

实验一 图像获取与直方图均衡化

1 实验目的

- ① 掌握 ROS 下通过 OpenCV 读取 ZED 相机的图像的方法
- ② 掌握通过 topic 控制机器人运动的方法
- ③ 掌握图像的直方图计算及其均衡化原理，并设计算法

2 实验仪器

- ① 机器人硬件：移动机器人、笔记本
- ② 系统和软件：Ubuntu16.04、ROS-kinetic-full、OpenCV（仅用于实现图像的读取操作）

3 实验原理

3.1 图像直方图计算

图像直方图使用以表示数字图像中亮度分布的直方图，标绘了图像中每个亮度值的像素数，如图 1 所示。将直方图除以面积进行归一化就成了概率密度函数，图像中灰度值为 k 的像素得到概率密度满足：

$$p_k = \frac{n_k}{N} \quad (1.1)$$

其中 n_k 为灰度值为 k 的像素数， N 为图像中的像素总数。



Fig 1 图像及其直方图

3.2 直方图均衡化

直方图均衡化广泛应用于图像增强处理。图像的像素灰度变化是随机的，直方图的图形高低不齐，把原始图像的灰度直方图从比较集中的某个灰度区间变成在全部灰度范围内的均匀分布。直方图均衡化就是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。

设 r 和 s 分别表示原图像灰度级和经过直方图均衡化的图像灰度级，并将其归一化。设连续图像的概率分布为 $P_r(r)$ 和 $p_s(w)$ ，假定直方图均衡化变换函数为 $s = T(r)$ ，则变换前后有：

$$\int_r^{r+\Delta r} p_r(w)dw = \int_s^{s+\Delta s} p_s(w)dw$$

设 $C(r)$ 为累积分布函数，一般地有：

$$\int_{s_{\min}}^s p_s(w)dw = \int_{r_{\min}}^r p_r(w)dw = C(r)$$

若期望变换后输出图像的概率密度均匀分布，即：

$$p_s(s) = \frac{1}{s_{\max} - s_{\min}}$$

则累积分布函数满足：

$$\begin{aligned} C(r) &= \int_{s_{\min}}^s \frac{1}{s_{\max} - s_{\min}} dw \\ &= \frac{1}{s_{\max} - s_{\min}} (s - s_{\min}) \end{aligned}$$

因此，像素值经过直方图均衡化后满足：

$$s = [s_{\max} - s_{\min}]C(r) + s_{\min}$$

4 实验内容

4.1 硬件连接

- ① 笔记本连接对应编号机器人上的 WiFi，形成局域网
- ② 打开机器人上的主电源

4.2 访问机器人电脑

- ① 连接

打开终端，输入命令：\$ ssh eaibot@192.168.31.200

密码：eaibot（后文使用到将不再赘述）

通过该终端窗口即可远程访问机器人电脑。

② 测试：

```
$ roslaunch dashgo_driver driver.launch
```

打开另一个终端，运行以下命令，实现键盘控制机器人移动：

```
$ ssh eaibot@192.168.31.200
```

```
$ rosrundashgo_tools teleop_twist_keyboard.py
```

4.3 软件部署

4.3.1 ros 主从机配置

① 主机名查询确认

通过终端命令：\$ hostname #查询电脑主机名

个人笔记本主机名：zdh-ThinkPad-L480（戴尔电脑不同）

机器人电脑主机名：DashgoE1（通过访问机器人电脑查询）

② IP 查询确认

通过终端命令：\$ ifconfig -a #查询 IP

个人笔记本 IP：192.168.31.XXX

机器人电脑 IP：192.168.31.200

③ 在两台电脑的/etc/hosts 文件增加：（IP 和主机名中间是 tab，不是空格）

在个人笔记本 hosts 文件添加机器人电脑的信息：192.168.31.200

DashgoE1

在机器人电脑 hosts 文件添加个人笔记本的信息：192.168.31.XXX

zdh-ThinkPad-L480

④ 在两台电脑的 ~/.bashrc 中增加主机电脑（也就是机器人电脑）的

ros_master

```
export ROS_MASTER_URI=http:// 192.168.31.200:11311
```

```
source ~/.bashrc
```

⑤ 测试

在主机（机器人电脑）启动 roscore，在从机（个人笔记本）输入

rosparam list，查看是否能获取所有 rosparam

4.3.2 ros 通过 opencv 实时读取 ZED 相机图像

在第 2 步中仅仅通过 ros 的工具查看了图像，但是实验过程中我们要通过程序对图像进行数字处理，所以需要在程序中实现对图像的操作。

① 新建工作空间

```
$ mkdir -p /catkin_ws/src  
$ cd /catkin_ws/  
$ catkin_make
```

② 创建新的功能包并编译

```
$ cd catkin_ws/  
$ cd src/  
$ catkin_create_pkg image_pkg  
$ cd ..  
$ catkin_make  
$ source devel/setup.bash
```

③ 新建读取图像的 cpp 文件

进入功能包，创建 src 目录和 cpp 文件

```
$ mkdir src && cd src  
$ gedit subscriber.cpp
```

创建代码

④ 配置 CMakeList.txt 和 package.xml

⑤ 编译

⑥ 测试

主机启动 roscore，从机启动相机（第 2 步骤中），此时不再使用 ros 工具查看图片。通过编写的 subscriber.cpp 所生成的可执行文件来查看图片，以后图像处理的代码均可在 subscriber.cpp 中完成。

4.4 实验内容

4.4.1 操作步骤

① 打开终端，输入命令：\$ ssh eaibot@192.168.31.200

密码：eaibot（后文使用到将不再赘述）

通过该终端窗口即可远程访问机器人电脑。

```
$ roslaunch dashgo_driver driver.launch
```

② 新建终端

启动 image_pkg 包下的 subscriber 可执行文件。

```
$ rosrun image_pkg subscriber
```

4.4.2 直方图均衡化算法实现（按照要求自己编写程序）

在 subscriber.cpp 中添加如下功能程序：

- ① 统计每个灰度下的像素个数
- ② 统计灰度频率
- ③ 计算累计密度
- ④ 重新计算均衡化后的灰度值，四舍五入。参考公式： $(N-1)*T+0.5$
- ⑤ 直方图均衡化,更新原图每个点的像素值

附录: subscriber.cpp

```
#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;

int main(int argc, char **argv)
{
    VideoCapture capture;
    capture.open(1);//打开 zed 相机

    ROS_WARN("*****START");
    ros::init(argc,argv,"trafficLaneTrack");//初始化 ROS 节点
    ros::NodeHandle n;

    // ros::Rate loop_rate(10);//定义速度发布频率
    ros::Publisher pub =
n.advertise<geometry_msgs::Twist>("/smoother_cmd_vel", 5);//定义速度发布者

    if (!capture.isOpened())
    {
        printf("摄像头没有正常打开，重新插拔工控机上摄像头\n");
        return 0;
    }
    waitKey(1000);
    Mat frame;//当前帧图片
    int nFrames = 0;//图片帧数
    int frameWidth = capture.get(CV_CAP_PROP_FRAME_WIDTH);//图片宽
    int frameHeight = capture.get(CV_CAP_PROP_FRAME_HEIGHT);//图片高
```

```

while (ros::ok())
{
    capture.read(frame);
    if(frame.empty())
    {
        break;
    }

    Mat frIn = frame(cv::Rect(0, 0, frame.cols / 2, frame.rows)); //截取 zed 的左目
    图片

    // 此处增加直方图均衡化

    imshow("1",frIn);

    geometry_msgs::Twist cmd_red;

    // 车的速度值设置
    cmd_red.linear.x = LINEAR_X;
    cmd_red.linear.y = 0;
    cmd_red.linear.z = 0;
    cmd_red.angular.x = 0;
    cmd_red.angular.y = 0;
    cmd_red.angular.z = 0.2;

    pub.publish(cmd_red);

    ros::spinOnce();
    // loop_rate.sleep();
    waitKey(5);

}

return 0;
}

```