1.1

假设原图像直方图的灰度 r 经过第一次操作变为 s 再由第二次操作结果变为 t。那么我们由直方图均衡化的公式

$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} P_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$，以及

$t_k = T(s_k) = (L-1)\sum_{j=0}^{k} P_s(s_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$。

式中$P_r(r_j)$表示原始灰度值r_j对应出现的概率，$P_s(s_j)$表示直方图均衡化后灰度值s_j对应出现的概率，k=0,1,2,3,…,L-1。均衡化后对应的灰度值，概率大小不变。直方图均衡化均衡结果只与灰度值出现的概率大小有关，而与灰度值的大小无关。根据均衡化变换函数一个重要性质：单值且单调递增，即说明经过灰度变换后，相对大小不变，较大的灰度仍对应较大的灰度，其概率也不发生任何变化。所以有，$t_k = s_k$，这就说明,第二次直方图均匀化的结果与第一次直方图均匀化的处理结果是相同的。

1.2

1.

177	420	271	263
75	218	249	107
-131	-324	-107	-133
-173	-336	-362	-207

2.

正数表示正常灰度值，大于 255 的灰度可以直接取 255；负数则可以取绝对值，这都是离散的点，不会对图像有大的影响。

3.

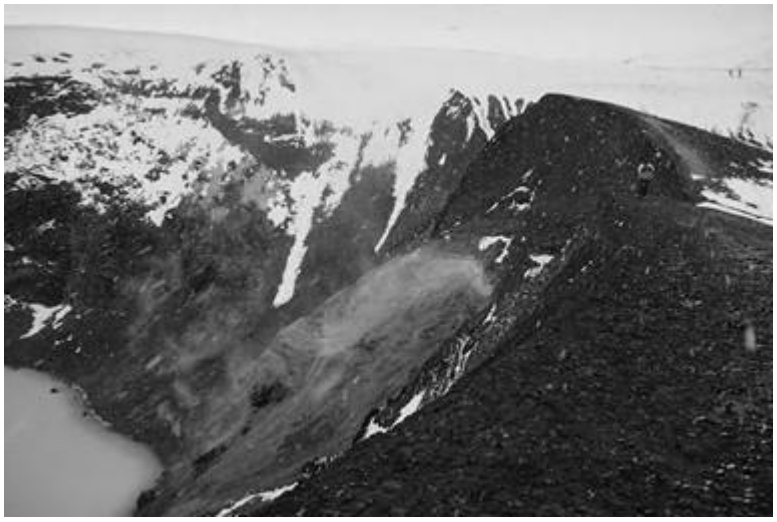
用于边缘检测，工业检测，不是辅助人工检测产品缺陷，就是更为通用地作为自动检测的预处理。突出灰度图像中看不见的小斑点。在灰度平坦区域中增强小突变能力。

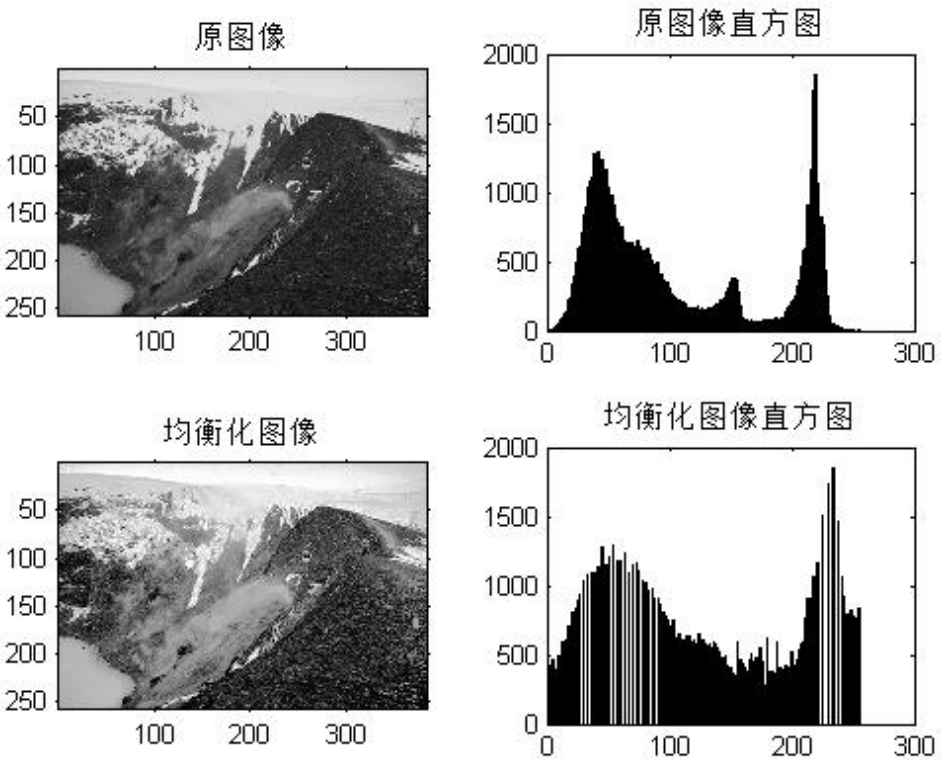
2.2

1.

2.

原图





3.

经过直方图均衡化处理后，原始图像的灰度直方图从比较集中的某些灰度区间变成在全部灰度范围内的均匀分布。就是说对图像进行非线性拉伸，重新分配图像像素值，使一定灰度范围内的像素数量大致相同。直方图均衡化就是把给定图像的直方图分布改变成“均匀”分布直方图分布。通过前后两张直方图对比，明显可以看出均衡后的直方图像素分布更加均匀，增加了像素灰度值的动态范围，达到了增强图像整体对比度的效果。不过，从图中也可以看

出一些缺点：比如均衡化后增加了背景杂讯的对比度并且降低了有用信号的对比度；还有，我的这幅直方图上有高峰，经处理后对比度不自然的过分增强。

4.

算法分析：

根据书本上的直方图均衡化公式

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} P_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j,$$

一步步计算出各个变量即可。简单来说就是将原图像任意（x，y）处的灰度值 r 通过公式所述函数映射到均衡化图像相应位置上的灰度值 s。

第一步：

创建长度为 256 的数组，统计原图像各个灰度值的数目；

```
NumPixel = zeros(1,256);%用长度为 256 的一维数组统计各灰度值的数目
for i = 1:height
    for j = 1: width
        NumPixel(img(i,j) + 1) = NumPixel(img(i,j) + 1) + 1;
    end
end
```

第二步：

计算各个灰度值 PDA；

```
PDA = zeros(1,256);
for i = 1:256
    PDA(i) = NumPixel(i) / (height * width * 1.0);
end
```

第三步：

计算累积分布函数 CDA，并且乘上 255 取整。

```
CDA = zeros(1,256);
for i = 1:256
    if i == 1
        CDA(i) = PDA(i);
    else
        CDA(i) = CDA(i - 1) + PDA(i);
    end
end
```

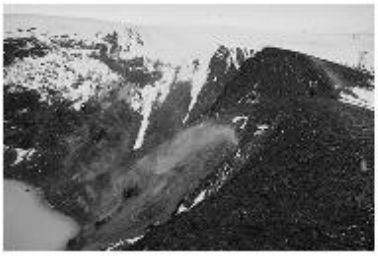
第四步：

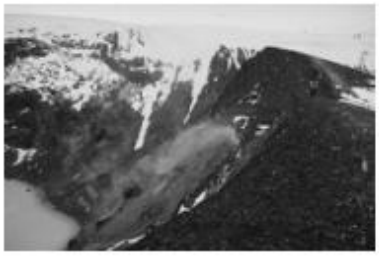
直接将原图各个位置像素灰度值映射到均衡化图像上即可。

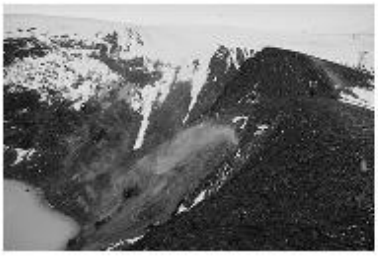
```
for i = 1:height
    for j = 1: width
        new(i,j) = CDA(img(i,j)+1);
    end
end
```

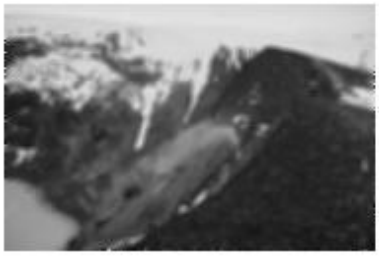
2.3

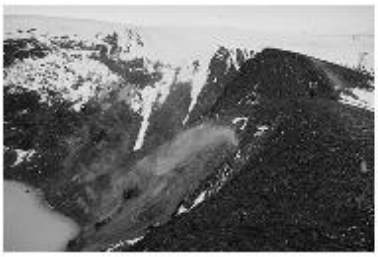
1.左边为原图，右边为平滑后图像



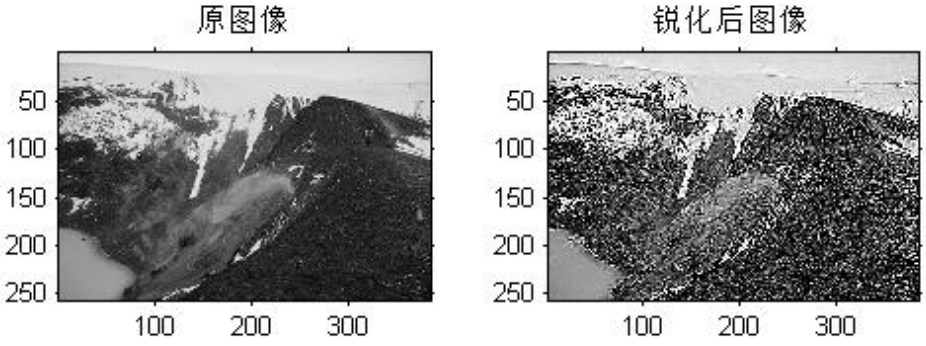








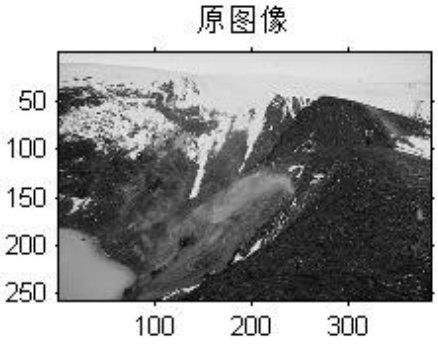
2.

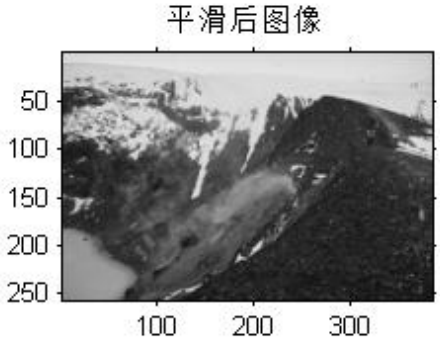


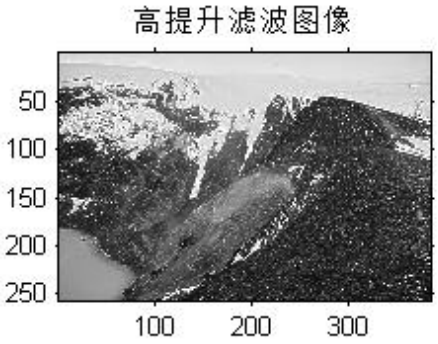
我选择的是第一种滤波器，【0 1 0；1 -4 1；0 1 0】。

图像锐化处理的作用是使灰度反差增强，从而使模糊图像变得更加清晰。积分，平均运算使图像模糊，相反微分运算能够突出图像细节，使图像变得更为清晰。由于拉普拉斯是一种微分算子，它的应用可增强图像中灰度突变的区域，减弱灰度的缓慢变化区域。因此，锐化处理可选择拉普拉斯算子对原图像进行处理，产生描述灰度突变的图像，再将拉普拉斯图像与原始图像叠加而产生锐化图像。简单来说，从模板形式就可以容易看出，如果在图像中一个较暗的区域中出现了一个亮点，那么用拉普拉斯运算就会使这个亮点变得更亮。因为图像中的边缘就是那些灰度发生跳变的区域，所以拉普拉斯锐化模板在边缘检测中很有用。

3.







我选择的 K 值为 3.

```
%下面这部分是高提升滤波处理；
mask = img - new;
highboost = img + 3 * mask;%k 值设为 3
```

4.

算法说明：

算法很简单，就是将图像矩阵与输入的滤波器模板矩阵进行相关运算即可。就是依序将模板矩阵与原图像相同大小矩阵位置相乘并求和，将得到的和赋值给模板中心位置像素，然后平移矩阵，直至便利整个图像矩阵。对于边界像素，我是取图像原值直接赋值给新图像。

```
for i=1:IH-n+1
    for j=1:IW-n+1
        c=temp1(i:i+(n-1),j:j+(n-1)).*a;  %进行相关运算，取出 temp1 中从(i,j)开始的 n 行 n 列元素与模板矩阵 a 相乘
        s=sum(c(:)); %求模板 c 矩阵中各元素之和
        temp2(i+(n-1)/2,j+(n-1)/2)=s; %将模板各元素的均值赋给模板中心位置的元素
    end
end
%边界处理，没有被赋值的边界元素全部取原值
new=uint8(temp2);
```

对于高提升滤波处理，也是按照书本所述公式即可。先模糊原图像，接着从原图

像中减去模糊图像得到模板，最后将模板加到原图像上即可。当中权值 K 应当取大于 1。我取的是 3。

```
%下面这部分是高提升滤波处理；
mask = img - new;
highboost = img + 3 * mask;%k 值设为 3
```