```
/*****************************************************************************************************************************
练习1：利用canny算子进行边缘提取
*****************************************************************************************************************************/

void canny_extracrtion()
{

//打开摄像头
VideoCapture cap;
cap.open(0);

//判断摄像头是否打开
if (!cap.isOpened())
{
cout << "不能打开视频文件" << endl;
return;

}

//读取当前帧照片
Mat frame;
cap.read(frame);


//定义图片容器
Mat dx;
Mat dy;
Mat canny_Mat1;
Mat canny_Mat2;
Mat gry_Mat;


//通过sobel算子得到x、y方向上的梯度
/************************************************
sobel（） 函数模型：
sobel( src,dx/dy,CV_16SC1,1,0,3);

参数介绍：
. src：Mat类的输入图像
. dx/dy：x/y方向上的梯度
. CV_16SC1：输出的格式
. 1，0表示对x或者对y方向求微分的次数
. 3表示sobel核的大小
************************************************/
Sobel(frame, dx, CV_16S, 1, 0, 3);
Sobel(frame, dy, CV_16S, 0, 1, 3);

//进行sobel边缘提取
/************************************************
Canny（） 函数模型：
Canny(InputArray dx, InputArray dy,
     OutputArray edges,
         double threshold1, double threshold2,
 bool L2gradient = false);

参数介绍：
. InputArray dx/InputArray dy：输入图像x/y方向上的梯度
. dx/dy：x/y方向上的梯度
. OutputArray edges：输出边缘图像，单通道，8bit；
. double threshold1：表示阈值1
. double threshold2：表示阈值2
. bool L2gradient:是否采用更精确的方式计算图像梯度
************************************************/

Canny(dx, dy, canny_Mat1, 20, 60);

//转化为灰度图，在进行边缘提取
cvtColor(frame, gry_Mat, COLOR_BGR2GRAY);

/************************************************
Canny（） 函数模型：
Canny( InputArray image, OutputArray edges,
                         double threshold1, double threshold2,
                         int apertureSize = 3, bool L2gradient = false );

参数介绍：
. InputArray image：输入图像，单通道8位图像
. OutputArray edges：输出边缘图像，单通道，8bit；
. double threshold1：表示阈值1
. double threshold2：表示阈值2
. int apertureSize：Sobel算子大小
. bool L2gradient：是否采用更精确的方式计算图像梯度
************************************************/
Canny(gry_Mat,canny_Mat2, 20, 60);

imshow("canny_Mat2", canny_Mat2);
imshow("canny_Mat1", canny_Mat1);

waitKey(30);

}
```