数字图像处理复习题

理想低通滤波器产生振铃现象的原因

其在频率域的尖锐过渡特性和对应的空间域滤波函数的sinc函数特性，以及与吉布斯现象相关的边缘效应

直方图均衡和同态滤波的异同

同：两者都旨在提高图像的对比度，使图像更加清晰

异：

- 直方图均衡是空域，同态滤波是频域
- 直方图均衡通过调整图像的直方图分布，使其尽可能均匀，从而增加图像的全局对比度。直方图均衡主要关注图像的整体对比度，而不考虑局部区域的对比度变化
- 同态滤波基于图像的照度和反射率分离，通过在对数域中处理图像，可以同时增加对比度和标准化亮度。同态滤波可以增强图像的局部对比度，尤其是在阴影区域，并且可以处理乘性噪声

局部增强相比于全局增强可以减弱阴阳脸的原因

局部增强相比于全局增强可以减弱阴阳脸的原因主要在于它们处理图像的方式不同。全局增强通常对整幅图像应用相同的增强参数，不考虑图像中不同区域的局部特性，这可能导致某些区域过度增强或不足，特别是在光照不均匀的情况下，比如阴阳脸现象，即人脸的一半明亮而另一半较暗。这种不均匀的光照条件会导致全局增强方法无法有效地处理图像中的局部细节，从而加剧阴阳脸的效果。

相比之下，局部增强方法会考虑图像中每个小区域的特性，对不同的区域应用不同的增强参数。这种方法能够更好地适应图像中的局部变化，特别是在处理阴阳脸这类光照不均匀的情况时。局部增强算法通常会计算图像中每个小区域的统计特性，如局部均值、方差等，然后根据这些统计特性来调整增强系数。例如，如果一个区域的局部方差较大，表明该区域可能是图像的高频部分，局部增强算法可能会降低该区域的增强系数，以避免过度增强导致的失真；反之，如果局部方差较小，表明该区域可能是图像的低频部分，增强系数会相对较大，以提升细节的可见度

直方图均衡化和规定化的过程，异同

直方图均衡化的过程：

1. **确定图像的灰度级**：将彩色图像转换为灰度图像，并确定灰度级范围。
2. **计算原始直方图的概率**：统计每个灰度级在原始图像中的像素所占比例。

3. **计算直方图概率的累加值**：计算累积分布函数（CDF），直到最后一个灰度级，总和为1。
4. **应用变换函数**：利用累积分布函数作为变换函数，将原始图像的灰度值映射到新的灰度值，使得新图像的直方图均匀分布。

直方图规定化的过程：

1. **确定原始图像和目标图像的直方图**：需要知道原始图像的直方图和希望达到的目标直方图。
2. **均衡化处理**：对原始图像和目标图像的直方图都进行均衡化处理，使它们变成相同的归一化的均匀直方图。
3. **逆变换**：对目标图像进行均衡化的逆运算，以得到一个变换函数，该函数将原始图像的直方图变换为目标直方图的形状。

异同点：

- **相同点**：两者都是通过改变图像的直方图分布来增强图像的对比度，直方图均衡化是直方图规定化的一个特例。
- **不同点**：
  - **目的**：直方图均衡化旨在自动增强整个图像的对比度，得到全局均衡化的直方图；而直方图规定化则更灵活，可以变换直方图，使之成为某个特定的形状，有选择地增强某个灰度值范围内的对比度。
  - **控制性**：直方图均衡化的具体效果不易控制，而直方图规定化可以更精确地控制最终的直方图形状，因此可以有目的地增强图像的特定部分。
  - **应用**：直方图均衡化通常用于自动增强图像对比度，而直方图规定化则用于将一幅图像的直方图匹配到另一幅图像的直方图，或者匹配到一个特定的直方图形状。

均值滤波和中值滤波的模板

以下是几个典型的加权均值滤波器：

$$H_1 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad H_2 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad H_3 = \frac{1}{2}\begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

中值滤波：没有特点模板？？只是找中值来替代

不同噪声图像对滤波器选择的方式和原因

椒盐噪声—中值滤波

- 椒盐噪声只在画面中的部分点上随机出现，所以根据中值滤波原理可知，通过数据排序的方法，将图像中未被噪声污染的点替代噪声点的值的概率比较大，因此中值滤波对椒盐噪声的抑制效果很好，同时画面的清晰度基本保持。
- 因为椒盐噪声的均值不为0，所以均值滤波不能很好地去除椒盐噪声点。

高斯噪声—均值滤波

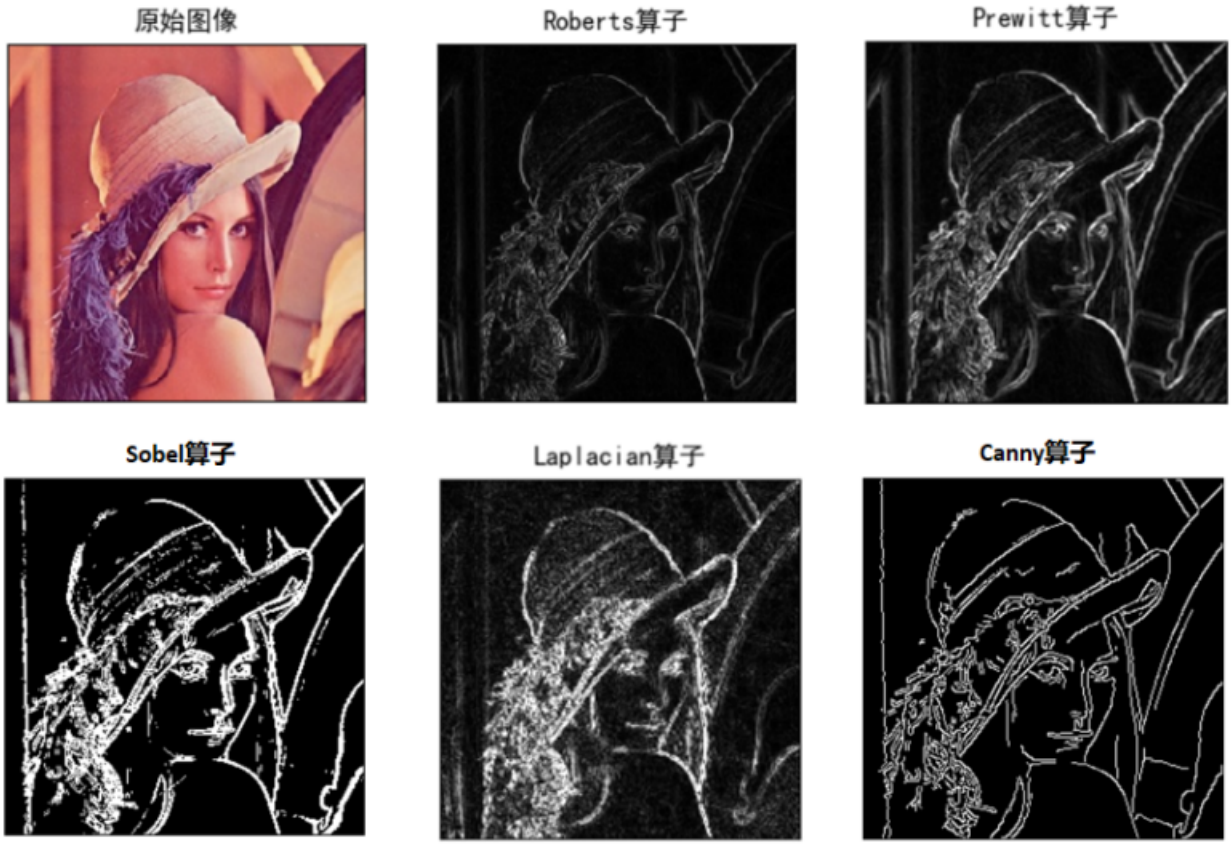
- 高斯噪声是以随机大小的幅值污染所有的点，因此无论怎样进行数据选择，得到的总是被污染的值，所以中值滤波对高斯噪声的抑制效果不是很好；
- 因为正态分布的均值为0，所以均值滤波可以消除高斯噪声。

roberts算子和sobel算子的特性

- $Roberts$ 算子: $d_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \ d_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
- $Prewitt$ 算子: $G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \ G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
- $Sobel$ 算子: $G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \ G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$
- $Laplacian$ 算子：4 领域: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$; 8 邻域: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
- $Canny$ 算子：实现步骤：1. 用高斯滤波器平滑图像
  2. 计算图像中每个像素点的梯度强度和方向
  3. 对梯度幅值进行非极大值抑制
  4. 用双阈值算法检测来确定真实和潜在的边缘

常见边缘检测对比（Roberts算子、Prewitt算子、Sobel算子、Laplacian算子、Canny算子）_利用sobel, prewitt和拉普拉斯算子,设置合理的阈值,提取边缘,比较结果,给出对比结-CSDN博客

边缘检测结果对比



低通滤波和高通滤波截止频率的影响

低通滤波器截止频率的影响：

1. **信号通过与衰减**：低通滤波器允许低于截止频率的信号通过，而高于截止频率的信号则会被大幅衰减。
2. **信号保持**：在输入频率小于截止频率时，低通滤波器能够保持信号的幅值和相位，但大于截止频率后，信号的幅值会大幅衰减，相位也会严重滞后。
3. **噪声滤除**：低通滤波器可以有效滤除高频噪声，但截止频率的设置需要权衡，太低会导致幅值衰减，太高则噪声未被滤除干净。

高通滤波器截止频率的影响：

1. **信号通过与衰减**：高通滤波器允许高于截止频率的信号通过，而低于截止频率的信号会被大幅衰减。
2. **细节增强**：高通滤波器常用于增强图像或信号中的细节，因为它可以去除低频成分，如背景或平均值。

异同点：

- **相同点**：两者都是通过截止频率来定义通带和阻带的界限，即信号得以通过的频率范围和被衰减的频率范围。
- **不同点**：
  - **频率响应**：低通滤波器对低频信号无衰减，而高通滤波器对高频信号无衰减。
  - **应用场景**：低通滤波器常用于去除噪声或平滑信号，而高通滤波器用于突出信号中的变化或细节。
  - **截止频率的影响**：低通滤波器的截止频率设置需要考虑信号的保真度和噪声的滤除效果，而高通滤波器的截止频率设置则需要考虑信号中需要保留的细节程度。

解释canny算子和log算子的异同和优缺点

Canny边缘检测步骤：

1. **噪声降低**：
   - 使用高斯滤波器对图像进行平滑处理，以减少图像中的噪声，因为噪声可能会产生错误的边缘响应。
2. **计算梯度幅度和方向**：
   - 使用 Sobel 算子计算图像的梯度幅度和方向。这涉及到对图像进行水平和垂直方向的差分，然后计算每个像素点的梯度幅度和方向。
3. **非极大值抑制**：
   - 在梯度方向上，对非边缘像素进行抑制，只保留局部梯度最大的像素点，以确保边缘的细薄性。
4. **双阈值检测**：
   - 选择两个阈值（高阈值和低阈值），用于确定强边缘、弱边缘和非边缘。强边缘是那些梯度值高于高阈值的边缘，弱边缘是那些梯度值在两个阈值之间的边缘。
5. **边缘跟踪**：
   - 通过滞后阈值技术（hysteresis thresholding），将强边缘和与之相连的弱边缘连接起来，形成完整的边缘。
6. **边缘细化**：
   - 对检测到的边缘进行细化，以得到更精确的边缘位置。
7. **结果输出**：
   - 输出最终的边缘图像，其中边缘以白色或黑色线条显示，非边缘区域为黑色。

Log边缘检测步骤：

1. **高斯平滑**：
   - 与Canny算子类似，首先使用高斯滤波器对图像进行平滑处理，以减少噪声的影响。
2. **计算拉普拉斯算子**：
   - 使用拉普拉斯算子（Laplacian of Gaussian, LoG）来计算图像的二阶导数。LoG算子是高斯函数和拉普拉斯算子的卷积，它可以用来检测图像中的边缘。
3. **零交叉检测**：
   - 寻找LoG算子响应的零交叉点，这些点通常对应于边缘的位置。
4. **细化边缘位置**：
   - 对于每个零交叉点，计算其邻域内的LoG响应，以确定边缘的确切位置。
5. **输出边缘图像**：
   - 输出最终的边缘图像，其中边缘以白色或黑色线条显示，非边缘区域为黑色。

Log和Canny边缘检测能力对比（附Matlab程序）_log算法和canny算法各自的特点和优劣-CSDN博客

解释分水岭算法的流程，以及和otsu算法分割出的图像的区别

分水岭算法流程：

分水岭算法是一种基于图像拓扑结构的分割方法，其基本思想是将图像视为一个地形图，其中每个像素点的灰度值代表该点的海拔高度。算法的流程大致如下：

1. **预处理**：对输入图像进行梯度计算，得到梯度图像。梯度图像中的每个像素值表示该点的边缘强度，即地形的坡度。
2. **标记过程**：对梯度图像中的所有像素按照灰度值进行分类，并设定一个测地距离阈值。
3. **寻找起点**：找到灰度值最小的像素点（通常是边缘强度最低的点），这些点作为分水岭变换的起始点。
4. **淹没过程**：模拟水从这些最低点开始淹没的过程。随着"水位"的上升，会碰到周围的邻域像素，测量这些像素到起始点的测地距离，如果小于设定阈值，则将这些像素淹没，否则在这些像素上设置"大坝"。

5. **构建分水岭**：随着水位的不断上升，会设置更多更高的大坝，直到灰度值的最大值，所有区域都在分水岭线上相遇，这些大坝就对整个图像像素进行了分区。

6. **输出结果**：最终，分水岭算法得到的是输入图像的集水盆图像，集水盆之间的边界点即为分水岭，表示图像的极大值点，通常用于表示图像的边缘信息。

分水岭算法与Otsu算法的区别：

1. **原理不同**：
   - **Otsu算法**：是一种自动阈值选择算法，通过遍历所有可能的阈值，找到使得前景和背景之间的类间方差最大的阈值，实现图像的二值化分割。
   - **分水岭算法**：基于图像的拓扑结构，通过模拟水淹没的过程来识别和分割图像中的不同区域或对象。

2. **应用场景不同**：
   - **Otsu算法**：适用于背景和前景对比度较高，且分布比较明显的图像，常用于简单的二值化处理。
   - **分水岭算法**：适用于图像中对象和背景灰度值相近，但对象之间有明显边界的情况，常用于更复杂的图像分割任务。

3. **结果不同**：
   - **Otsu算法**：输出的是二值化图像，将图像分为前景和背景两部分。
   - **分水岭算法**：输出的是多个分割区域，可以识别出图像中的多个对象或特征。

4. **参数敏感性不同**：
   - **Otsu算法**：对参数不敏感，只需要选择合适的阈值即可。
   - **分水岭算法**：对预处理和阈值选择较为敏感，需要仔细调整以获得最佳分割效果。

- **Otsu算法**：适用于简单的二值化分割，对全局对比度变化敏感，对噪声有一定的敏感性，但对复杂图像的分割能力有限。
- **分水岭算法**：适用于多对象的复杂图像分割，能够保持边缘信息，但对噪声非常敏感，且可能需要后处理来优化分割结果。

ncuts算法需要的存储空间大小是多少，它的流程是什么

Ncuts算法流程：

1. **构建图模型**：将图像的像素点视为图的顶点，根据像素点之间的相似性（如颜色、纹理等）构建权重矩阵。

2. **计算拉普拉斯矩阵**：基于权重矩阵和度矩阵计算图的拉普拉斯矩阵 L=D−W，其中 D 是度矩阵，W 是邻接矩阵。

3. **求解特征值和特征向量**：计算拉普拉斯矩阵$D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$的特征值和特征向量。

4. **聚类分割**：根据特征向量将像素点聚类成 k 类，(用第二小的特征值（最小的非零特征值）及特征向量，对图进行划分（分成两个子图）)完成图像的分割。

Ncuts算法理由：

- **最小化割代价**：Ncuts算法通过最小化图的割代价（即两个子集之间的边权重之和）来实现图像分割，这有助于保持图像中相似区域的连通性。
- **保持区域一致性**：通过最小化割代价，Ncuts算法能够较好地保持图像中相似区域的一致性，避免将同一区域内的像素点分割到不同的区域。

Ncuts算法所需的存储空间大小：

Ncuts算法的存储空间主要取决于图的邻接矩阵和拉普拉斯矩阵的大小。对于一个包含 N 个像素点的图像，邻接矩阵和拉普拉斯矩阵都是 N×N 的矩阵。因此，所需的存储空间大约是 O(N2)，即与图像中像素点的数量的平方成正比。

Mean Shift 算法的优缺点

优点：

1. **不需要预设聚类数**：Mean Shift算法不需要事先指定聚类的数量，这使得它在聚类数量未知或难以确定的情况下非常有用。

2. **处理非线性数据**：与K-Means算法不同，Mean Shift能够处理非线性分离的数据。

3. **单一参数**：Mean Shift算法只需要一个参数——带宽大小（bandwidth size），这个参数具有物理意义，与K-Means中的K值不同。

4. **发现变量聚类中心数**：算法能够发现数据中的变量聚类中心数。

5. **对异常值鲁棒**：Mean Shift算法对异常值和噪声具有较好的鲁棒性。

缺点：

1. **依赖带宽大小**：算法的输出依赖于带宽大小，选择合适的带宽大小可能并不简单。

2. **计算成本高**：Mean Shift算法相对于其他聚类算法来说计算成本较高，因为它需要多次迭代才能收敛到模式。

3. **维度灾难**：算法对特征空间的维度不友好，随着维度的增加，计算复杂度会显著增加。

4. **收敛速度慢**：对于大型数据集，Mean Shift算法相对较慢，并且对高维数据的扩展性较差，计算时间通常是O(n log n)到O(n^2)。
5. **过度分割和欠分割风险**：较小的带宽可能导致过度分割，而较大的带宽可能导致欠分割。
6.

开闭运算的区别，在去噪时怎么选择

闭运算具有磨光物体内边界的作用，而开运算具有磨光图像外边界的作用。

开闭运算的区别：

1. **目的不同**：
   - 开运算主要用于去除小的白噪声和分离相邻物体。
   - 闭运算主要用于填充小的黑洞和断裂，以及去除小的黑噪声。
2. **效果不同**：
   - 开运算后，图像中的亮区域会减少，因为腐蚀操作会缩小亮区域。
   - 闭运算后，图像中的亮区域会增加，因为膨胀操作会扩大亮区域。

在去噪时如何选择：

1. **白噪声**：如果图像中的噪声主要是白噪声，即小的亮点，应选择开运算。
2. **黑噪声**：如果图像中的噪声主要是黑噪声，即小的暗点，应选择闭运算。
3. **物体连接**：如果需要保持物体的连接性，开运算可以断开细小的连接，而闭运算可以连接相邻的物体。
4. **平滑边界**：开闭运算都可以平滑物体的边界，但开运算更适用于平滑较小物体的边界，而闭运算更适用于平滑较大物体的边界。

对含噪图像采用开运算还是闭运算，为什么?

在处理含噪图像时，选择开运算还是闭运算取决于噪声的类型和处理目标。以下是两种运算的特点和适用场景：

1. **开运算（Opening）**：
   - 开运算是先进行腐蚀操作，再进行膨胀操作。它主要用于去除图像中的噪声、断开连接的细小物体、以及平滑对象的边缘等操作。

- 开运算特别适合去除白噪声，即图像中的亮斑噪声。这是因为腐蚀操作会消除小的亮斑，随后的膨胀操作则不会使这些被消除的亮斑恢复。
- 开运算还可以用来分离物体，消除细小的连接，以及平滑较大物体的边界。

2. **闭运算（Closing）**：

   - 闭运算是先进行膨胀操作，再进行腐蚀操作。它主要用于填补图像中的小孔洞、连接断开的物体、以及平滑物体的边缘等操作。
   - 闭运算特别适合去除黑噪声，即图像中的暗斑噪声。膨胀操作会连接邻近的物体并填补小的空洞，随后的腐蚀操作则不会使这些新连接的物体断开。
   - 闭运算还有助于关闭前景物体内部的小孔，或物体上的小黑点，使物体更加完整。

总结来说，如果含噪图像中的噪声主要是亮斑（白噪声），则应选择开运算；如果噪声主要是暗斑（黑噪声），则应选择闭运算。此外，开运算和闭运算也可以结合使用，先进行开运算去除噪声，再进行闭运算填补孔洞和连接断开的物体，以达到更好的图像处理效果。

SIFT特征提取流程

SIFT（尺度不变特征变换）特征提取流程主要包括以下几个步骤：

1. **尺度空间极值检测**：通过构建高斯差分金字塔（DoG金字塔），在不同尺度上检测潜在的关键点。这一步骤利用高斯微分函数来识别对于尺度和旋转不变的兴趣点，确保在图像发生尺度变化时，相同的特征点能在不同尺度空间中被检测到，实现尺度不变性。

2. **关键点定位**：对检测到的候选关键点进行精确定位，消除低对比度的特征点和不稳定的边缘响应点，确保剩下的关键点是稳定且可靠的。

3. **方向赋值**：为每个关键点分配一个或多个方向，使得特征描述符对旋转具有不变性。这是通过计算关键点邻域内像素的梯度方向和大小来实现的，确保旋转对特征描述子的影响最小化。

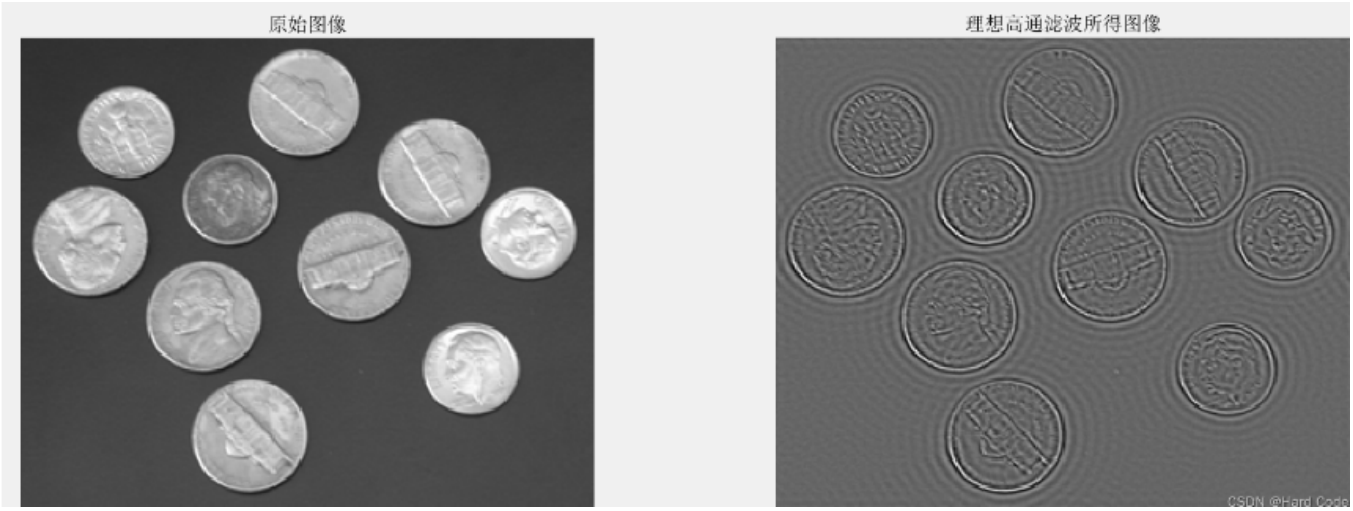
4. **关键点描述符生成**：在关键点周围的邻域内生成描述符，这些描述符对尺度和旋转具有不变性。描述符通常由关键点邻域内的像素的梯度方向和大小构成，形成特征向量。

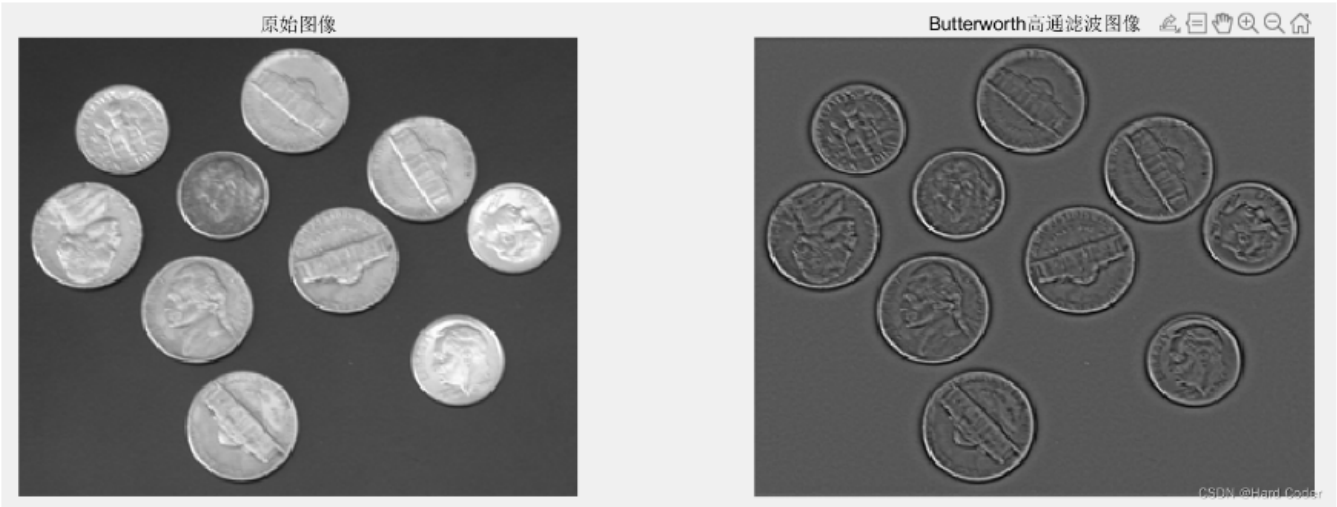
为什么SIFT能够保持尺度不变性和旋转不变性

1. **尺度不变性**：SIFT通过在不同尺度空间上查找关键点来实现尺度不变性。关键点的检测是基于高斯差分金字塔，这意味着无论图像如何缩放，相同的特征点都能在相应的尺度空间中被检测到，从而保证了尺度不变性。

2. **旋转不变性**：SIFT为每个关键点分配一个主方向，这个方向是根据关键点邻域内像素的梯度方向确定的。在生成特征描述符时，这些方向信息被用来确保描述符与关键点的方向一致，这样即使图像发生旋转，描述符仍然能够匹配到相应的特征点，实现旋转不变性。

理想高通滤波器和巴特沃斯高通滤波器滤波图像的分辨和理由





有振铃现象