

7.2.6 双边滤波

接下来我们介绍双边滤波，这种滤波方式有一个很大的缺点，那就是运行速度十分缓慢，但相对的，去噪效果也比较好。

双边滤波能在保持图像轮廓清晰的情况下进行噪声的去除，是一种需要结合图像的空间邻近度与像素值相似度的去噪方法。在卷积时，该卷积方法会同时考虑像素点与像素点之间的空间信息和颜色信息（颜色相似度），从而做到在平滑图像的同时，保存图像边缘。因为它需要的数据量比前几种滤波方法要多得多，所以运行速度比较缓慢。

双边滤波使用的函数为 `cv2.bilateralFilter` 函数，其主要语法如下。

```
dst=cv2.bilateralFilter(src,d,sigmaColor,sigmaSpace)
```

其参数分别解释如下。

- ❑ `src`: 要进行双边滤波的原图像。
- ❑ `d`: 邻域直径。
- ❑ `sigmaColor`: 灰度值相似性高斯函数标准差。
- ❑ `sigmaSpace`: 空间高斯函数标准差。
- ❑ `dst`: 输出图像。

双边滤波在进行图像卷积运算的时候会同时使用空间权重和灰度值相似性权重。

灰度值相似性高斯函数用来判断周边的像素点是否与中心像素点灰度值相近，只有灰度值相似的像素点才会被用作卷积运算。空间高斯函数用来确认像素点是否处在中心点周边，只有在中心点周边的像素点才会对像素点的中心点产生影响。

双边滤波有一个十分经典的应用：美颜滤镜。我们以图 7.15 所示的原图为例进行介绍，代码如下。

```
import cv2
import numpy as np
#读取图像
img=cv2.imread('6.jpg')
x=img.shape[0]
y=img.shape[1]
#双边滤波
dst=cv2.bilateralFilter(img,11,40,40)
#图像显示
cv2.namedWindow('img',cv2.WINDOW_NORMAL)
cv2.resizeWindow('img',int(y),int(x))
cv2.imshow('img',dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

运行上述代码，结果如图 7.16 所示。