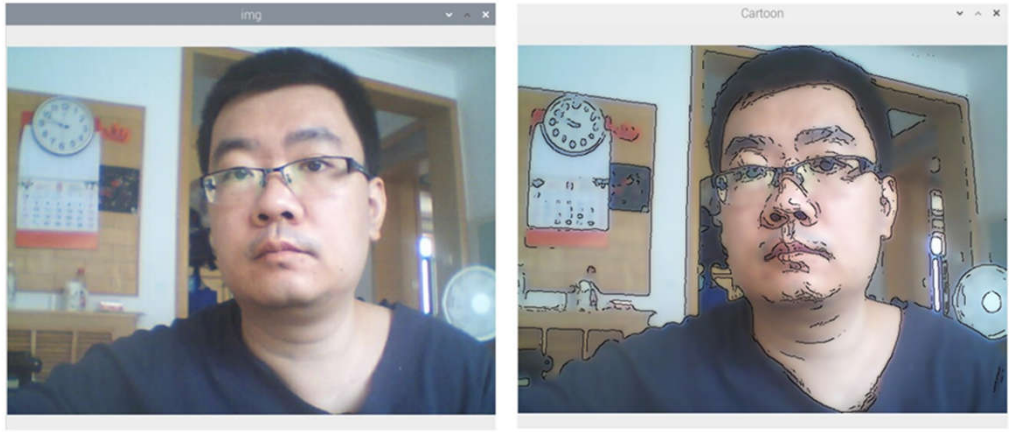
智能系统与控制

树莓派：OpenCV 边缘检测（卡通特效）



于泓
鲁东大学
信息与电气工程学院
2022.1.9

Candy边缘检测

在 OpenCV 中提供了很多的边缘提取算法，其中 Canny 算法是一种在现实生活中较为经常使用的方法，在实际的应用中利用 Canny 算法进行边缘提取通常需要用到步骤。

Canny 算法进行特征提取的算法，大致如下:
(1) 利用 Soble 算子计算每个像素点 X 方向和 Y 方向的梯度 G_x 与 G_y, 并利用他们计算边缘强度 G_{edge};
(2) 根据非最大值抑制的原则，选取局部拥有较大 G_{edge} 的像素点作为备选的边缘点;
(3) 利用两个阈值 minVal 和 maxVal 对备选边缘点进行进一步筛选，对于 G_{edge} 大于 maxVal 的像素点，被选定为强边缘， G_{edge} 小于 minVal 的点被抛弃；对于边缘强度在 minVal 和 maxVal 之间的点，根据其是否与强边缘点有连接，有连接则保留为边缘点，没有则被抛弃。代码 14 行上的两个参数 150 与 200 就是边缘选择阈值 minVal 和 maxVal。通过调节这两个参数可以控制提取边缘数目的多少。

```python
import cv2
import numpy as np

if __name__ == '__main__':
    # 图像颜色转换 模糊处理 边缘提取  膨胀与 腐蚀细化
    #读取图像
    img = cv2.imread("lena.png")

    #颜色转换
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # 图像模糊 简单去噪
    imgBlur = cv2.GaussianBlur(imgGray,(5,5),0)

    # 边缘提取
    imgCanny = cv2.Canny(imgBlur,150,200)

    # 图像的膨胀 对细小的边缘进行连接
    kernel = np.ones((5,5),np.uint8)
    imgDialation = cv2.dilate(imgCanny,kernel, iterations=1)

    # 图像的腐蚀细化 获取更为准确的边缘的定位
    imgEroded = cv2.erode(imgDialation,kernel,iterations = 1)

    # 各阶段图像像显示
    cv2.imshow("output",img)
    cv2.imshow("Gray",imgGray)
    cv2.imshow("Blur",imgBlur)
    cv2.imshow("Canny",imgCanny)
    cv2.imshow("Dialate",imgDialation)
    cv2.imshow("Erode",imgEroded)
    cv2.waitKey(0)
```

灰度转换

方差

核的大小

边缘连接

腐蚀细画

(a) 原始图像　(b) 灰度图像　(c) 高斯模糊

(d) Canny边缘　(e) 边缘膨胀　(f) 腐蚀细化

卡通特效

最近市场上出现了很多图像风格化的软件，其中的卡通风格受到很多人的喜欢，卡通风格图像有两个主要特点：

第一是图像的边缘非常清晰突出；
第二是图像的颜色特别平滑。

在本节中我们将利用 Opencv 的边缘检测功能以及图像平滑功能实现简易的卡通风格变换。

```python
import cv2
import numpy as np


if __name__ == "__main__":
    # 打开摄像头
    cap = cv2.VideoCapture(0)

    while True:
        # 读取图像帧
        success, img = cap.read()

        #颜色转换
        imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        # 中值滤波图像去噪
        imgBlur = cv2.medianBlur(imgGray,5)

        # 边缘提取
        imgCanny = cv2.Canny(imgBlur,50,50)

        # 图像平滑
        color = cv2.bilateralFilter(img, 9, 300, 300)

        # 平滑图像与边缘图像的融合
        img_edge = cv2.cvtColor(255-imgCanny, cv2.COLOR_GRAY2RGB)
        cartoon = cv2.bitwise_and(color, img_edge)

        # 卡通图像显示
        cv2.imshow("Cartoon",cartoon)
        cv2.imshow("img",img)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    cap.release()
```

中值滤波

bilateralFilter 是一种双边滤波的方案，从距离空间以及色彩空间两个角度进行滤波处理。该算法在进行色彩平滑的同时还能够较好的保存图像的边缘信息。 bilateralFilter() 函数的第二个参数表示滤波窗口的大小，第三、第四个参数分别表示距离空间模板以及色彩空间模板权重的方差。