哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

机器视觉基础实践

授课人：刘改霞

办公地址：K103

课程概述

学时	8学时
上课地点	实训楼K217
最多容纳人数	16人
课程内容	基于OpenCV的视觉识别基础
	任务一：不同颜色小球识别
	RoboMaster EP运动控制
	任务二：不同颜色小球跟踪追随

考核方式

考核环节	考核内容及评价细则	所占分值
工程认知	项目报告	15%
工程素养	1. 考勤 2. 5S管理规范（整理、整顿、清扫、清洁和素养）执行情况	15%
工程技能	根据课堂任务完成程度给出	30%
工程综合	/	/
工程创新	/	/

哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

机器视觉基础实践

第一讲:基于OpenCV的视觉识别基础

授课人：刘改霞

办公地址：K103

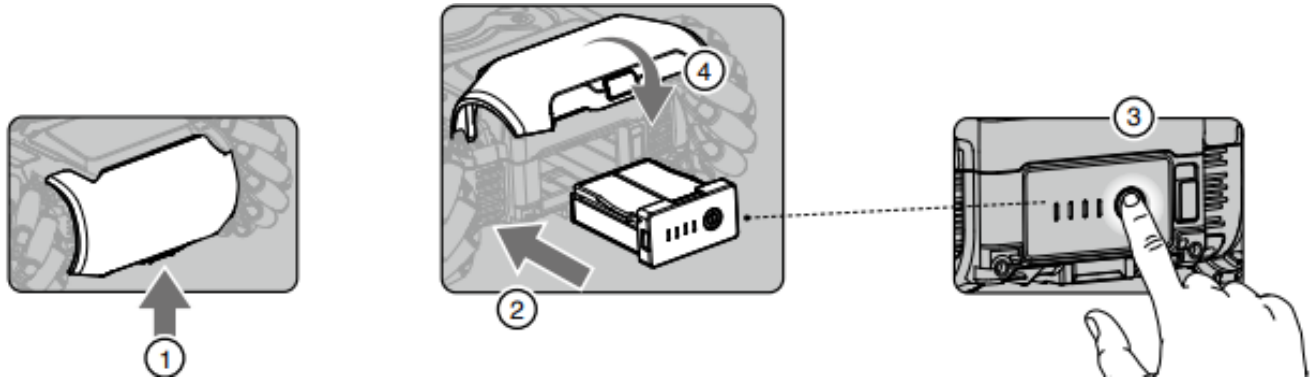
识别特定颜色小球

识别特定颜色的小球

- 图像获取
- 颜色识别
- 特征筛选

如何获取图像？

1. 开启Robomaster机器人



2. 建立Rbomaster机器人和智能设备的连接（支持wifi直连（推荐）、wifi组网、USB直连三种模式）

- 开启机器人电源，切换智能中控的连接模式开关至直连模式
- 打开电脑的无线网络访问列表，选择位于机身智能中控贴纸上对应的WiFi名称，输入8位密码，选择连接

图像获取

如何获取图像？

3. 通过Robomaster SDK调用相机模块获取视频流及图像

```
from robomaster import robot
import cv2


if __name__ == '__main__':
    ep_robot = robot.Robot()
    ep_robot.initialize(conn_type="ap")

    ep_camera = ep_robot.camera
    # 每次获取最新的1帧图像显示，并停留1秒
    ep_camera.start_video_stream(display=False, resolution='360p')
    for i in range(0, 200):
        img = ep_camera.read_cv2_image()
        cv2.imshow("Robot", img)
        cv2.waitKey(1)
    cv2.destroyAllWindows()
    ep_camera.stop_video_stream()

    ep_robot.close()
```

从第三方robomaster模块中引入robot类或函数

引入第三方opencv模块

创建Robot类的实例对象

初始化机器人，conn_type指定机器人连接模式，ap表示WiFi直连模式，sta表示wifi组网模式，rndis使用USB连接

获取camera模块对象

开始获取视频流，display参数指定是否显示获取到的视频流，resolution参数指定视频的尺寸大小，支持360p，540p，720p。

循环读取200帧图像数据

调用opencv中imshow方法在“Robot”窗口中显示图像

调用opencv中waitKey刷新图像显示，参数为延迟时间ms

销毁所有窗口

停止获取视频流

释放机器人对象相关资源

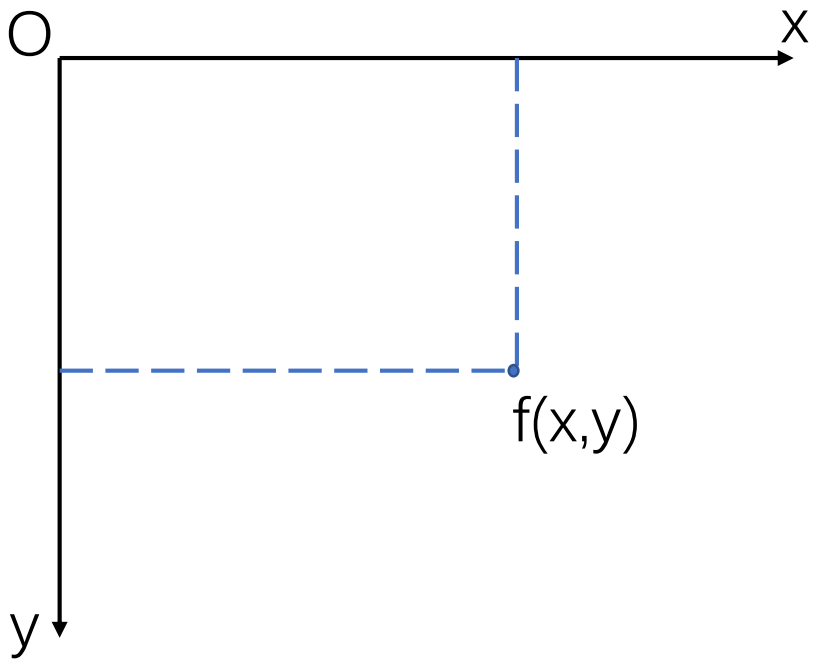
Robomaster SDK安装及相关说明:

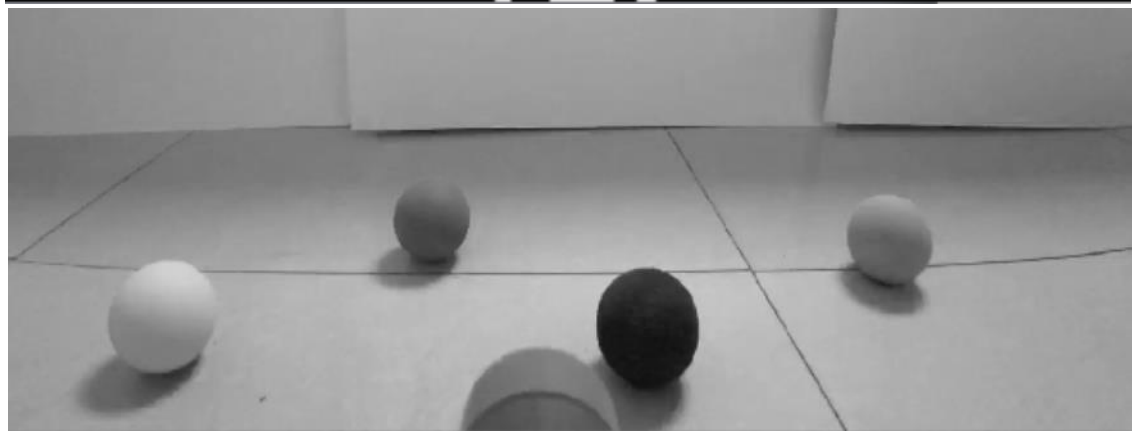
https://robomaster-dev.readthedocs.io/zh_CN/latest/python_sdk/installs.html#sdk-windows

Robomaster SDK例程:

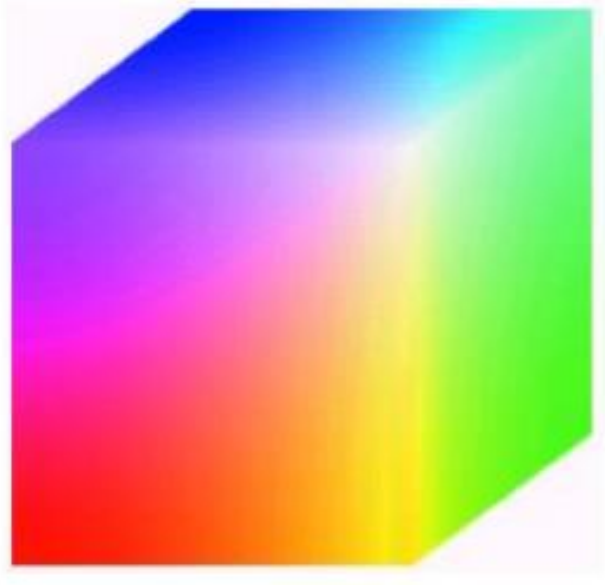
https://github.com/dji-sdk/RoboMaster-SDK/tree/master/examples

数字图像





彩色图像——RGB颜色空间

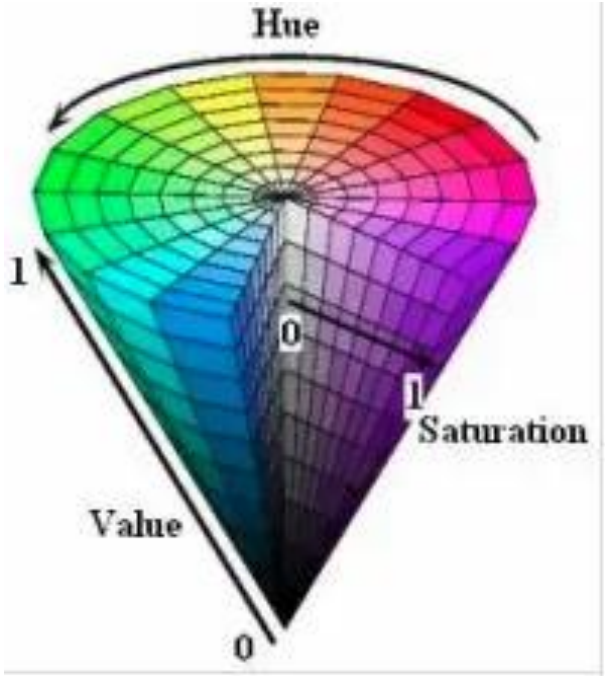


色彩是人的眼睛对于不同频率的光线的不同感受，色彩既是客观存在的（不同频率的光）又是主观感知的，有认识差异。

RGB颜色空间： R(Red:红)、G(Green:绿)、B(Blue:蓝)三种基本色为基础，进行不同程度的叠加，产生丰富而广泛的颜色，所以俗称三基色模式。

- 从RGB值中很难知道该颜色大致属性。
- RGB颜色空间中R、G、B三个值是高度相关的（B-R：0.78，R-G：0.98，G-B：0.94）这些高相关性的存在，对图像中的颜色处理形成了挑战。

彩色图像——HSV颜色空间



Hue——色调，表示色彩信息，即所处的光谱颜色的位置。

Saturation——饱和度，表示所选颜色的纯度和该颜色做大纯度之间的比例

Value——明度，表示色彩的明亮度

HSV将明亮度信息从彩色中分解出来，而色调和饱和度与人类感知是相对应的，

HSV比RGB模型更接近人们对色彩的感知，在处理颜色识别时应用较多。

	黑	灰	白	红		橙	黄	绿	青	蓝	紫
Hmin	0	0	0	0	156	11	26	35	78	100	125
Hmax	180	180	180	10	180	25	34	77	99	124	155
Smin	0	0	0	43		43	43	43	43	43	43
Smax	255	43	30	255		255	255	255	255	255	255
Vmin	0	46	221	46		46	46	46	46	46	46
Vmax	46	220	255	255		255	255	255	255	255	255

彩色图像——颜色空间的转换

cv2.cvtColor(src, code)

src：需要处理的图像

code：颜色空间的转换方式，如COLOR_BGR2HSV、COLOR_BGR2GRAY

函数返回处理完后所得到的hsv图像

课堂小任务

2.将机器人获取到的bgr图像转换成hsv图像并显示在窗口（窗口可命名为hsv）中

RGB to HSV conversion formula

The R,G,B values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$Cmax = \max(R', G', B')$$

$$Cmin = \min(R', G', B')$$

$$\Delta = Cmax - Cmin$$

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

$$V = Cmax$$

彩色图像分割

cv2.inRange(src, lowerb, upperb)

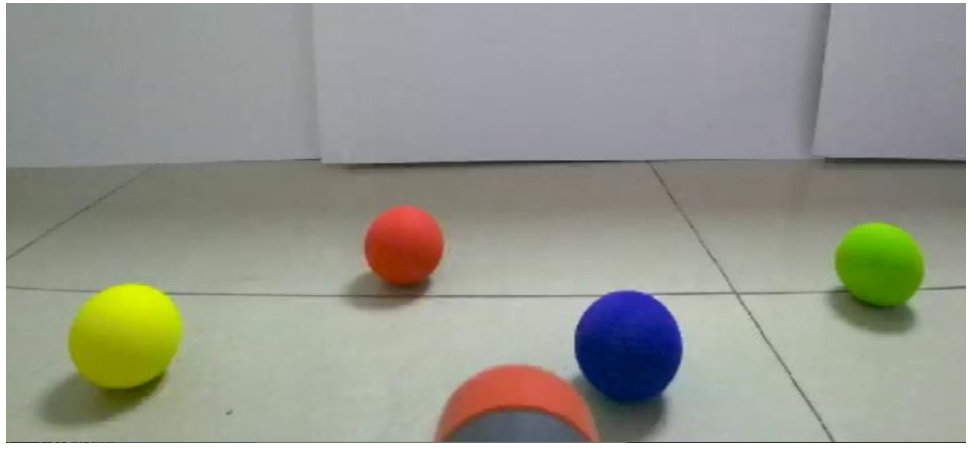
检查数组元素是否在另外两个数组元素值之间,返回处理完后得到的二值图像

彩色图像某个像素点各通道的值都位于lowerb和upperb相应通道所对应的值之间时,该像素点的输出值将被置为255，否则将被置为0

src:输入要处理的图像,可以为单通道或多通道

lowerb:下边界数组或标量，如（lower_h, lower_s, lower_v）

upperb:上边界数组或标量,如（upper_h, upper_s,upper_v）



课堂小任务

3.运行基于RGB颜色模型进行图像分割的示例程序bgr_segmentation.py,编写基于HSV颜色模型进行图像分割程序,比较基于RGB与基于HSV进行图像分割的效果,获取目标颜色HSV阈值，并进行记录

（包括：红色——[lower_h,lower_s,lower_v],[upper_h,upper_s,upper_v]

绿色——[lower_h,lower_s,lower_v],[upper_h,upper_s,upper_v]

蓝色——[lower_h,lower_s,lower_v],[upper_h,upper_s,upper_v]

黄色——[lower_h,lower_s,lower_v],[upper_h,upper_s,upper_v]）

4. 通过Robomaster SDK调用相机模块获取视频流，根据所得到的目标颜色HSV阈值，对视频流中图像帧进行处理获得二值图，并将其显示在窗口中，二值图中感兴趣小球应显示为白色，其他不感兴趣区域应显示为黑色。

课堂小任务

4. 通过Robomaster SDK调用相机模块获取视频流，根据所得到的目标颜色HSV阈值，对视频流中图像帧进行处理获得二值图，并将其显示在窗口中，二值图中感兴趣小球应显示为白色，其他不感兴趣区域应显示为黑色。

```
img_binary = cv2.inRange(img_hsv, (lower_h,lower_s,lower_v), (upper_h,upper_s,upper_v))
```

颜色识别

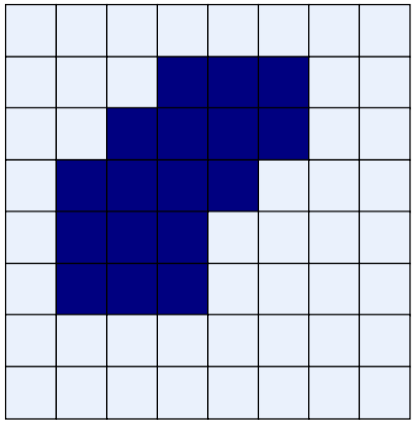
形态学常用的方法有膨胀、腐蚀、开运算、闭运算等，大多以膨胀和腐蚀为基础操作

膨胀：目标像素的值替换为结构元素覆盖区域的局部最大值，可使二值图中非零区域（白色区域）扩张，同时填充凹面

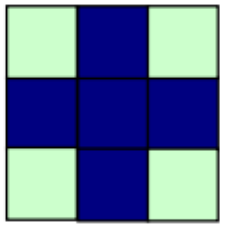
腐蚀：目标像素的值替换为结构元素覆盖区域的局部最小值，可使二值图中非零区域（白色区域）缩减，同时消除突起

开运算：先将图像进行腐蚀，然后对腐蚀的结果进行膨胀，可用于去除孤立的小点、毛刺等

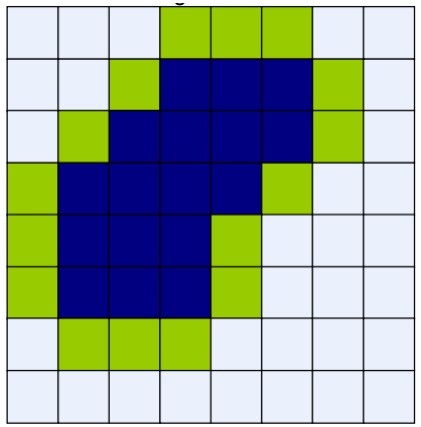
闭运算：先将图像进行膨胀，然后对膨胀的结果进行腐蚀，可用于填平小孔、弥合缝隙



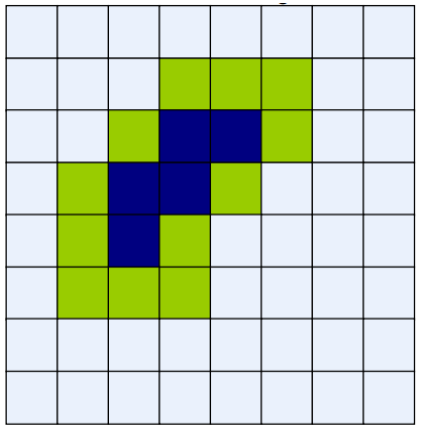
原图像



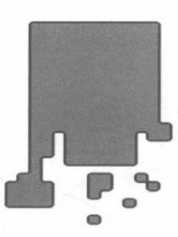
结构元素



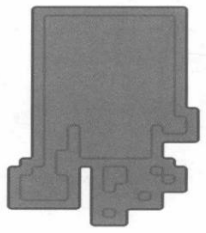
膨胀运算



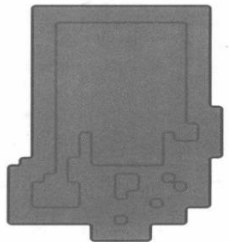
腐蚀运算



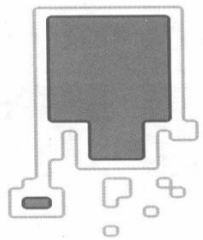
原图



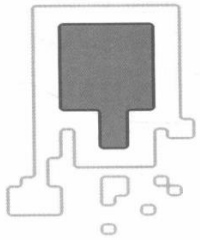
膨胀



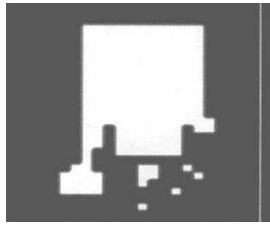
二次膨胀



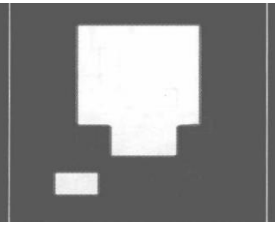
腐蚀



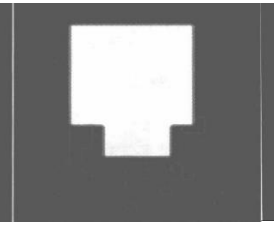
二次腐蚀



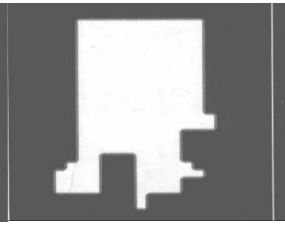
源图像



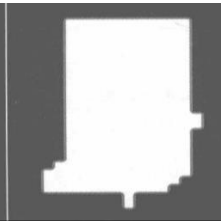
开运算



二次开运算



闭运算



二次闭运算

颜色识别

Opencv通用形态学函数：

cv2.morphologyEx(src, op, kernel,…)

src：待处理的原图像

op：形态操作类型,有MORPH_ERODE（腐蚀）, MORPH_DILATE（膨胀）, MORPH_OPEN（开运算）, MORPH_CLOSE （闭运算）等

Kernel:结构元素

Iterations：执行次数

返回经过形态学处理之后的图像

```
kernel = np.ones((3,3), np.uint8)  # 定义3*3全1结构元素
# 调用通用形态学处理函数morphologyEx对二值图进行处理，返回形态处理后得到的二值图
# op可取cv2.MORPH_ERODE(腐蚀), cv2.MORPH_DILATE(膨胀), cv2.MORPH_OPEN(开运算), cv2.MORPH_CLOSE(闭运算)
closing = cv2.morphologyEx(hsv_divide_Binary,op=cv2.MORPH_CLOSE,kernel=kernel, iterations=2)
```

课堂小任务

5. 利用形态学处理方法对经阈值分割后获取到的二值图像进行腐蚀、膨胀、开运算、闭运算等形态运算操作，仔细观察输出图像变化，分析膨胀、腐蚀、开运算、闭运算等形态学操作对图像造成的影响。

并通过合适的形态学操作，去除背景中的孤立点、毛刺等，并填充目标物体区域内的空洞，方便后续对目标物体区域的定位。

轮廓特征

轮廓可以简单地认为是连续的点（沿边界）连在一起的曲线。轮廓在形状分析和物体检测中比较有用。

查找轮廓：cv.findContours(image, mode, method, …)

image：待处理的图像，二值图像

mode：图像检索模式，**RETR_EXTERNAL**—只检测外轮廓，RETR_LIST—检测的轮廓不建立等级关系，RETR_CCOMP—建立两个等级的轮廓，上面一层为外边界，里面一层为内孔的边界信息，RETR_TREE—建立一个等级树结构的轮廓

method：轮廓近似方法，CHAIN_APPROX_NONE—存储所有边界点，**CHAIN_APPROX_SIMPLE**:压缩水平、垂直、对角线方向，只保留端点，这一操作将大大减少返回的点数

返回counters，hierarchy 其中counters为检测到的轮廓数组，其中每一个轮廓都是由点的数组组成，hierarchy为各轮廓之间的层级关系

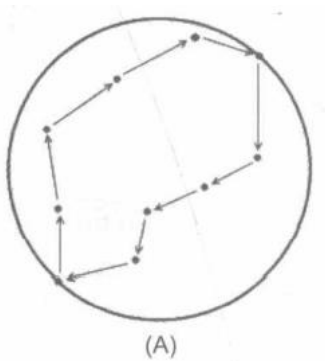
轮廓特征：

cv2.contourArea(points)->retval轮廓的面积

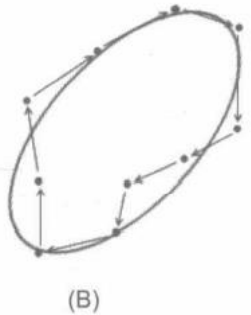
cv2.arcLength(points, closed) ->retval轮廓的周长

cv2.minEnclosingCircle(points) ->center, radius 轮廓最小外包圆

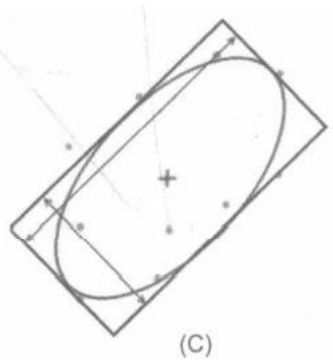
cv2.fitEllipse(points) ->center, size, angle 对轮廓进行椭圆拟合



10个点的最小外包圆



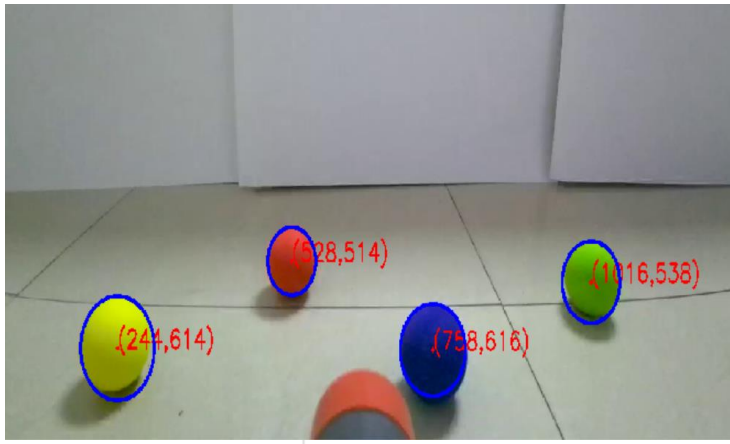
最佳拟合椭圆



拟合椭圆返回参数示例

课堂小任务

6. 在课堂小任务5获得的二值图中进行轮廓查找，通过轮廓特征筛选，如轮廓面积、轮廓周长、轮廓最小外包圆、拟合椭圆特征等，对目标小球位置进行定位，输出得到目标小球中心点位置信息。



二值图的轮廓查找及绘制可参考如下代码：

```
# 在二值图中查找轮廓
contours, hierarchy = cv2.findContours(morphology_img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
for idx, contour in enumerate(contours):  # 遍历查找到的轮廓
    area = cv2.contourArea(contours[idx])  # 计算每个轮廓的面积
    box = center_idx, size_idx, angle_idx = cv2.fitEllipse(contours[idx])  # 对每个轮廓进行椭圆拟合
    #  如果拟合到的椭圆长轴短轴比小于1.3，则认为是圆形
    if max(size_idx[0], size_idx[1]) < min(size_idx[0], size_idx[1]) * 1.3:
        center_idx = np.uint16(center_idx)  # 转化为整型
        size_idx = np.uint16(size_idx)  # 转化为整型
        cv2.drawContours(img_bgr, contours, idx, (0, 0, 255), 3)  # 绘制检测到的且符合条件的轮廓
        cv2.ellipse(img_bgr, box, (255, 0, 0), 2)  # 在img_bgr中画出拟合得到的椭圆，颜色为（255，0，0），线粗为2
        cv2.circle(img_bgr, center_idx, 1, (0, 0, 255), 3)  # 在img_bgr中画出检测到的圆形圆心，颜色为（0，0，255），线粗为2
        # 在img_bgr图像中center_idx位置添加字符串 ' (%d, %d)' % center_idx—为格式化输出字符串，
        # 字体为cv2.FONT_HERSHEY_SIMPLEX，颜色为(0, 0, 255)，线粗为2
        cv2.putText(img_bgr, '(%d, %d)' % (center_idx[0], center_idx[1]), center_idx, cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
        diameter = np.mean(size_idx)  # 设置圆形直径为椭圆长轴短轴均值
        print(diameter)  # 打印检测到的圆形直径
```

课堂任务

编写程序,通过Robomaster EP摄像头获取视频流,实现对相机视野范围内特定颜色的小球的实时检测

要求及说明:

1. 提供红色、黄色、蓝色、绿色四种颜色的小球，每组成员可任选一种或多种颜色的小球进行实验
2. 在视频流中实时显示检测到的小球轮廓
3. 若相机视野范围内有多个指定颜色的小球，要求只检测距离机器人最近的一个指定颜色小球

哈爾濱工業大學(深圳)

HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

机器视觉基础实践

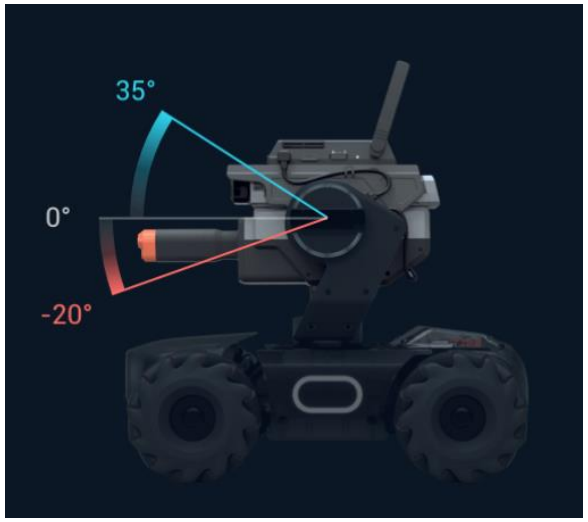
第二讲：Robomaster EP控制

授课人：刘改霞

办公地址：K103

Robomaster EP运动控制

底盘



云台

机械臂及机械爪

Robomaster EP运动控制

➢ 底盘速度控制（即时动作）

ep_chassis = ep_robot.chassis

ep_chassis.drive_speed(x, y, z, timeout) #设置底盘速度，立即生效

x：float[-3.5, 3.5], 指定x轴向运动速度，x正时，前进，x负时，后退

y：float[-3.5, 3.5], 指定y轴向运动速度，y正时，右移，y负时，左移

z：float[-600, 600], 指定z轴向运动速度即旋转速度，y正时，右旋，y负时，左旋

Timeout：float[0, inf]，超过指定时间内未收到脉轮转速指令，主动控制机器人停止，单位s

➢ 底盘距离控制（任务动作）

ep_chassis.move(x, y, z, xy_speed, z_speed).wait_for_completed() #控制底盘运动到指定位置，坐标轴原点为当前位置

调用该方法，程序会阻塞在该语句，直至动作执行完毕或执行超时

X：float[-5, 5], x轴向运动距离，单位m

Y：float[-5, 5], y轴向运动距离，单位m

z：float[-1800, 1800]，z轴向旋转角度，单位°

Xy_speed：float[0.5, 2]，xy轴向运动速度，单位m/s

Z_speed：float[10, 540]，z轴向旋转速度，单位°/s

严禁全速冲撞硬度较大的物体，如墙壁等。

Robomaster EP运动控制

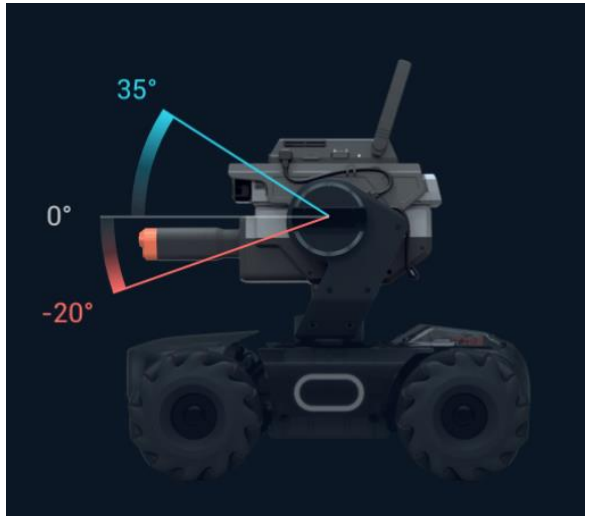
➢ 云台速度控制（即时动作）

ep_gimbal = ep_robot.gimbal

ep_gimbal.drive_speed(pitch_speed, yaw_speed) #设置云台转动速度，立即生效

pitch_speed： float[-360, 360], 指定云台俯仰轴向运动速度°/s，正时，上仰，x负时，下俯

yaw_speed： float [-360, 360], 指定云台航向轴向运动速度°/s，正时，右转，y负时，左转



➢ 云台距离控制（任务动作）

ep_ gimbal.moveto(pitch=0, yaw=0, pitch_speed=30, yaw_speed=30).wait_for_completed() #控制云台运动到指定位置，坐标轴原点为上电位置

调用该方法，程序会阻塞在该语句，直至动作执行完毕或执行超时

pitch：float: [-25, 30]，pitch 轴角度，单位 °，正时，上仰，x负时，下俯

yaw：float: [-250, 250]，yaw 轴角度，单位 °，正时，右转，y负时，左转

pitch_speed：float: [0, 540]，pitch 轴运动速速，单位 °/s

yaw_speed：float: [0, 540]，yaw 轴运动速度，单位 °/s

Robomaster EP运动控制

➢ 设置整机运动模式

ep_robot.set_robot_mode(mode)

可选模式有:

自由模式（FREE）：云台和底盘运动分离，互不影响

云台跟随底盘模式（CHASSIS_LEAD）：云台始终跟随底盘沿航向轴旋转

底盘跟随云台模式（GIMBAL_LEAD）: 底盘跟随云台绕航向轴旋转

如：ep_robot.set_robot_mode(mode= ‘CHASSIS_LEAD’)

课堂小任务1:

编写程序，控制机器人进行跳舞，舞姿自定义

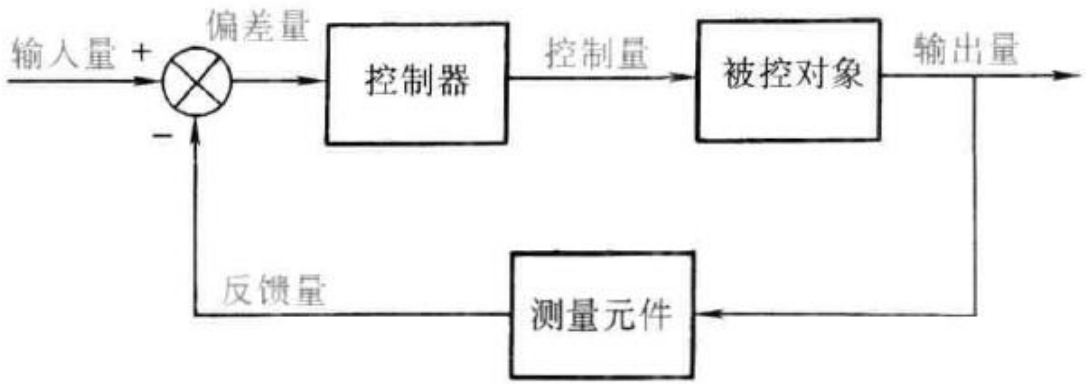
要求：需使用到前面介绍的所有接口

所谓控制就是通过施加适当的输入，使被控对象中可变的物理量按照希望的方式保持或加以改变，以此来得到期望的输出。

- 被控对象：需要被控制的客观对象
- 被控量：被控对象中可变的属性或物理量
- 期望值：被控量的期望值

➢ 闭环控制

将检测出来的输出量送回到系统的输入端，并与输入信号比较的过程称为反馈。输入信号与反馈信号之差，称为偏差信号，偏差信号作用于控制器上，控制器对偏差信号进行某种运算，产生一个控制作用，使系统的输出量趋于给定的数值。闭环控制的实质，就是利用负反馈的作用来减小系统的误差。



闭环控制系统

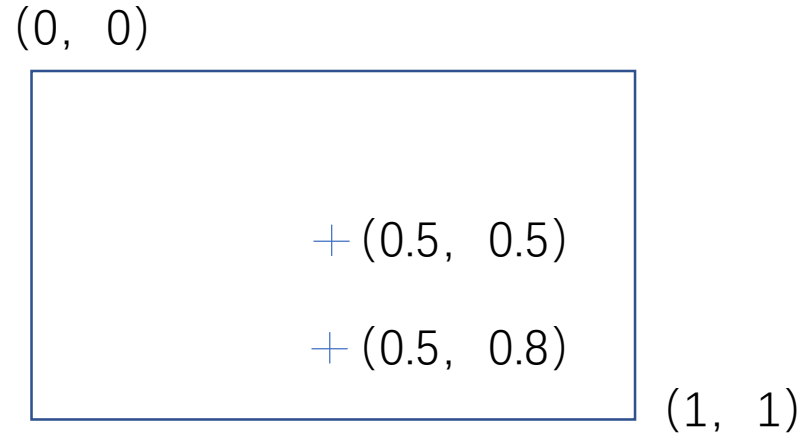
特定颜色小球跟随

控制底盘跟随小球方法尝试：

被控对象：底盘

被控量：底盘x以及y轴向运动速度 或 底盘x以及z轴向运动速度

期望值：根据实际情况进行确定（如（0.5，0.8））



① 图片归一化，x = x/img_width，y=y/img_height，其中img_width,img_heigh,img_depth = img.shape

② 获取小球中心坐标（x，y）

③ 为了让小球中心（x,y）靠近期望值，如（0.5，0.8），需控制底盘左右移动及前后移动

④ 将小球中心横坐标与期望值的偏差（x-0.5）转换为对底盘横向移动速度的控制量，当小球位于期望位置的左侧时，(x-0.5)为负值，底盘左移，当小球位于期望位置的右侧时，(x-0.5)为正值，底盘右移,且偏差越大，移动速度越大，偏差越小，移动速度越小，偏差为零，速度为零，不再进行移动，符合期望。同样的，可将小球中心纵坐标与期望值的偏差（y-0.8）转换为对底盘纵向移动速度的控制量。

⑤实际上，即使小球位于视野边缘，控制量最大才为0.5，控制量太小，移动速度太慢，为了加快反应速度，可给控制量乘上一个系数Kp进行成比例放大，如，Kpx*（ x-0.5 ），Kpy*（0.8-y），调整Kpx，Kpy值，值到底盘跟随效果较好。

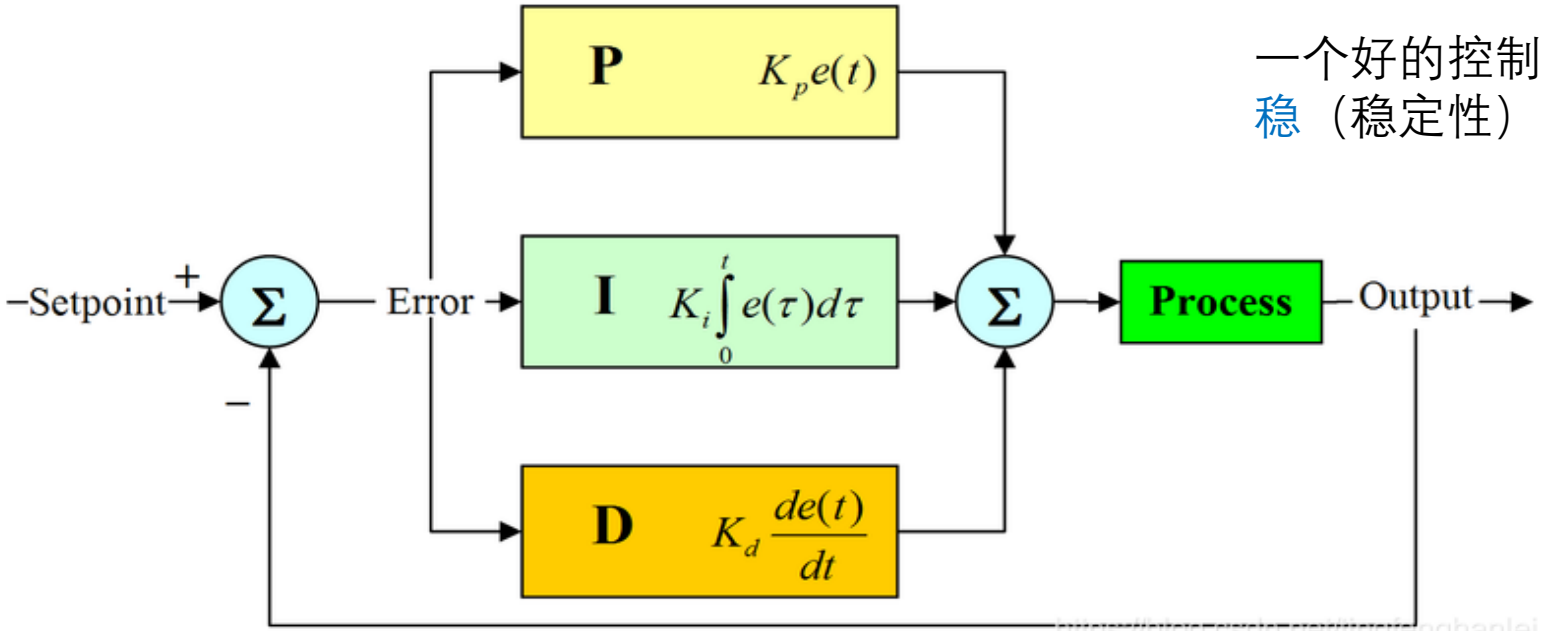
特定颜色小球跟随

课堂任务：

基于上节课实现的Robomaster EP对相机视野范围内特定颜色小球的实时检测功能，利用所讲控制方法（比例控制）控制底盘或云台实现对特定颜色小球跟踪追随。

PID控制

PID控制是最早发展起来的控制策略之一，它结构简单、稳定性好、工作可靠、调整方便，成为工业控制领域应用最为广泛的核心控制器和控制技术。



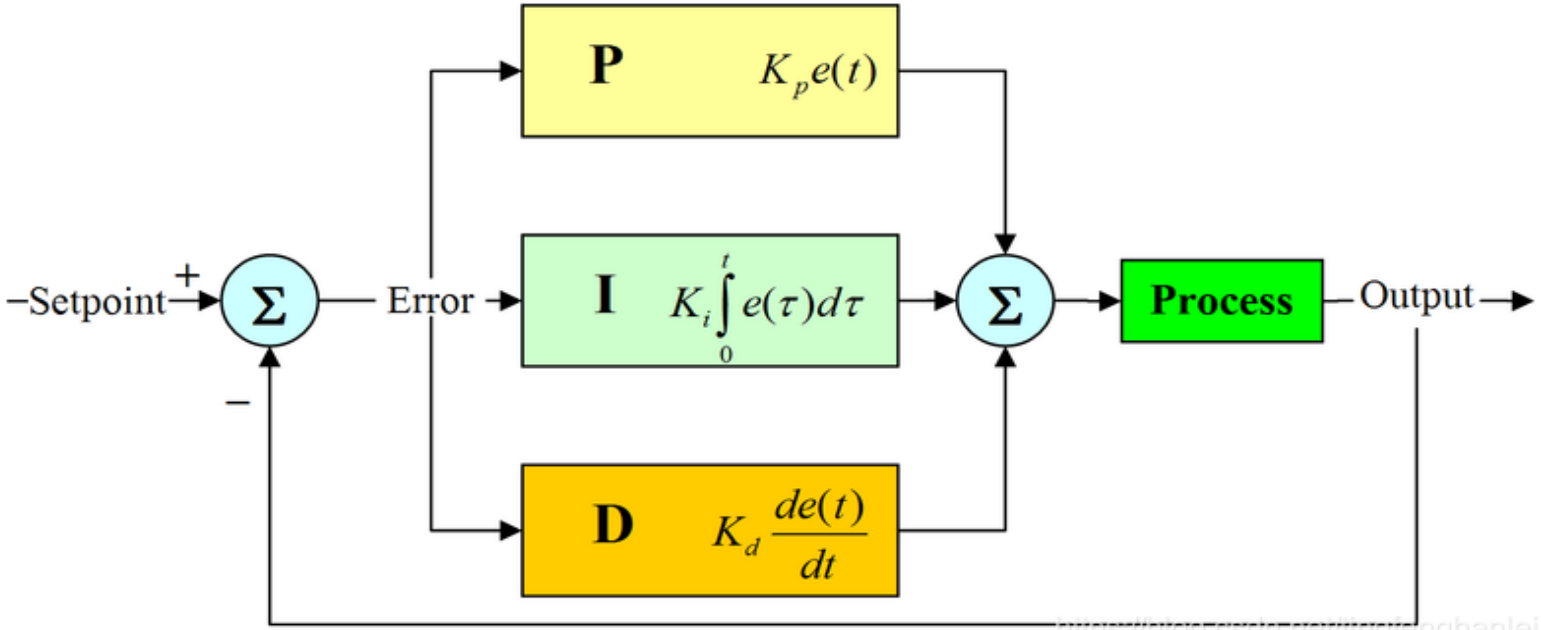
一个好的控制系统具备的三个基本特征：
稳（稳定性）、快（快速性）、准（准确性）

比例：控制器输入信号的当前状态——快速作用于输出

积分：控制器输入信号的历史变换——消除过去的累积误差

微分：控制器输入信号的未来变化趋势——具有超前控制作用

PID控制



$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

数字化

$$u(k) = K_p\, e(k) + K_i \sum_{j=0}^{k} e(j) + K_d[e(k) - e(k-1)]$$

→ 位置式PID

$$\Delta u(k) = u(k) - u(k-1) = K_p\,[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)]$$

$$u(k) = u(k-1) + K_p\,[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)]$$

→ 增量式PID

PID控制

比例系数Kp对系统性能的影响：

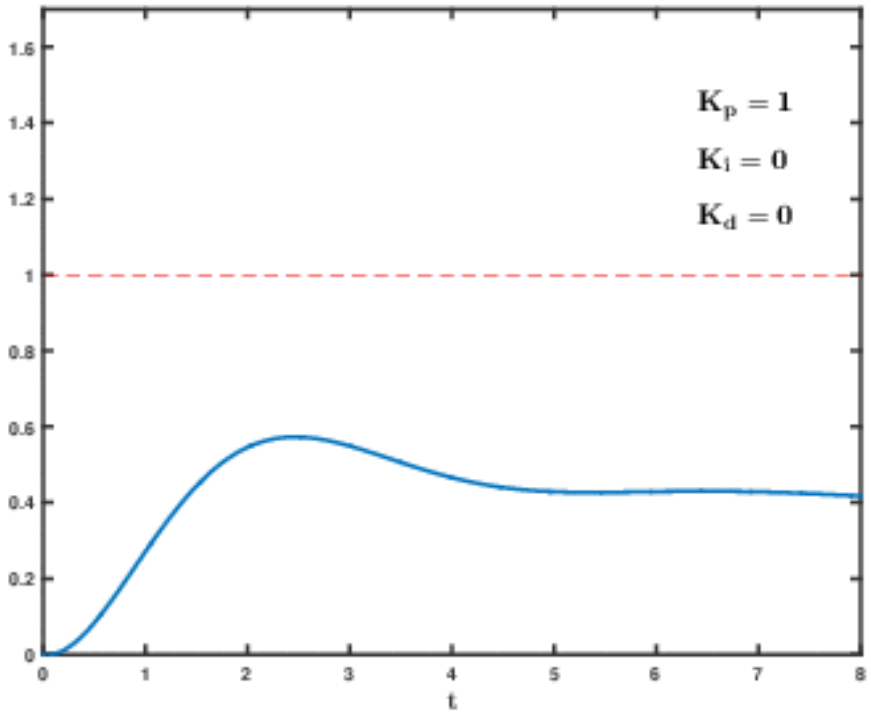
- 对系统静态性能的影响：在系统稳定的情况下，Kp增加，系统稳态误差将减小，可调高控制精度。
- 对系统动态性能的影响：Kp增加，系统响应加快；如果Kp偏大，震荡次数增多，调节时间加长，如果Kp过大，系统趋于不稳定。

积分系数Ki对系统性能的影响

- 对系统静态性能的影响：Ki增大，积分作用加强，有利于消除静差，如果Ki太小，系统将不能消除静差。
- 对系统动态性能的影响：Ki减小，削弱积分作用，有利于减小超调量，克服震荡；如果Ki太大，将导致超调量过大，甚至产生震荡，系统将不稳定。

微分系统Kd对系统性能的影响

- Kd增加，系统响应加快，超调量减小，增加系统的稳定性，但系统对干扰的敏感性增加，对干扰的抑制能力减弱。



PID控制

- PID参数整定——找出一组最佳Kp、Ki、Kd调节参数

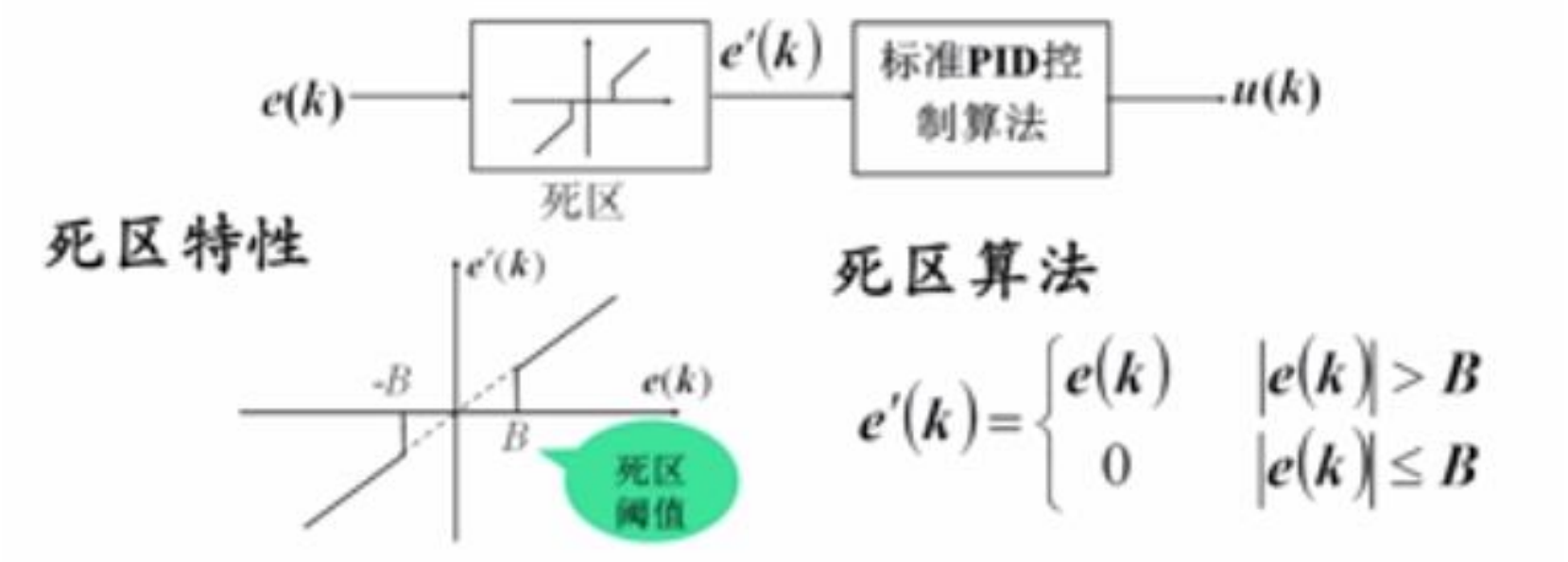
试凑法

——根据PID各个参数变化对系统性能的影响，按照先比例、后积分、再微分的步骤进行整定

- 只采用比例控制，Kp由小变大，若响应时间、超调、静差已达到要求，只采用比例调节即可;
- 若静差不满足要求，则加入积分控制，将Kp减小，例如取0.8Kp代替Kp，Ki由小变大，反复修改Kp和Ki值，力争在消除静差的前提下，得到满意的控制效果
- 若动态性能不满足设计要求（超调量过大或调节时间过长），则加入微分控制，Kd由小到大，同时改变Kp和Ki的值，直到得到满意的控制效果。

PID控制

- 带死区的PID控制算法



死区阈值:

按照控制系统的控制精度要求来设计，根据控制精度，达到控制精度后，进入死区，控制器输出保持状态

适可而止

特定颜色小球跟随

课堂小任务3:

编写PID控制算法，对PID参数进行整定，实现底盘或云台对指定颜色小球的跟踪追随。

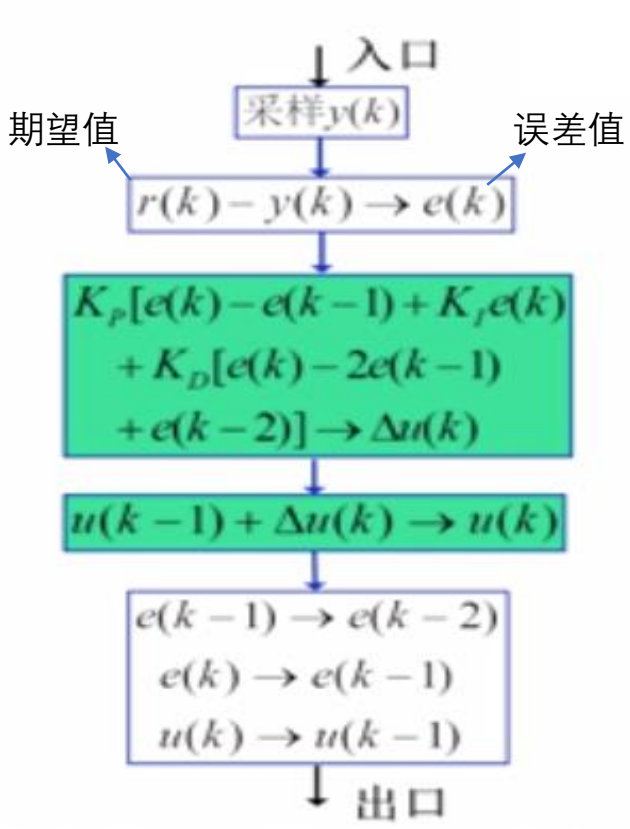
PID控制

◆ 位置式PID

$$u(k) = K_p\, e(k) + K_i \sum_{j=0}^{k} e(j) + K_d[e(k) - e(k-1)]$$

◆ 增量式PID

$$u(k) = u(k-1) + K_p\,[e(k) - e(k-1)]$$
$$+K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)]$$



增量式PID算法流程