机器视觉均值滤波作业

SZ170410221-朱方程

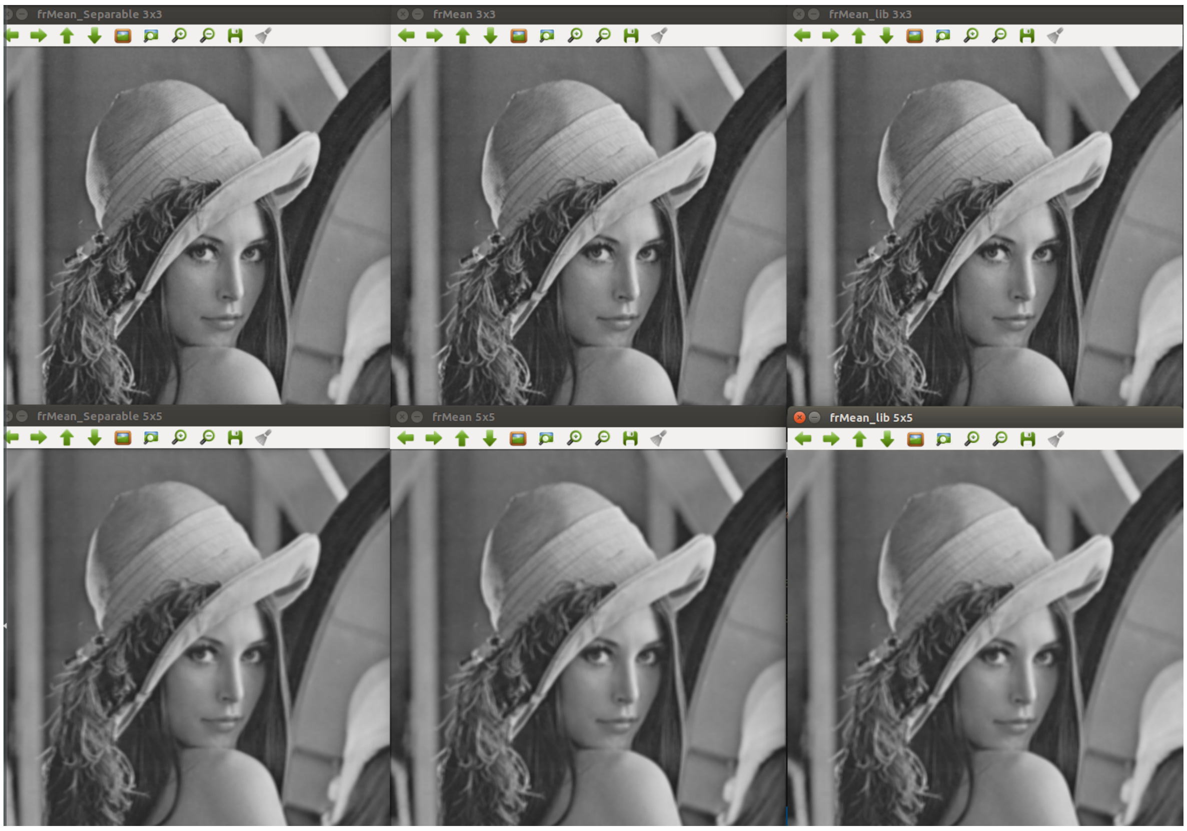
Code

```
//运行平台Ubuntu16.04
#include <stdlib.h>
#include <cv.h>
#include <highgui.h>
#include <opencv2/opencv.hpp>
#include <opencv2/core/core.hpp>
#include "ros/ros.h"
#include "std_msgs/String.h"
#include "std_msgs/Bool.h"
#include "std_msgs/Float32.h"
#include <geometry_msgs/Twist.h>
#include "sensor_msgs/Image.h"
#define LINEAR_X 0
using namespace cv;
using namespace std;
//按定义对图像进行均值滤波
void Mean(Mat input, Mat output, int MaskSize)
{
    int center_k=MaskSize/2;
    int center_l=MaskSize/2;
    double mask = 1.00000/(MaskSize*MaskSize);
    //将模板函数与图像进行卷积
    for(int i=0;i<input.rows;i++){
        for(int j=0;j<input.cols;j++){
            double sum = 0;
            for (int k = 0; k <MaskSize; k++){
                for (int l = 0; l < MaskSize; l++){
                    if(i+(k-center_k)>=0 && j+(l-center_l)>=0)
                    sum = sum+input.at<uchar>(i+(k-center_k),j+(l-
center_l))*mask;
                }
            }
            //对输出图像重新赋值
            output.at<uchar>(i,j)=sum;
        }
    }
}
//可分离的滤波器
void Mean_Separable(Mat input, Mat output, int MaskSize)
{
    int center_k=MaskSize/2;
    int center_l=MaskSize/2;
    double mask_row = 1.000000/MaskSize;
    double mask_col = 1.000000/MaskSize;
    //先将模板函数与图像进行横向卷积
    for(int i=0;i<input.rows;i++){
```

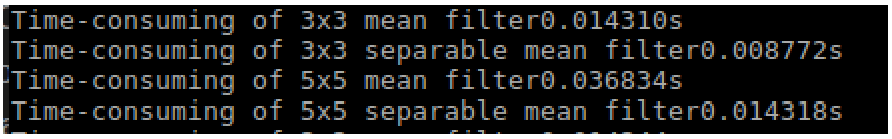
```cpp
        for(int j=0;j<input.cols;j++){
            double sum = 0;
            for (int l = 0; l < MaskSize; l++){
                if(j+(l-center_l)>=0)
                sum = sum+input.at<uchar>(i,j+(l-center_l))*mask_row;
            }
            //对输出图像重新赋值
            output.at<uchar>(i,j)=sum;
        }
    }
    //先将模板函数与图像进行纵向卷积
    for(int i=0;i<input.rows;i++){
        for(int j=0;j<input.cols;j++){
            double sum = 0;
            for (int k = 0; k < MaskSize; k++){
                if(i+(k-center_k)>=0)
                sum = sum+output.at<uchar>(i+(k-center_k),j)*mask_col;
            }
            //对输出图像重新赋值
            output.at<uchar>(i,j)=sum;
        }
    }
}


int main(int argc, char **argv)
{

    VideoCapture capture;
        capture.open(0);//打开 zed 相机

    ROS_WARN("*****START");
    ros::init(argc,argv,"trafficLaneTrack");
        ros::NodeHandle n;

    ros::Rate loop_rate(10);
        ros::Publisher pub = n.advertise<geometry_msgs::Twist>("/smoother_cmd_vel", 5);
    if (!capture.isOpened())
    {
        printf("摄像头没有正常打开\n");
        return 0;
    }
    waitKey(1000);
    Mat frame;
    while (ros::ok())
    {


        Mat frIn1 = imread("/home/fangcheng/Library/lena1.png",1);
            if(frIn1.empty())
        {
            break;
        }
        Mat frIn;
        cvtColor(frIn1, frIn, CV_RGB2GRAY);//RGB彩色图转换成Gray灰度图


        /*capture.read(frame);
        if(frame.empty())
        {
            break;
        }
        Mat frIn1 = frame(cv::Rect(0, 0, frame.cols, frame.rows));//截取zed 的图片
        Mat frIn;
        cvtColor(frIn1, frIn, CV_RGB2GRAY);//RGB彩色图转换成Gray灰度
        */
        clock_t start,finish;

        //均值滤波处理
        Mat frMean3 = frIn.clone();
        Mat frMean_Separable3 = frIn.clone();
        Mat frMean5 = frIn.clone();
        Mat frMean_Separable5 = frIn.clone();
        //计时
        start=clock();
        Mean(frIn,frMean3,3);
        finish=clock();
        printf("Time-consuming of 3x3 mean filter%fs\n",(double)(finish-start)/CLOCKS_PER_SEC);

        start=clock();
        Mean_Separable(frIn,frMean_Separable3,3);
        finish=clock();
        printf("Time-consuming of 3x3 separable mean filter%fs\n",(double)(finish-start)/CLOCKS_PER_SEC);

        start=clock();
        Mean(frIn,frMean5,5);
        finish=clock();
        printf("Time-consuming of 5x5 mean filter%fs\n",(double)(finish-start)/CLOCKS_PER_SEC);

        start=clock();
        Mean_Separable(frIn,frMean_Separable5,5);
        finish=clock();
        printf("Time-consuming of 5x5 separable mean filter%fs\n",(double)(finish-start)/CLOCKS_PER_SEC);

        imshow("frIn",frIn);//灰度图像
        imshow("frMean 3x3",frMean3);//按定义编写的均值滤波
        imshow("frMean_Separable 3x3",frMean_Separable3);//可分离的均值滤波器
        imshow("frMean 5x5",frMean5);//按定义编写的均值滤波
        imshow("frMean_Separable 5x5",frMean_Separable5);//可分离的均值滤波器
        //opencv库函数均值滤波
        Mat frMean_lib3 = frIn.clone();
        Mat frMean_lib5 = frIn.clone();
        blur(frIn,frMean_lib3,Size(3,3),Point(-1,-1));
        blur(frIn,frMean_lib5,Size(5,5),Point(-1,-1));
        imshow("frMean_lib 3x3",frMean_lib3);
        imshow("frMean_lib 5x5",frMean_lib5);

        ros::spinOnce();
        waitKey(5);
    }
    return 0;
}
```

滤波效果

原图:



耗时



```
Time-consuming of 3x3 mean filter0.014310s
Time-consuming of 3x3 separable mean filter0.008772s
Time-consuming of 5x5 mean filter0.036834s
Time-consuming of 5x5 separable mean filter0.014318s
```