 0;
                sum3 += delta_kl * win[q];
                sum4 += delta_kl;

            }
```

```
                if (sum4 > (k + 1) / 2)
                    R = sum3 / sum4;
                else
                {
                    R = ((float)in[(j + 1) * W + i] + (float)in[j * W + i + 1] + (float)in[j * W + i - 1] +
(float)in[(j - 1) * W + i]) / 4;
                }
                out[j * W + i] = R; //赋值给窗口中心
                sum1 += out[j * W + i];
                count++;
            }
        }
        cout << "使用< Sigma 滤波>完成图像平滑" << endl;
}
```

4.5 Lee 滤波及其增强算法

（1）算法原理及流程

流程图如下所示：



（2）代码实现

为节约篇幅，仅展示增强滤波算法中四重循环的核心部分，GDAL 导入与输出部分忽略：

```
    void LeeFilter(unsigned char* in, unsigned char* out, int W, int H)
{
    int sum = W * H * sizeof(unsigned char);     //图像所占容量
    memcpy((unsigned char*)out, (unsigned char*)in, sum);
    float sum1 = 0, count = 0, sum2 = 0;
```

```
        for (int j = 1; j < H - 1; j++)
        {
            for (int i = 1; i < W - 1; i++)
            {
                int k = 0;
                unsigned char win[9];
                for (int jj = j - 1; jj < j + 2; jj++)
                {
                    for (int ii = i - 1; ii < i + 2; ii++)
                    {
                        win[k++] = in[jj * W + ii];      //将窗口的九个元素记录到一个数组中
                    }
                }
                float sum = 0;// 首先计算均值方差
                for (int i = 0; i < k; i++)
                    sum += win[i];
                float mean = sum / k; // 计算均值

                float sum2 = 0, look = 1;
                for (int i = 0; i < k; i++)
                    sum2 += (win[i] - mean) * (win[i] - mean);
                float variance = sum2 / k; // 计算方差
                float C_u = 0.5227 / sqrt(look); // 计算 C_u
                float C_I = sqrt(variance) / mean;//计算 C_I
                float w = 1 - (C_u * C_u) / (C_I * C_I); // 计算 w
                // 以下是增强 Lee 滤波判断准则
                float C_max = sqrt(3) * C_u;//计算 C_max
                // 判断准则
                float g = 0;
                if (C_I < C_u)
                {
                    g = mean;
                }
                else if (C_I >= C_u && C_I <= C_max)
                    g = mean * (1 - w) + win[(k + 1) / 2 - 1] * w;
                else
                    g = win[(k + 1) / 2 - 1];
                out[j * W + i] = g; //赋值给窗口中心
                sum1 += out[j * W + i];
                count++;
            }
        }
        cout << "使用< Lee 滤波>完成图像平滑" << endl;
    }
```

五、实习结果与分析

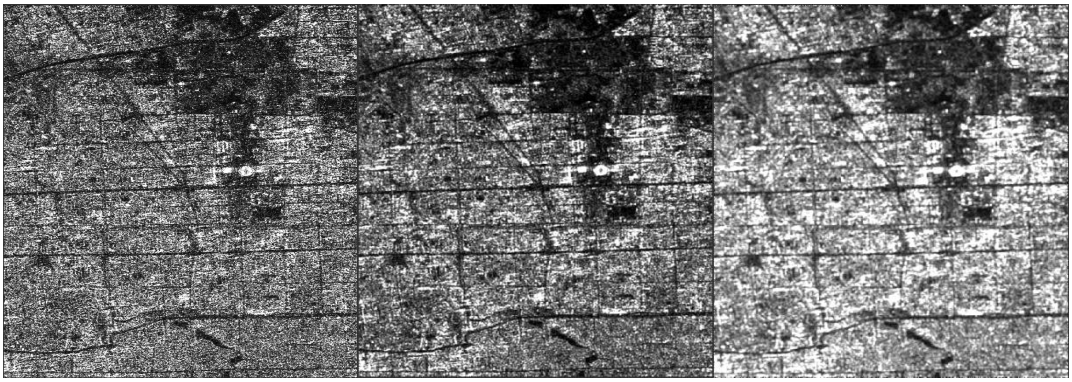
5.1 不同滤波方法及窗口大小的对比

下面将选择实习数据中的一张影像为例，使用不同滤波方法、不同窗口大小，来对滤波结果进行比较和分析。

主观评价：主观评价标准，即通过人眼视觉效果进行定性评价，选择典型地物。

客观评价：从量化角度考虑滤波效果，能够辅助主观评价标准进行定量评价。

5.1.1 均值滤波



原图　　3X3 窗口　　5X5 窗口

主观评价：

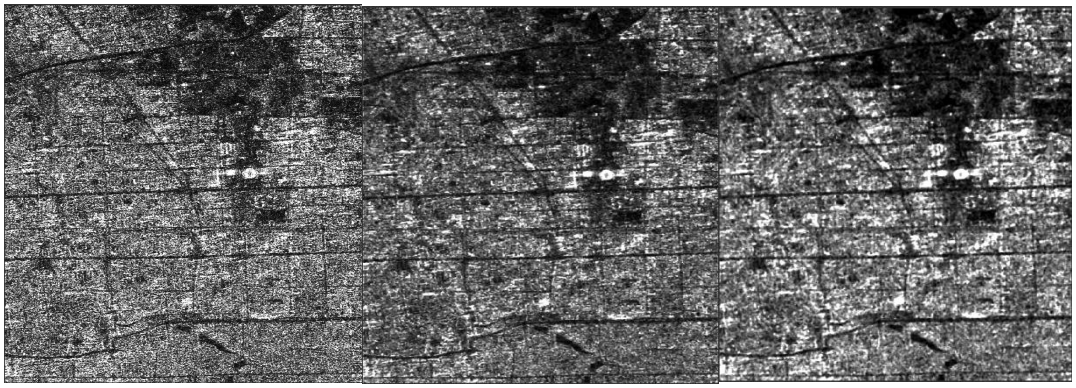
均值滤波	
建筑	均值滤波会平滑建筑物的边缘，导致细节信息丢失但能有效减少噪声。
道路	道路边缘可能会变得模糊，但整体路面的噪声会减少。
植被	植被区域的纹理可能会被平滑，细节丢失，但噪声得到抑制。
水体	水体通常在 SAR 影像中表现为较暗区域，均值滤波可能会导致周围较亮区域的噪声被平均进暗区域，降低影像对比度。
整体评价	•效果：均值滤波通过取周围像素的平均值来替换中心像素的值，从而减少噪声。 •优点：简单易行，能有效减少随机噪声，计算简单，速度快，均匀区域的斑点噪声去除效果较好。 •缺点：可能会模糊边缘，降低影像的清晰度，不适合保留边缘信息。细节保持得不好，图像边缘变模糊，点目标损失大，随着处理窗口的增大，图像的整体模糊和分辨率下降更严重。

客观评价：

	原图	均值 3×3	均值 5×5
均值	58.94	58.49	58.45
方差	3541.16	1292.66	857.98
等效视数	0.98	2.65	3.98
辐射分辨率	3.03	2.08	1.76

随着滤波窗口的变大，图像的均值无明显变化，但是方差和辐射分辨率明显减小，等效视数逐渐增大。

5.1.2 中值滤波



原图　　3X3 窗口　　5X5 窗口

主观评价：

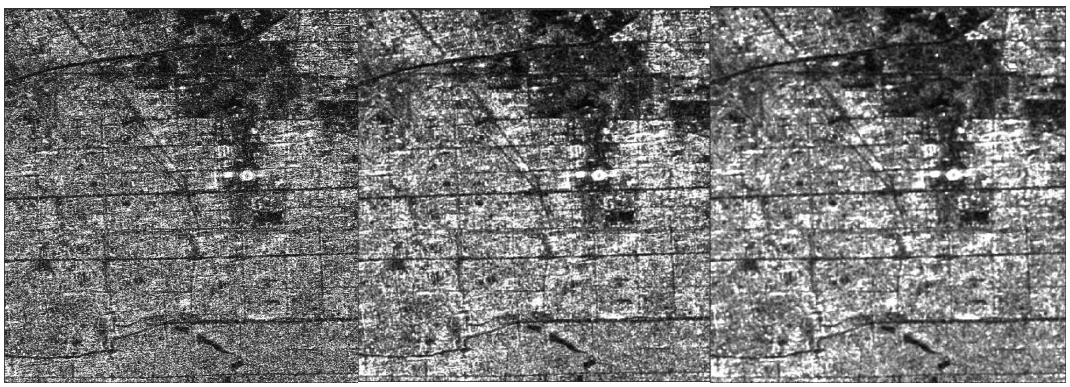
中值滤波	
建筑	中值滤波可以较好地保留建筑物的边缘信息，同时减少椒盐噪声。
道路	道路边缘信息得到保留，且能有效去除椒盐噪声。
植被	植被区域的纹理细节可能不如均值滤波清晰，但对去除椒盐噪声有效。
水体	水体区域的噪声得到抑制，但可能会引入一些不自然的纹理。
整体评价	•效果：与均值滤波效果基本相同。中值滤波通过替换像素值为其邻域内的中值来减少噪声，特别是椒盐噪声。均匀区域的斑点噪声去除效果较好，但细节保持得不好，图像边缘变模糊。 •优点：保留边缘信息，不会产生模糊。 •缺点：对高斯噪声的抑制效果不如均值滤波。与均值滤波相比，滤波效果会稍微更好一些，但边缘保持效果稍差，随机性更强。

客观评价：

	原图	中值 3×3	中值 5×5
均值	58.94	36.90	32.01
方差	3541.16	767.15	374.65
等效视数	0.98	1.78	2.73
辐射分辨率	3.03	2.43	2.05

随着滤波窗口的变大，图像的均值明显减小，且肉眼明显的感觉到图像变暗，但是方差和辐射分辨率明显减小，等效视数逐渐增大。

5.1.3 Sigma 滤波



原图　　3X3 窗口　　5X5 窗口

主观评价：

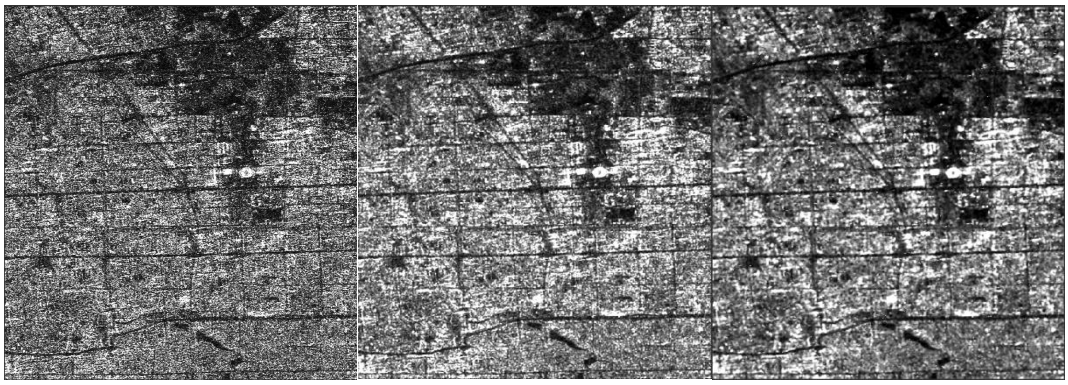
Sigma 滤波	
建筑	Sigma 滤波是基于高斯分布的，它会保留建筑物的边缘同时减少噪声。
道路	道路边缘得到较好的保留，噪声减少，但不如中值滤波那样直接有效。
植被	植被区域的纹理和细节得到较好的保留，噪声也得到有效控制。
水体	水体区域的噪声得到抑制，且能保持较好的影像对比度。
整体评价	•效果：Sigma 滤波是一种自适应滤波器，它根据邻域内的像素值分布来确定滤波窗口的大小。由于 Sigma 滤波只选择了窗口内像素灰度值与其中心像素的灰度值比较接近的像元做平均处理，因此其在滤波效果与均值滤波差不多的情况下，边缘和细节的保持有明显提升。 •优点：能够自适应地处理不同区域的噪声，保留边缘信息。 •缺点：算法复杂，计算量较大。

客观评价：

	原图	Sigma 3×3	Sigma 5×5
均值	58.94	53.31	50.89
方差	3541.16	666.26	427.53
等效视数	0.98	4.27	6.06
辐射分辨率	3.03	1.71	1.48

随着滤波窗口的变大，图像的均值逐渐减小，且方差和辐射分辨率明显减小，等效视数明显增大。

5.1.4 增强 Lee 滤波



原图　　3X3 窗口　　5X5 窗口

主观评价：

增强 Lee 滤波	
建筑	Lee 滤波专门针对 SAR 影像设计，能有效减少相干斑点噪声，同时保留建筑物边缘。
道路	道路边缘得到很好的保留，且滤波后的影像在道路区域较为平滑，噪声减少。
植被	植被区域的纹理细节得到较好的保留，相干斑点得到有效抑制。
水体	水体区域的噪声被有效抑制，且水体边缘与周围地物的界限清晰。
整体评价	•效果：改进 Lee 滤波方法由于区分了边缘、匀质区、点目标，因此在滤波效果不变的情况下，能较好地保持边缘和点目标。Lee 滤波方法对于在保持边缘等细节信息方面不是十分理想，但同质区则比较有效。 •优点：有效减少 SAR 影像中的相干斑点，保留边缘和细节信息。

	•缺点：滤波效果可能受到地形和地物特性的影响。

客观评价：

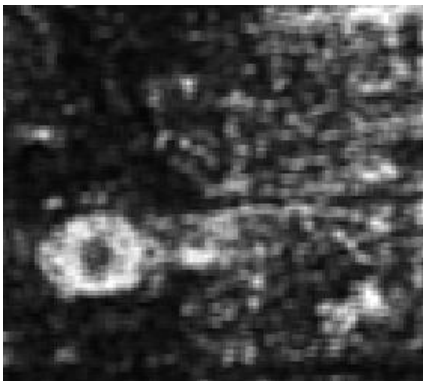
	原图	Lee 3×3	Lee 5×5
均值	58.94	57.59	57.95
方差	3541.16	1058.90	1084.89
等效视数	0.98	3.13	3.10
辐射分辨率	3.03	1.94	1.95

随着滤波窗口的变大，图像的均值变化不大，且方差和辐射分辨率明显减小，等效视数明显增大。

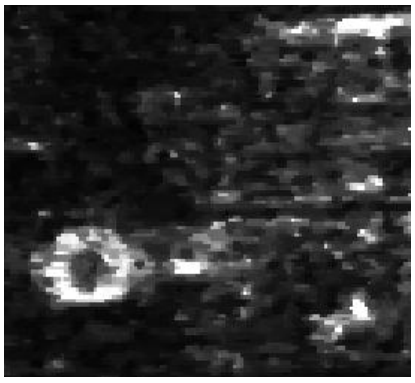
5.2 不同滤波结果综合对比

由以下效果可以看出，均值滤波和中值滤波滤波效果明显，中值滤波结果整体最暗，但边缘损失非常严重，图像边缘变模糊，点目标损失大，随着处理窗口的增大，图像的整体模糊和分辨率下降更严重。正是由于这两种传统滤波算法不适合相干斑噪声的乘性特点造成的。

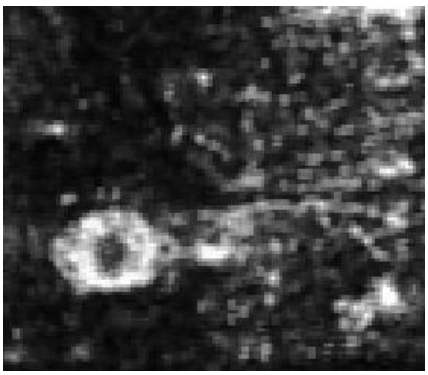
sigma 滤波因为只对窗口内部分像元进行平均，其在亮度上会变得更亮；其余的基于局域统计特性的自适应滤波算法中，个人觉得增强 Lee 滤波和 Lee 滤波在纹理区域效果较好。



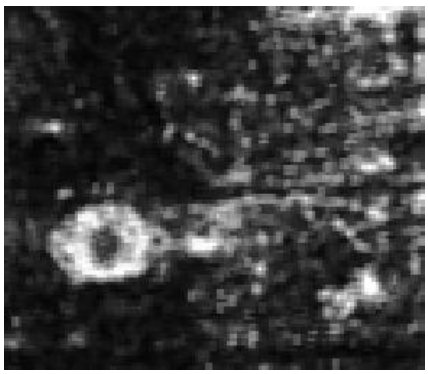
均值滤波（3*3）



中值滤波（3*3）



Sigma 滤波（3*3）

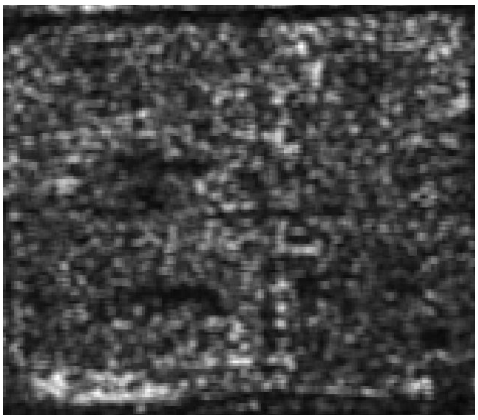


增强 Lee 滤波（3*3）

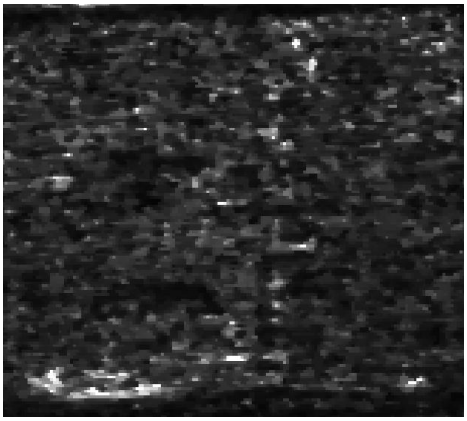
在建筑区域，均值滤波和中值滤波滤波效果比较明显，中值滤波细节保留效果最差亮度最暗，sigma 算法的区别也不是很大，Lee 滤波中均质区域存在保留了过多的原图像素的问题。由于城市地区建筑物边缘明显，Sigma 滤波和 Lee 滤波更适合保留这些边缘信息，而中值滤波可能会导致边缘模糊。水体通常较为平坦，均值滤波和 Lee 滤波可以减少噪声，但 Lee 滤波在处理 SAR 影像时更为有效。在这个过程中可以体会到算法不具备一成不变性，面对对于更多原像素的保留还是更多平均结果的保留，要根据对纹理区域和匀质区域滤波的需

求进行动态调整。

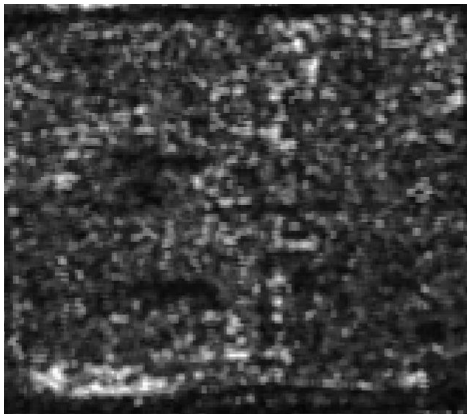
需要注意的是，每种滤波方法都有其适用场景和局限性，实际应用中需要根据具体的地物特性和影像质量要求来选择合适的滤波方法。此外，滤波处理的效果也受到滤波窗口大小、形状等参数的影响，因此在实际应用中需要进行参数优化以达到最佳效果。



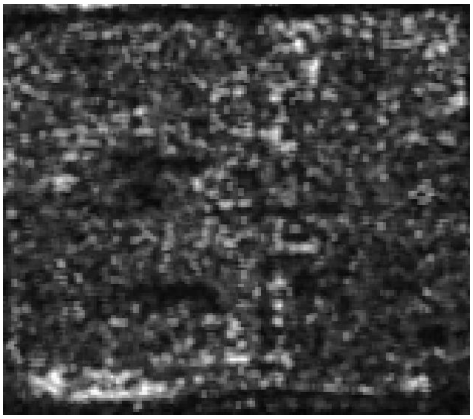
均值滤波（3*3）



中值滤波（3*3）



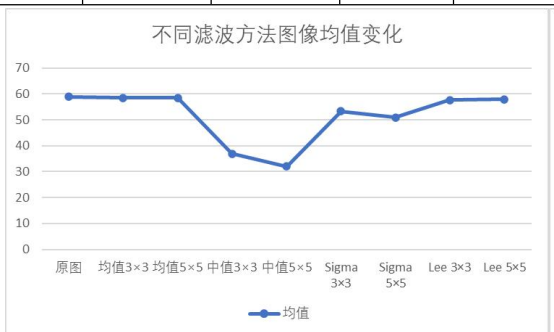
Sigma 滤波（3*3）



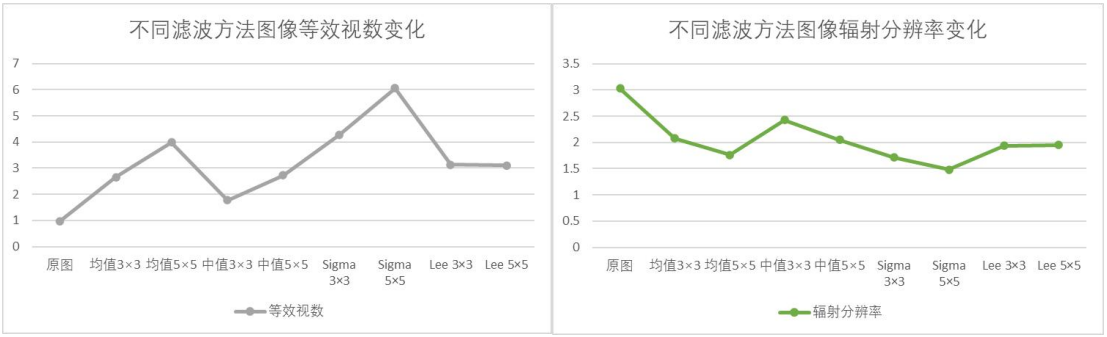
增强 Lee 滤波（3*3）

八种不同滤波方式和窗口大小的均值、方差、等效视数、辐射分辨率结果如下所示：

	原图	均值 3×3	均值 5×5	中值 3×3	中值 5×5	Sigma 3×3	Sigma 5×5	Lee 3×3	Lee 5×5
均值	58.94	58.49	58.45	36.90	32.01	53.31	50.89	57.59	57.95
方差	3541.16	1292.66	857.98	767.15	374.65	666.26	427.53	1058.90	1084.89
等效视数	0.98	2.65	3.98	1.78	2.73	4.27	6.06	3.13	3.10
辐射分辨率	3.03	2.08	1.76	2.43	2.05	1.71	1.48	1.94	1.95



可见，窗口越大、方差越小，这一点不管是什么滤波方法都是一样的。而且，所有经过滤波图像的方差都比原图像要小。最大方差是原图（3541.16），最小方差是中值滤波 5×5（374.65）。



可见，窗口越大、等效视数越大，这一点不管是什么滤波方法都是一样的。而且，所有经过滤波图像的等效视数都比原图像要大。最小等效视数是原图（0.98），最大等效视数是 sigma 滤波 5×5（6.06）。窗口越大、辐射分辨率越小（越小越好），这一点不管是什么滤波方法都是一样的。而且，所有经过滤波图像的辐射分辨率都比原图像要好。最大辐射分辨率是原图（3.03），最小辐射分辨率是 sigma 滤波 5×5（1.48）。

六、实习总结与心得

本次实习中，在编程时我或多或少的遇到了一些问题，总结如下。

一是配置 GDAL 环境时。在第一次实习中 GDAL 环境总是难以配置成功，在网上查阅许多资料和帖子，有用的寥寥无几，配置了两天才成功找到方法，GDAL 的版本要与 VS 编译器的版本配套。其次环境配置成功后出现程序可以编译但无法运行的情况，经过上网查找资料，并与老师交流，我发现是缺少动态链接库的原因，将相同版本 GDAL 的 dll 文件全部放在编程目录下，即可正常运行了。本次实习是第一次接触 GDAL 库，GDAL 是一个开源的地理空间数据转换库，支持多种格式的地理数据读写，包括 SAR 影像。GDAL 库的使用也不简单，通过阅读官方文档和一些教程，我逐步掌握了如何使用 GDAL 进行数据读取、处理和写入，经过查阅资料和咨询同学终于了解了如何使用 GDAL 库编辑 tiff 文件。

二是编写滤波算法。其次我发现某些滤波算法的结果明显有问题，有些参数意义也不明确，但经过仔细检查发现程序编写无错，均是按照所提供的技术指南上的公式进行的。因此，我们大胆猜测可能是所提供的公式有误。经过沟通和尝试后，发现确实如此，公式也得到了修改。

通过这次实习，我不仅提升了自己的编程能力，还对 SAR 影像处理有了更深入的了解。这次实习经历对我来说是非常宝贵的。它不仅增强了我的技术技能，还让我学会了如何在实际工作中应用这些技能。我期待将来能够将这段经历中学到的知识和技能应用到更多的项目中，为 SAR 遥感影像处理领域做出自己的贡献。