图像增强

图像增强是一种常用的图像处理方法，是指将原来不清晰的图像变得清晰或强调某些感兴趣的特征，抑制不感兴趣的特征，使之改善图像质量、丰富信息量，加强图像判读和识别效果。图像增强常用方法有灰度值变换和直方图增强，该方法主要调整灰度图像的明暗对比度，是对图像中各个像素的灰度值直接进行处理；空间域滤波也是常用的图像增强方法，常用方法包括均值滤波、中值滤波、高斯滤波等各种平滑方法；除了空间域滤波，还有频域滤波。频率域图像增强首先通过傅里叶变换将图像从空间域转换为频率域，然后在频率域内对图像进行处理，最后通过傅里叶反变换转换到空间域。频率域内的图像增强通常包括低通滤波、高通滤波和同态滤波等。

本次大恒课堂结合 HALCON，主要介绍三种图像增强的方法：灰度值变换、图像平滑和傅里叶变换，希望大家能够有所收获。

1.1 灰度值变换

图像灰度值变换主要为了提高图像的对比度。灰度值变换是一种点操作，即变换后的灰度值仅仅依赖于输入图像上同一位置的原始灰度值。

1.1.1 对比度增强

最常见的灰度值变换是灰度值线性变换，即将图像的像素点的灰度值按照线性变换函数进行变换从而改变对比度：g(x,y)= f(x,y) * k + b。在 HALCON 中常用的灰度值变换算子有 scale_image、add_image、sub_image 和 invert_image 等。

scale_image(Image : ImageScaled : Mult, Add :)按照如下变换，缩放图像的灰度值（运算过程中的上溢出或者下溢出都会被剔除）：

$$g' := g * Mult + Add$$

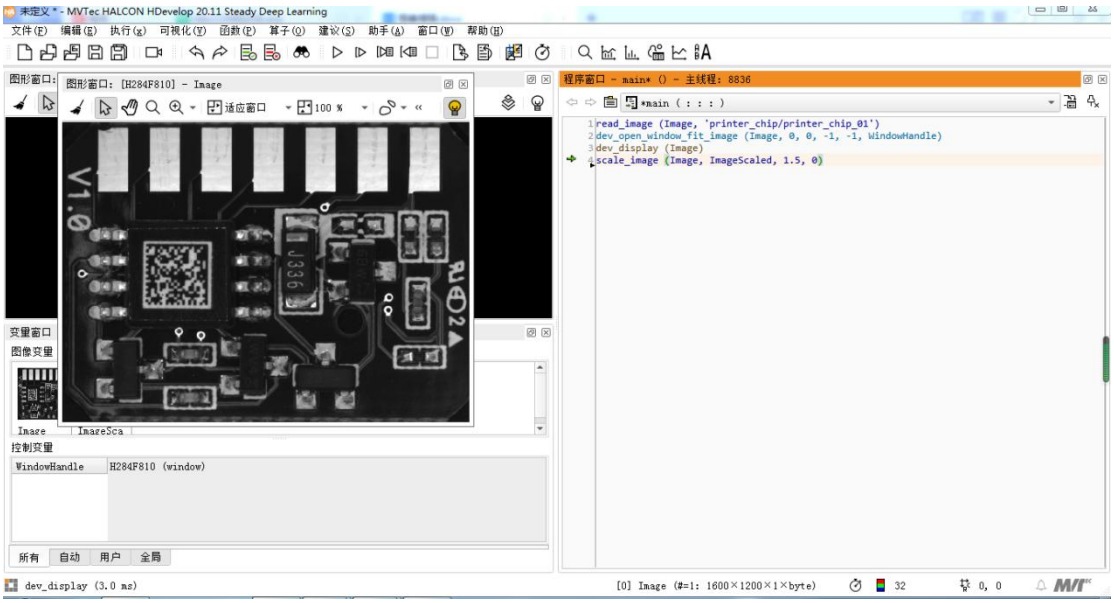
其中参数 Mult 和 Add 的最佳值选取如下：

$$Mult = 255/GMax-GMin \qquad Add = -Mult * GMin$$

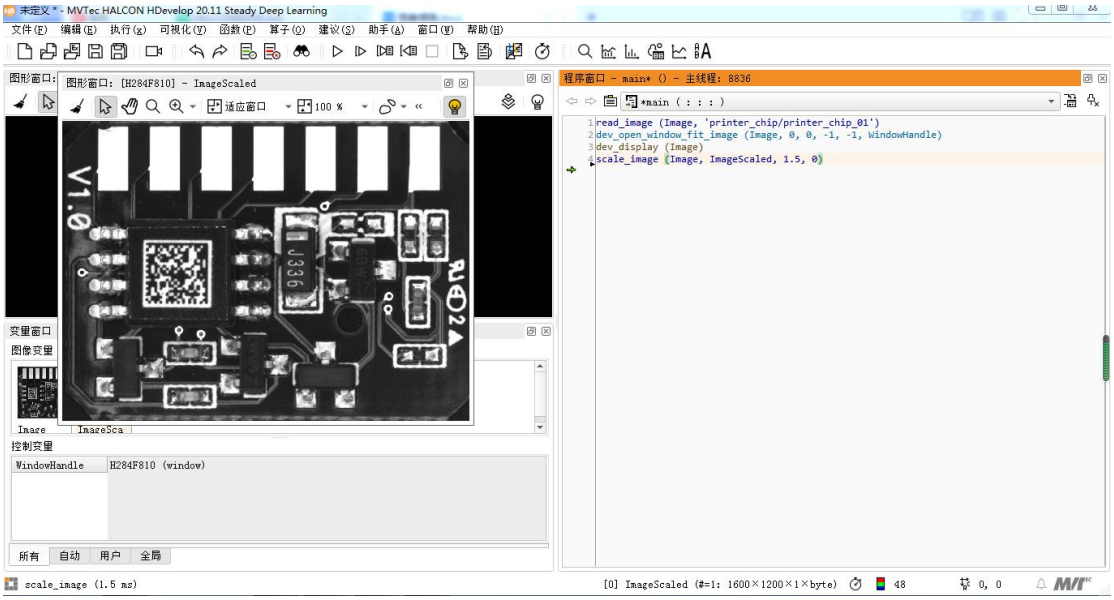
GMax 和 GMin 的值可由算子 min_max_gray 决定。

下面是 HDevelop 处理程序：

对比度增强前：



对比度增强后（Mult = 1.5，Add = 0）：



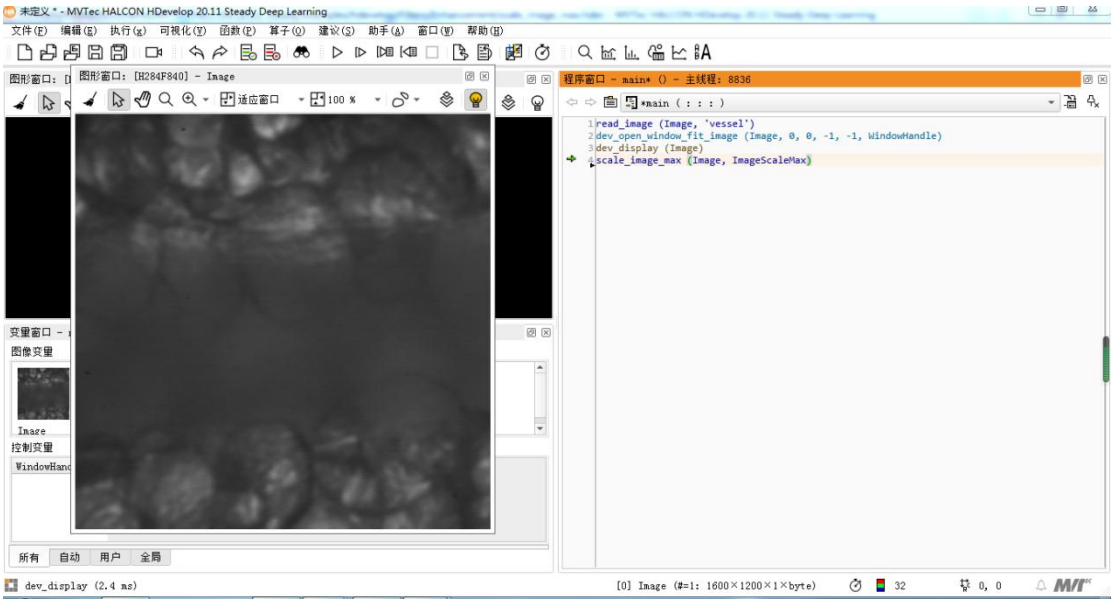
1.1.2 对比度归一化

上面的算子需要手动指定 Mult 和 Add 参数，而对比度归一化方法可以基于当前图像情况自动确定这两个参数。对比度归一化方法让变换后的图像灰度值覆盖最大取值范围[0,255]，这样就可以充分利用动态值范围。不同灰度的数量没有变化，但总体上视觉效果增强。HALCON 中实现该操作的算子是“scale_image_max”。

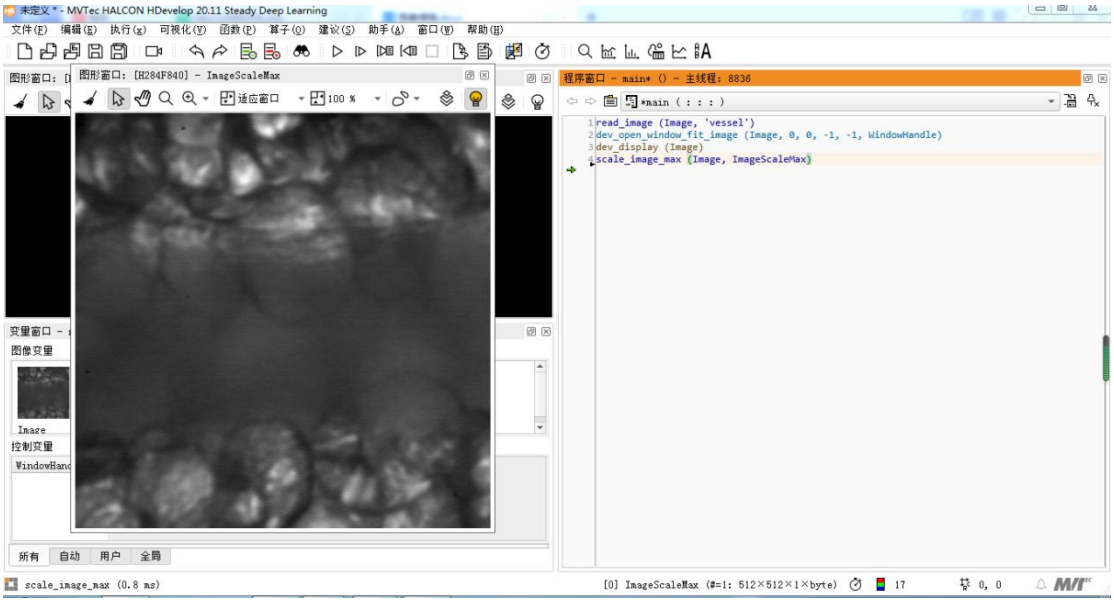
scale_image_max(Image : ImageScaleMax : :)最大灰度值范围覆盖到 0 至 255。

下面是 HDevelop 处理程序：

对比度归一化前：



对比度归一化后：

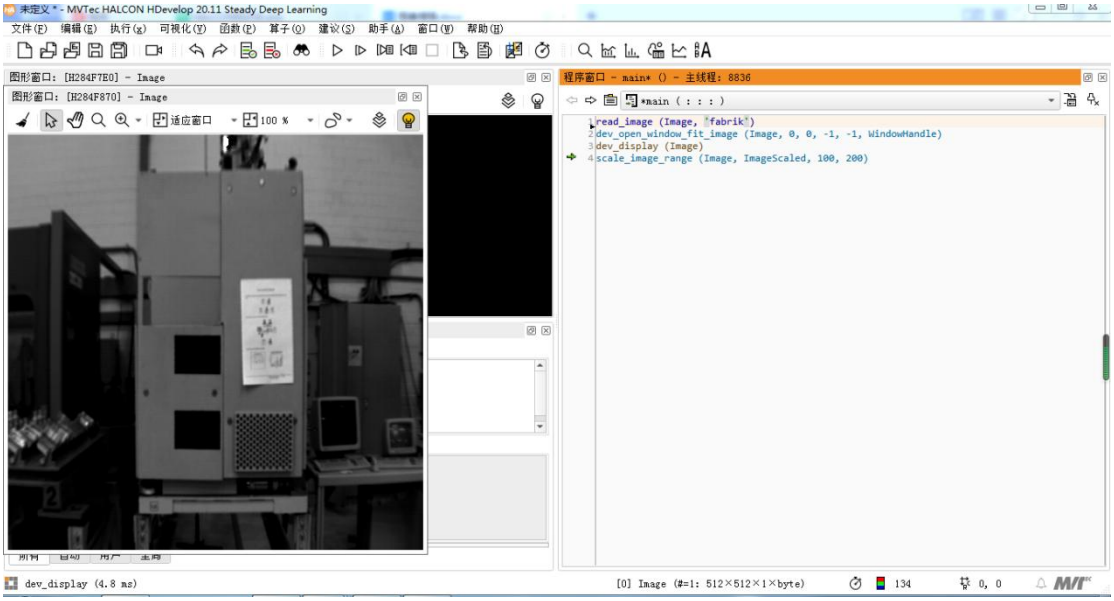


如果图像中存在一个非常亮（对于 byte 图像，灰度值接近 255）或者非常暗（对于 byte 图像，灰度值接近 0）的像素值，那么上述归一化处理的改善效果就并不理想。此时，HALCON 提供了另一个对比度归一化的算子“scale_image_range“。

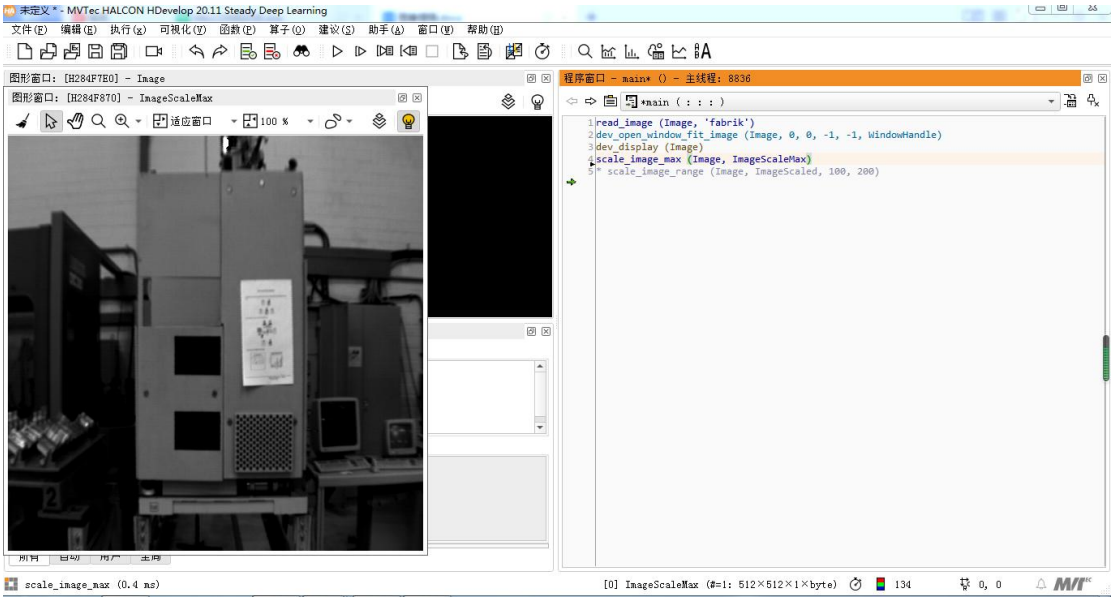
scale_image_range(Image：ImageScaled：Min, Max：）将图像的灰度值从区间[Min,Max]扩展到[0,255]。

下面是 HDevelop 处理程序：

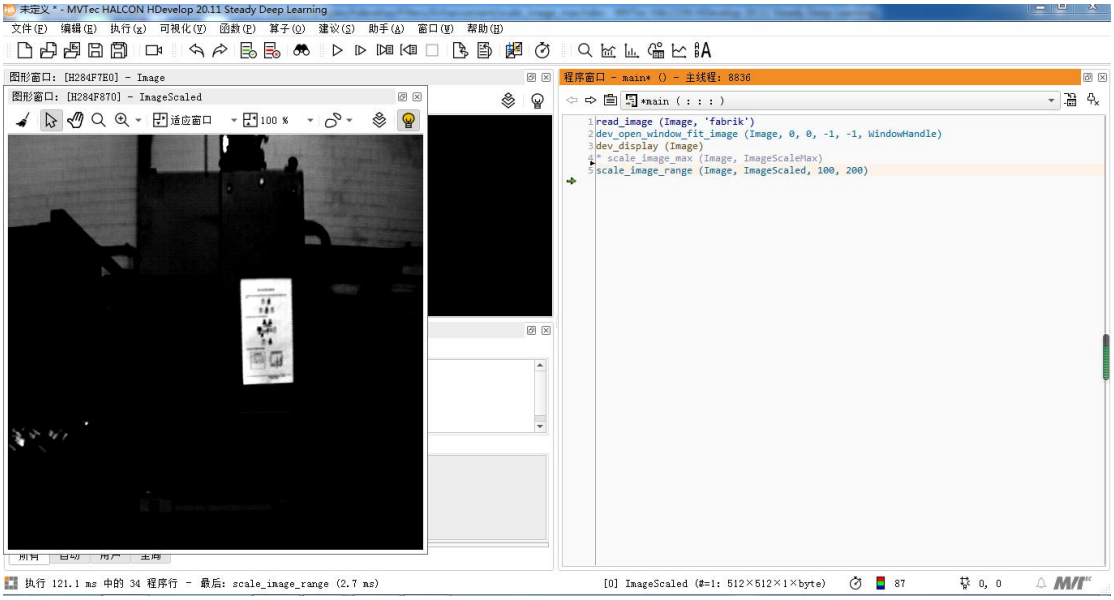
对比度归一化前：



使用 scale_image_max 处理效果：



使用 scale_image_range 处理效果：



以上是对灰度值变换的介绍。

1.2 图像平滑

由于图像采集、处理、传输等过程不可避免的会受到噪声的污染，这样会妨碍人们对图像理解及分析处理。常见的图像噪声有高斯噪声、椒盐噪声等，这些噪声体现在图像上一般为随机出现的黑点或白点。噪声属于高频信号，图像平滑从信号处理的角度看就是去除其中的高频信息，保留低频信息。因此我们可以对图像实施低通滤波。低通滤波可以去除图像中的噪声，对图像进行平滑。常用方法有均值滤波、中值滤波、高斯滤波、纹理滤波等。图像平滑的基本原理是，将噪声所在像素点的像素值处理为其周围临近像素点的值的近似值。

1.2.1 均值滤波

均值滤波也称为线性滤波，其采用的主要方法为邻域平均法。线性滤波的基本原理是用均值代替原图像中的各个像素值，即对待处理的当前像素点（x，y），选择一个模板(或称卷积、掩模)，如 3x3 区域，该模板由其近邻的若干像素组成，求模板中所有像素的均值，再把该均值赋予当前像素点（x，y），作为处理后图像在该点上的灰度 g（x，y），即 g（x，y）=∑f（x，y）/m，m 为该模板中包含当前像素在内的像素总个数。

如下图所示，对第 5 行第 5 列的像素点进行均值滤波时，通常情况下，我们会以该当前像素为中心，对行数和列数相等的一块区域内的所有像素点的像素取平均值。例如,我们可以以当前像素点的像素周围 3x3 区域内所有像素点的像素取平均值，也可以对周围 5x5 区

域内所有像素点的像素值取平均值。

23	158	140	115	131	87	131
238	0	67	16	247	14	220
199	197	25	106	156	159	173
94	149	40	107	5	71	171
210	163	198	**226**	223	156	159
107	222	37	68	193	157	110
255	42	72	250	41	75	184
77	150	17	248	197	147	150
218	235	106	128	65	197	202

当前像素点的位置为第 5 行第 5 列时，我们对其周围 5x5 区域内的像素值取平均，计算方法如下：

像素点新值 = [(197+25+106+156+159)+(149+40+107+5+71)+(163+198+226+223+156)

+ (222+37+68+193+157)+(42+72+250+41+75)]/25

= 126

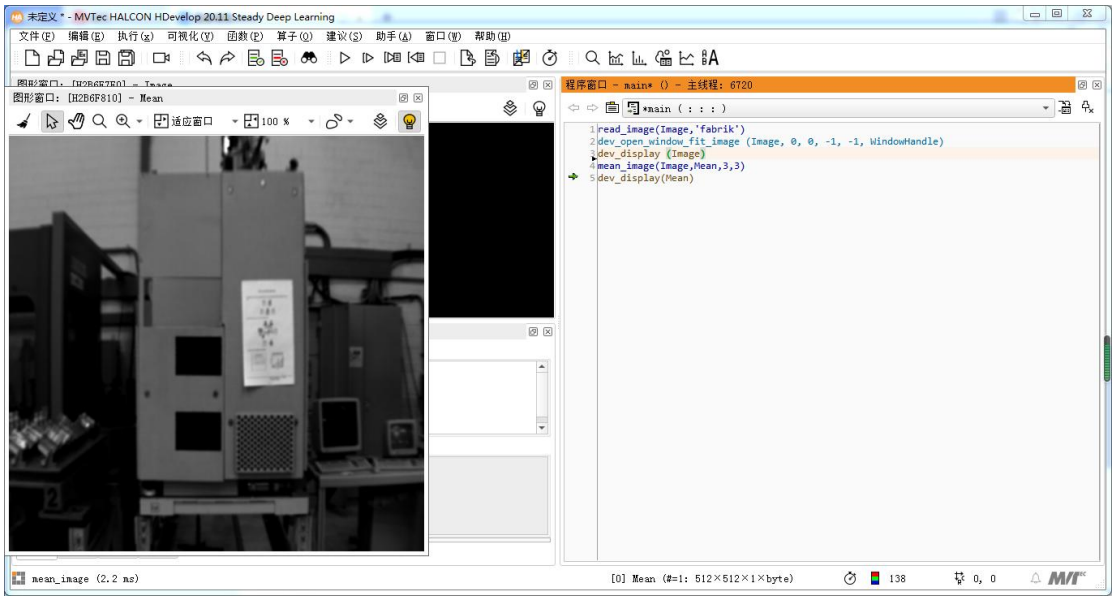
计算得到新值以后，我们将新值作为当前像素点均值滤波后的像素值。将图中的每一个像素点计算其周围 5x5 区域内的像素值均值，并将其作为当前像素点的新值，即可得到当前图像的均值滤波结果。图像的边界并不存在 5x5 的领域区域，针对边缘的像素点，可以只取图像内存在的周围领域点的像素值均值。

在 HALCON 中，与均值滤波有关的算子是 mean_image。

mean_image(Image：ImageMean：MaskWidth, MaskHeight：)该算子对所有输入图像(Image)的灰度值进行线性平滑，滤波矩阵大小为 MaskHeight x MaskWidth（奇数）。

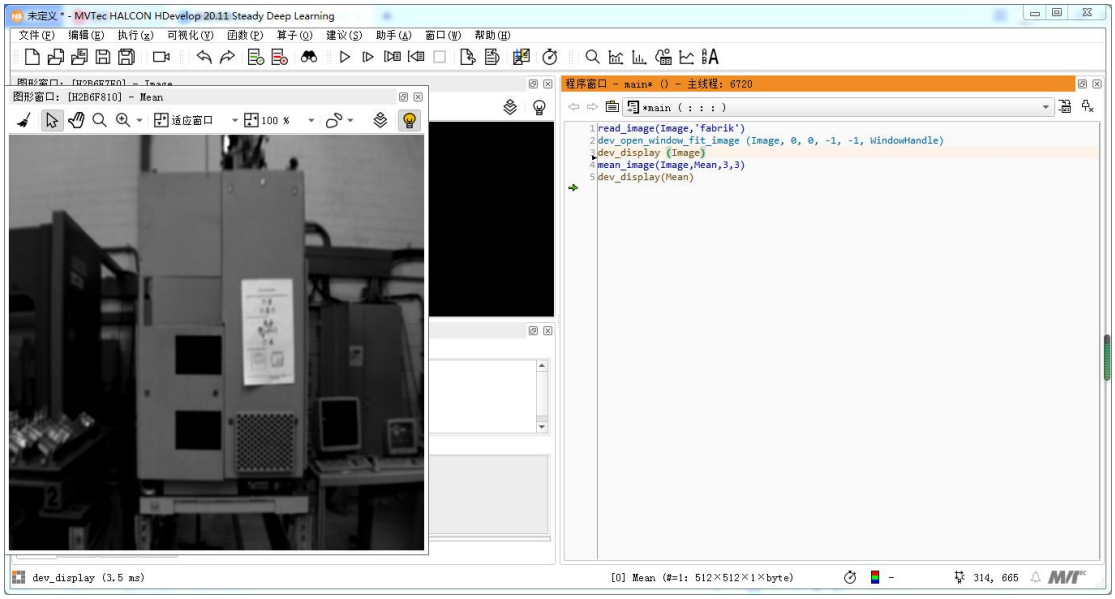
下面是 HDevelop 处理程序：

均值滤波前：

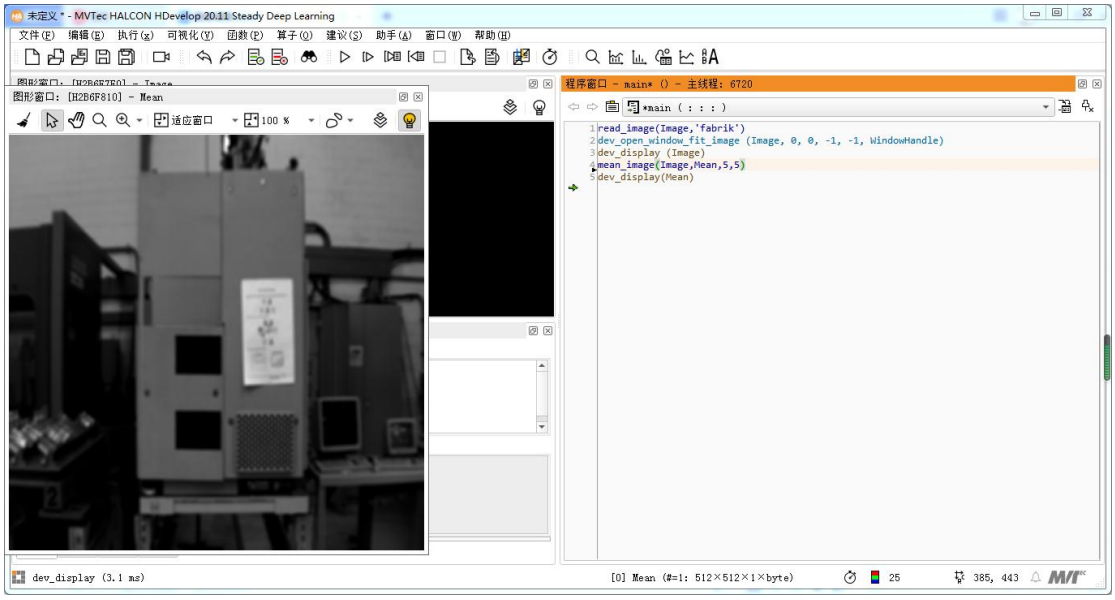


使用 mean_image 处理效果：

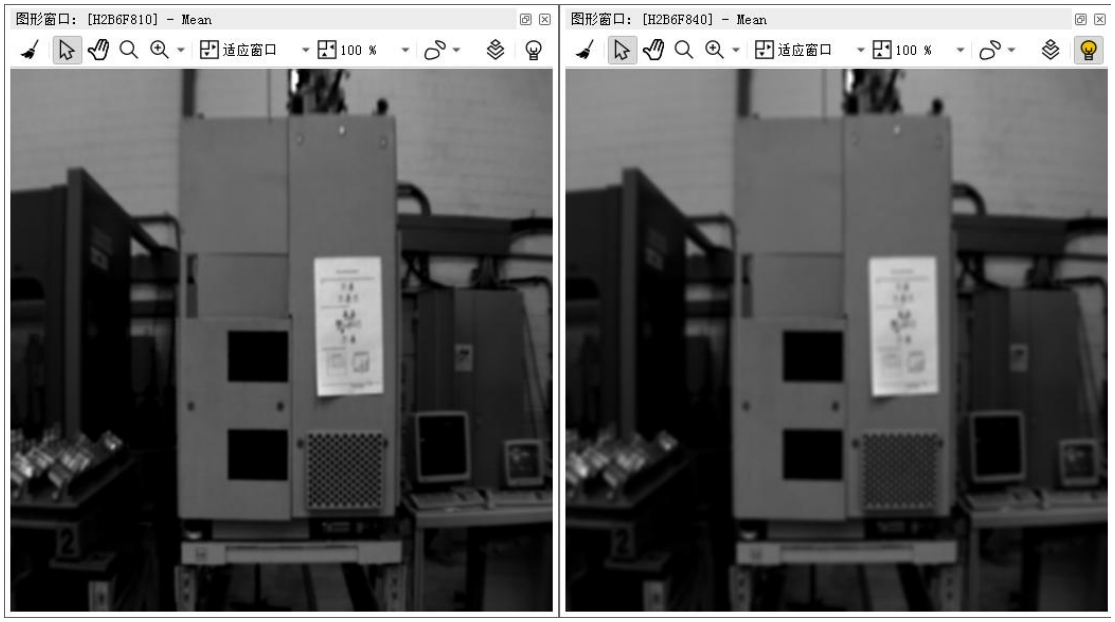
MaskHeight x MaskWidth = 3x3



MaskHeight x MaskWidth = 5x5



对比效果：左图 3x3，右图 5x5



均值滤波的优点是算法简单，计算速度较快，缺点是在去噪的同时去除了很多细节部分，将图像变得模糊。

1.2.2 中值滤波

均值滤波是线性滤波，而中值滤波法是一种非线性平滑技术，它将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值。方法是用某种结构的二维滑动模板，将板内像素按照像素值的大小进行排序，生成单调上升（或下降）的为二维数据序列。二维

中值滤波输出为 g（x,y）=med{f(x-k,y-l),(k,l∈W)} ，其中，f(x,y)，g(x,y)分别为原始图像和处理后图像。W 为二维模板，通常为 3x3，5x5 区域，也可以是不同的形状，如线状，圆形，十字形，圆环形等。

如下图所示，对第 5 行第 5 列的像素点进行中值滤波时，将把窗口中所含的像素点按灰度级的升或降序排列，用位于中间的灰度值来代替该点的灰度值。例如,我们可以以当前像素点的像素周围 5x5 区域内所有像素点的像素值降序排列，取排序像素集中位于中间位置的值作为中值滤波后的像素值。

23	158	140	115	131	87	131
238	0	67	16	247	14	220
199	197	25	106	156	159	173
94	149	40	107	5	71	171
210	163	198	**226**	223	156	159
107	222	37	68	193	157	110
255	42	72	250	41	75	184
77	150	17	248	197	147	150
218	235	106	128	65	197	202

250>226>223>222>198>197>193>163>159>157>156=156>149>107>106>75>72>71>68>42>41>40>37>25>5，降序排列后，中间值为 149。计算得到新值以后，我们将新值作为当前像素点中值滤波后的像素值。

在 HALCON 中，与均值滤波有关的算子是 median_image。

median_image(Image : ImageMedian : MaskType, Radius, Margin :)使用不同形状的模板(或称卷积、掩模)对输入图像进行中值滤波，模板的形状可以用“MaskType”来选择圆形“circle”或矩形“square”，模板的半径可以用“Radius”来选择，Radius = 1 或 = 2 相当于 3x3 或 5x5 区域。参数“Margin”与边界处理有关：

gray value：假定图像边界之外的像素是恒定的（具有指定的灰度值）。

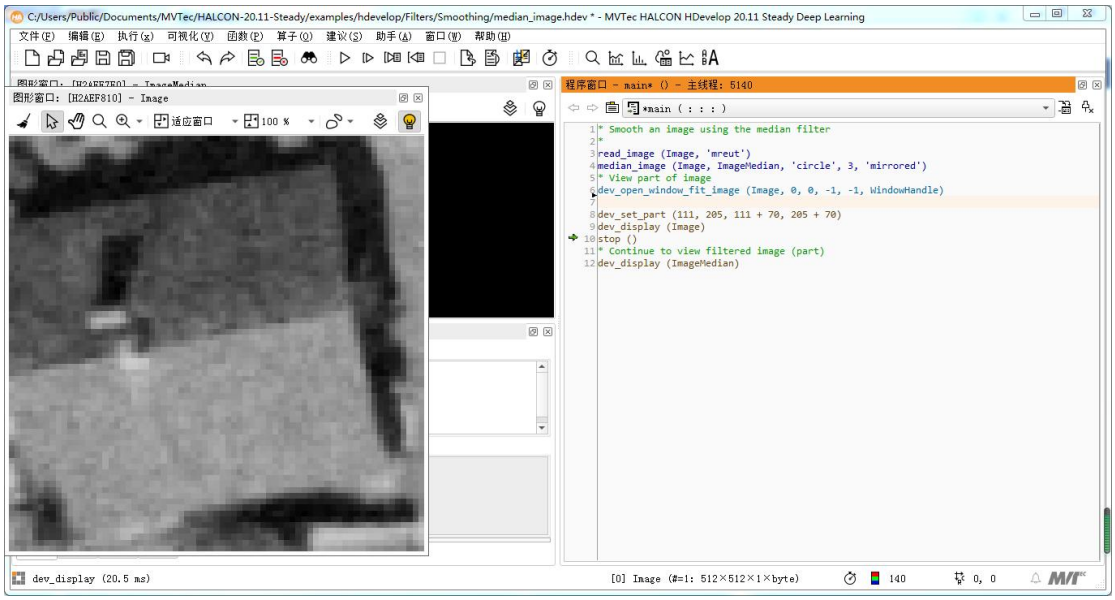
continued：边界像素的连续。

cyclic：图像边界的循环连续。

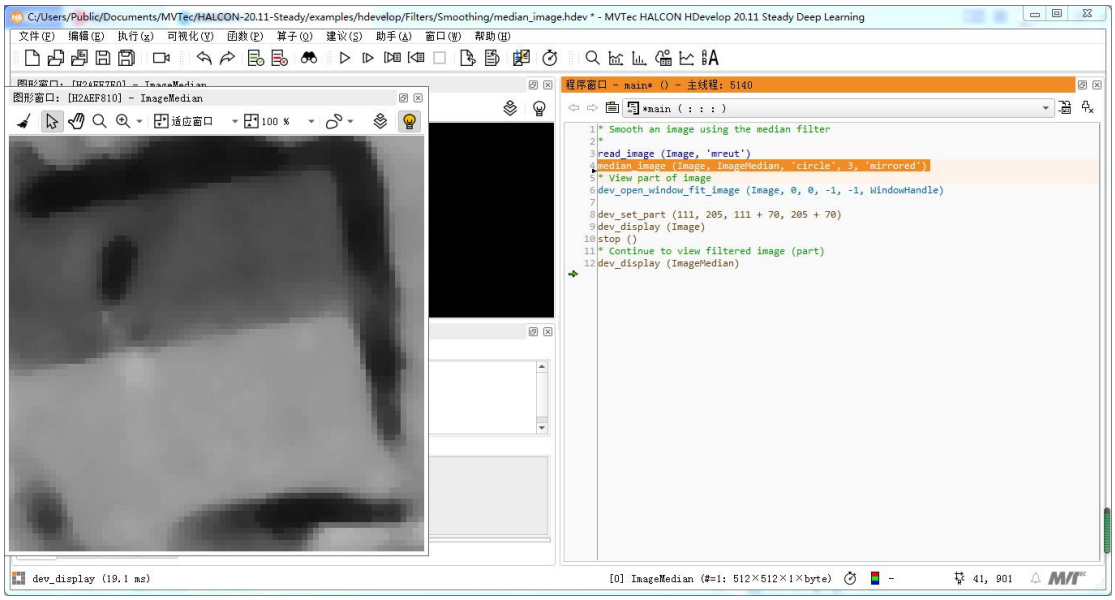
mirrored：图像边界处的像素反射。

下面是 HDevelop 处理程序：

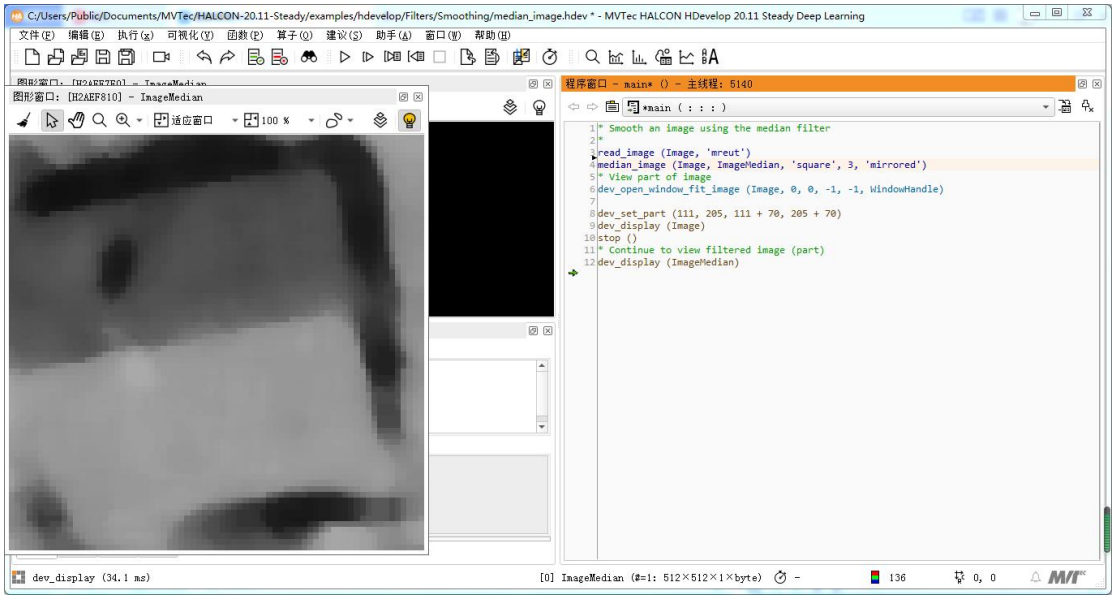
中值滤波前：



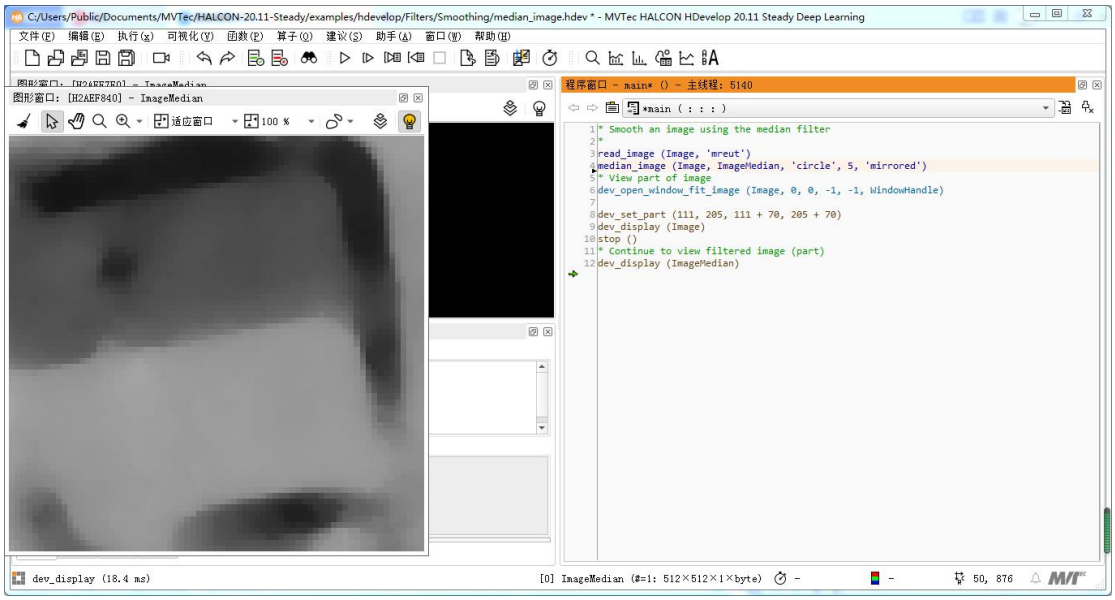
MaskType='circle'，Radius=3



MaskType='square'，Radius=3



MaskType='circle'，Radius=5

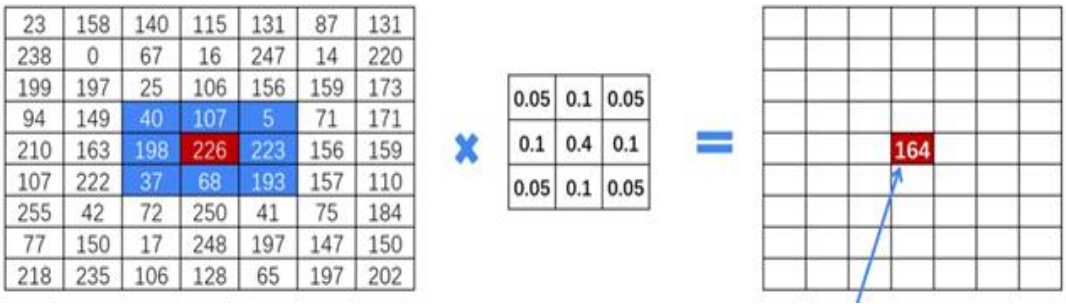


与均值滤波相比，中值滤波可消除图像中孤立的噪声点，又可产生较少的模糊。

1.2.3 高斯滤波

高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，广泛应用于图像处理的减噪过程。高斯滤波的具体原理是:用一个模板(或称卷积、掩模)扫描图像中的每一个像素，用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值，如下图所示。图像高斯平滑也是邻域平均的思想对图像进行平滑的一种方法，在图像高斯平滑中，对图像进行平均时，不

同位置的像素被赋予了不同的权重。高斯平滑与简单平滑不同，它在对邻域内像素进行平均时，给予不同位置的像素不同的权值。



在 HALCON 中，与高斯滤波有关的算子有 smooth_image，gauss_filter，derivate_gauss，'binomial_filter'等，下面主要介绍 gauss_filter。

gauss_filter(Image : ImageGauss : Size :) 使用离散高斯函数平滑图像。一个离散高斯近似函数为：

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

利用高斯分布函数，生成高斯模板，然后用模板取扫描图像中的每一个像素，用模板确定的领域内像素的加权平均值作为新图像中模板中心位置的像素值。平滑效果随着模板尺寸增加而增强，算子支持的尺寸如下：

3 (0.600)

5 (1.075)

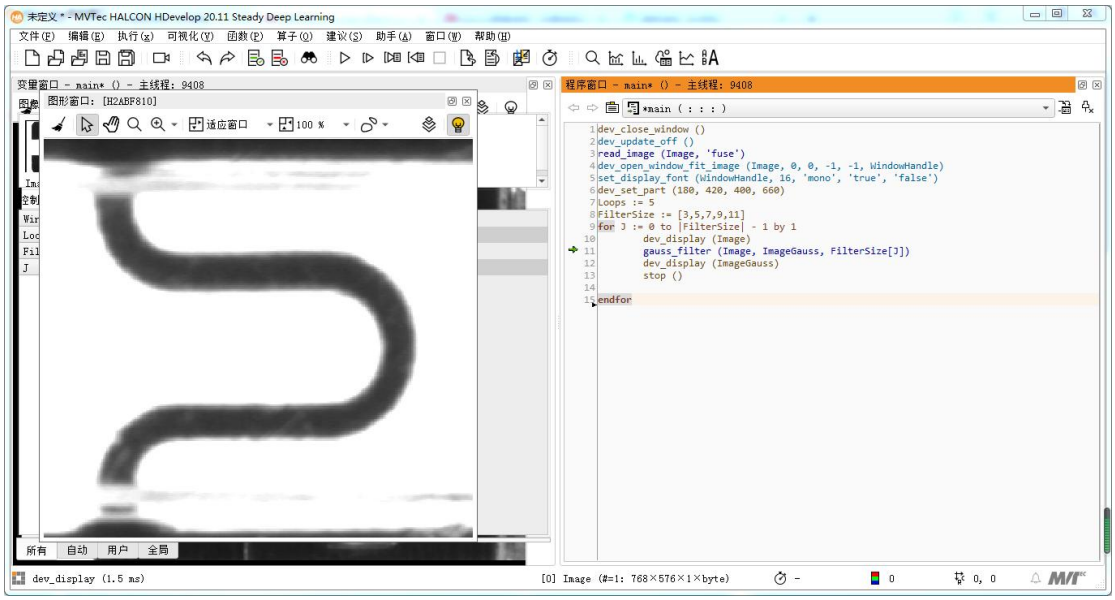
7 (1.550)

9 (2.025)

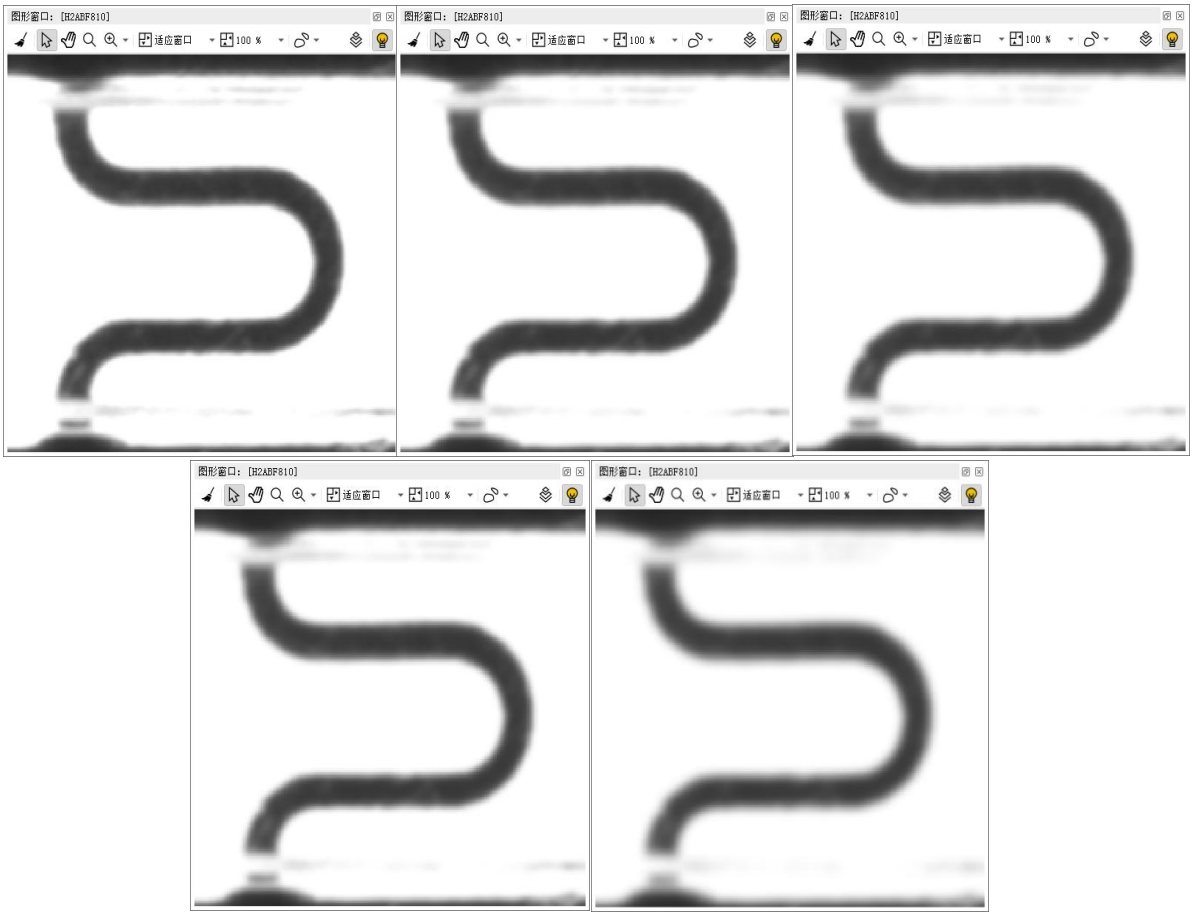
11 (2.550)

下面是 HDevelop 处理程序：

高斯滤波前：



下面分别是 Size=3、5、7、9 和 11 的处理效果：



高斯滤波是一种线性平滑滤波，可以去除高斯噪声，其效果是降低图像灰度的尖锐变化，也就是图像模糊了。高斯滤波对于抑制服从正态分布的噪声效果非常好，其代价是使图像变得模糊。

1.3 傅里叶变换

傅里叶变换(FT, Fourier Transform)的作用是将一个信号由时域变换到频域，把图像数据由横坐标时间、纵坐标采样值的波形图格式，转换为横坐标频率、纵坐标振幅(或相位)的频谱格式。原来的空间域图上，每一点的坐标是（x,y），Z 坐标是灰度值；在频域图上，每一点的坐标是（x,y），Z 坐标是幅度值。随着域的不同，对同一个事物的了解角度也就随之改变，因此在时域中某些不好处理的地方，在频域就可以较为简单的处理，一般用于对出现频率高的像素点的分析以及噪声的去除。

傅里叶变换的函数为：

$$F(m,n)=\frac{1}{c}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}e^{s2\pi i(km/M+ln/N)}f(k,l)$$

经过傅里叶变换得到的频谱图像有以下特征：

1. 对称性频谱图像一般以图像中心为原点，左上右下对称，右上左下对称。

2. 频率是表征图像中灰度变化剧烈程度的指标。灰度变化剧烈的地方（如边缘），梯度大，该位置频率高；在图像中灰度变化缓慢的地方，就是梯度小的地方，频率低。

3. 频域图像中心一般是低频成分。从中心往外频率是逐渐增加的，每一点亮度值越高，表示这个频率特征很凸出，亮点越多表示该频率成分越多。

HALCON 中与傅里叶变换相关的算子有：fft_generic 快速傅里叶变换、fft_image_inv 快速傅里叶逆变换、optimize_fft_speed 优化 FFT 运行时间等。对 fft_generic 进行介绍之前，我们来了解一下快速傅里叶变换：

快速傅立叶变换(FFT)是离散傅立叶变换(DFT)的快速算法。DFT 的算法属于线性变换，由于对每个采样点，都要做一次全部点的加权求和的运算，因此当采样点比较多时，运算速度会很慢。FFT 运算结果和 DFT 是相等的。其原理是利用权值的对称性与周期性，把采样点分解成两份，每份的点数是原来的一半，这样运算量也会减半。然后可以继续分解为 4 份、8 份、16 份……以此不断提升效率。

fft_generic(Image : ImageFFT : Direction, Exponent, Norm, Mode, ResultType :)

该算子计算图像的快速傅里叶变换，可实现正/逆向变换。部分参数定义如下：

Direction：‘from_freq’从频域到空间域（前向变换）；‘to_freq’从空间域到频域（后向变换）。

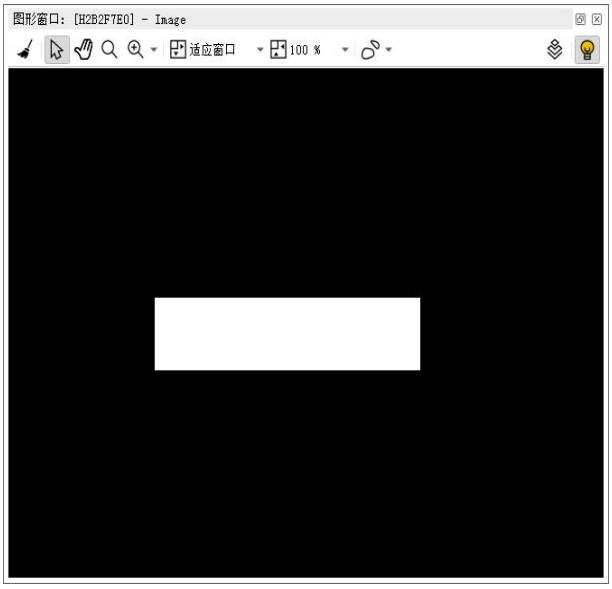
Exponent：对应公式中指数符号 s，前向变换用-1，后向变换用 1。

Norm：对应公示中 c，为归一化系数。‘none’表示不做归一化，此时 c=1；‘sqrt’表示 c=宽高乘积开方；‘n’表示 c=宽高乘积。

Mode:‘dc_center’频域图中心频率为 0，‘dc_edge’频域图边角点频率为 0。

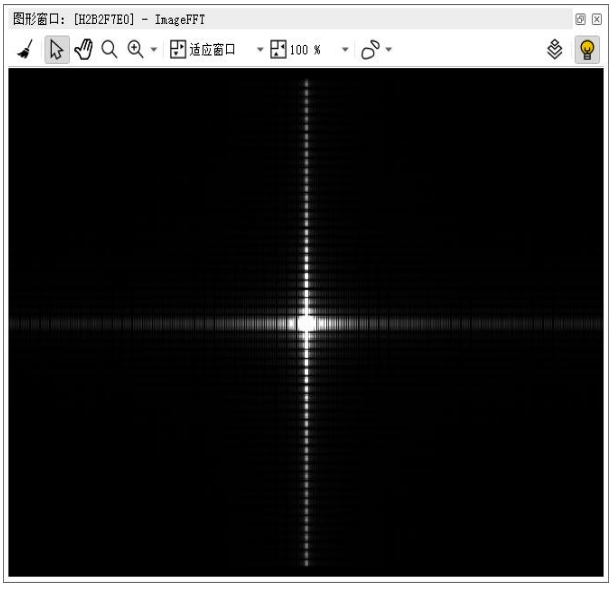
ResultType：指定后向变换后图像类型，前向变换图像类型必须为‘complex’。

下面是一张测试图像，我们来看一下在频谱中心为高频时的情况，还有频谱中心为低频时的情况。



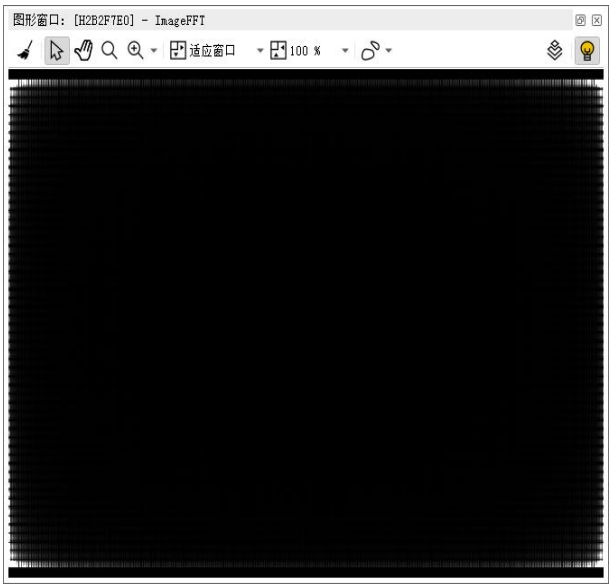
频谱中心为低频时：

fft_generic (Image, ImageFFT, 'to_freq', -1, 'sqrt', 'dc_center', 'complex')



频谱中心为高频时：

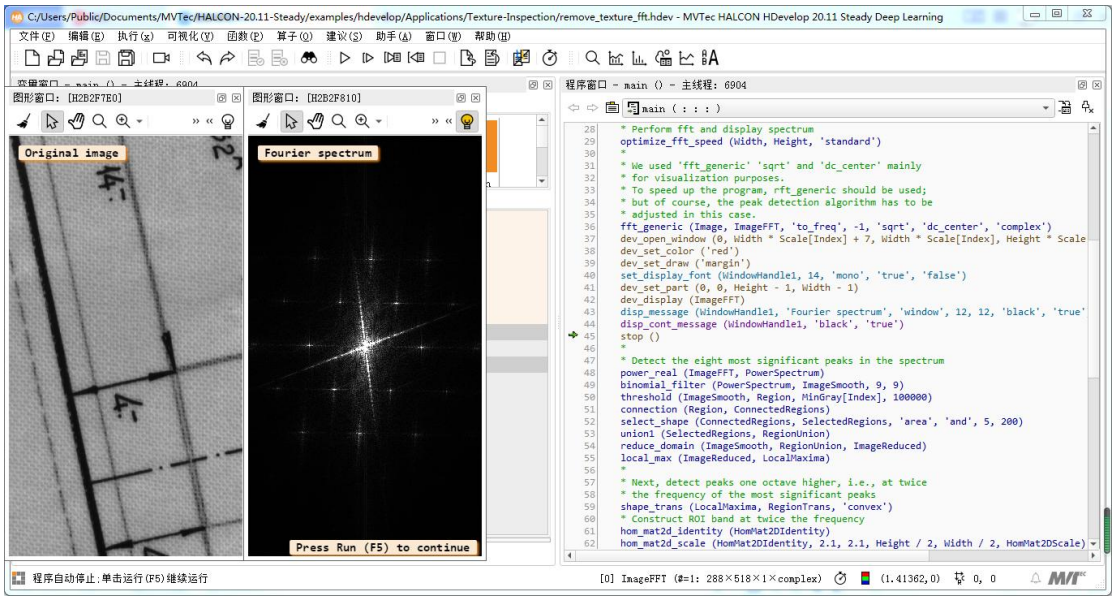
fft_generic (Image, ImageFFT, 'to_freq', -1, 'sqrt', 'dc_edge', 'complex')



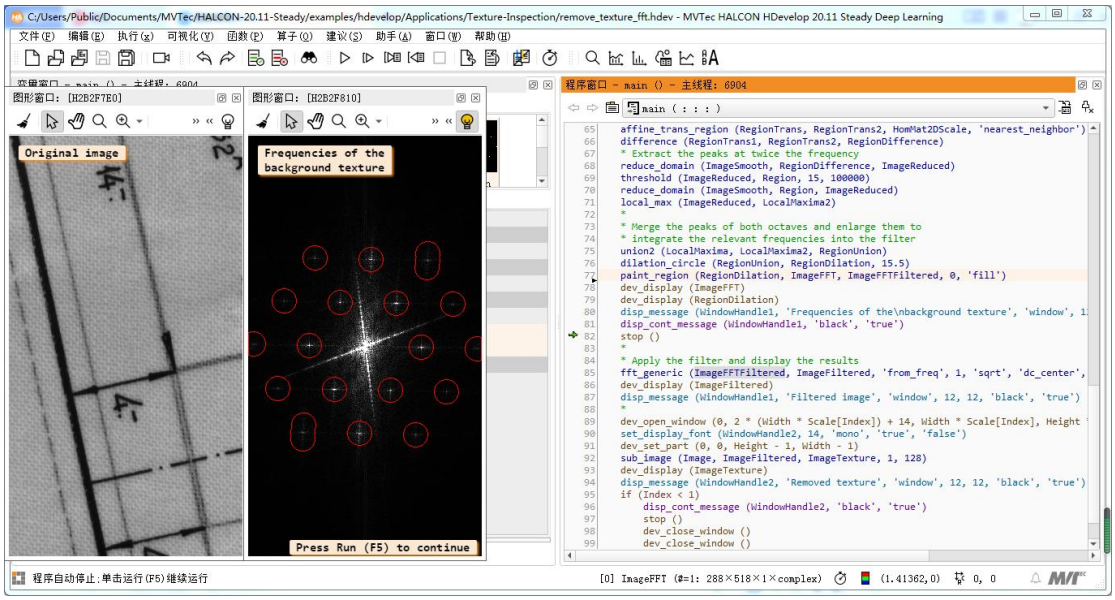
我们可以看出，中心为低频的时候，竖直方向比较亮，说明变化比较剧烈，比较明显。因为竖直方向是长边，所以变化的像素多，涵盖的频谱多。对于水平方向，是短边，也有变化，但是涵盖的频谱不够多，所以不够宽，变化剧烈的少，所以亮度也不够。低频代表图像中变化不明显的地方，为背景。高频为变化剧烈的地方，说明是边界。

下面我们看下傅里叶变换在 HDevelop 图像处理中的具体应用。以图像中的纹理为例，有时为了后续的处理效果，需要去除这些纹路的干扰，如果从空间域上删除难度较大，那么此时可以用频域做对应变换。以下是使用快速傅里叶变换去除图像纹理的程序及处理效果：

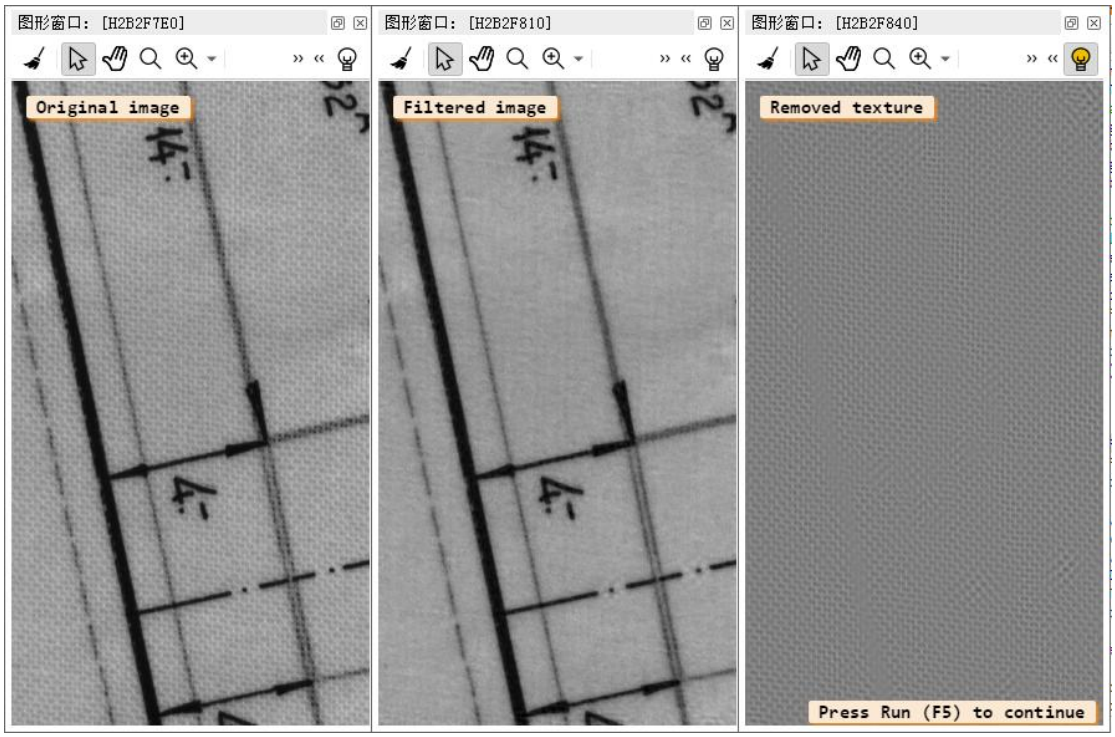
首先将图像转换成频谱图：



变换后，能够看见频谱图上有很多亮点，这些亮点属于高频成分，也就是需要去除的纹理。目前使用的是用算子 paint_region 直白的将某个区域染成黑色，以达到去除的目的。



以下是原图和处理后的效果对比：



以上就是图像增强的全部内容，所涉及的专业内容比较多，如有不理解的问题，请您留言或随时咨询大恒图像，技术热线：400-999-7595。请关注“大恒图像”微信公众号，视觉学院→大恒课堂栏目下阅读其他关于图像处理方法的文章。