

Air-Hockey Robot

文件结构

***Material** (文件夹) : 放图片等其他材料

***module** (文件夹) : 放主要函数库, func.py (生成GUI以及图像显示的主要函数), Hockey.py (定义桌子、手柄和球类以及相关的图像预处理算法函数), gui_module.py (基于func.py的GUI函数库, 包括透视变换、球和手柄的追踪), Strategy.py (策略函数, 包括生成运动的主要函数)

***save** (文件夹) : 放参数调整的存档

***XYZ-GUI.py**: 主界面函数, 包括串口通信和图像处理接口

基本类定义

桌子 (Desk)

```
class Desk:
    def __init__(self, id=-1):
        self.id = id # 摄像头编号
        self.frame = None # 帧
        self.capture = None # 视频流
        self.corner_points = {0: (0, 0), 1: (0, 0), 2: (0, 0), 3: (0, 0)} # 角点字典
        self.frame_transformed = None # 变换后的帧
```

球 (Ball)

```
class Ball:
    def __init__(self, frame=None):
        self.frame_original = frame # 初始图像
        self.frame_segmentation = None # 颜色分割后的图像
        self.frame_thresh = None # 二值化后的图像
        self.frame_preprocess = None # 预处理后的图像
        self.frame_locate = None # 目标定位标定
        self.frame_track = None # 目标追踪处理后的图像
        self.radius = 0 # 目标外接球半径
        self.x = 0 # 目标质心的x坐标测量
        self.y = 0 # 目标质心的y坐标测量
        self.vx = 0 # 目标运动的x速度
        self.vy = 0 # 目标运动的y速度
        self.uint_x = 0 # 目标运动单位方向的x分量
        self.uint_y = 0 # 目标运动单位方向的y分量
        self.kernel_open_size = 4 # 开运算核
        self.kernel_close_size = 3 # 闭运算核
        self.lower = np.array([115, 50, 50]) # 蓝色阈值的下限
        self.upper = np.array([125, 255, 255]) # 蓝色阈值的上限
        self.corner_points = {0: (0, 0), 1: (0, 0), 2: (0, 0), 3: (0, 0)} # 角点字典
```

```
self.time = 0 # 两帧停留的时间
self.sec=0 # 计数器
```

手柄 (Paddle)

```
class Paddle:
    def __init__(self, frame=None):
        self.frame_original = frame # 初始图像
        self.frame_original = None # 初始图像
        self.frame_segmentation = None # 颜色分割后的图像
        self.frame_thresh = None # 二值化后的图像
        self.frame_preprocess = None # 预处理后的图像
        self.frame_locate = None # 目标定位标定
        self.radius = 0 # 目标外接球半径
        self.x = 0 # 目标质心的x坐标测量
        self.y = 0 # 目标质心的y坐标测量
        self.angle = 0 # 目标应运动的方向
        self.kernel_open_size = 4 # 开运算核
        self.kernel_close_size = 3 # 闭运算核
        self.lower = np.array([70, 50, 50]) # 绿色阈值的下限
        self.upper = np.array([90, 255, 255]) # 绿色阈值的上限
        self.corner_points = {0: (0, 0), 1: (0, 0), # 角点字典
                              2: (0, 0), 3: (0, 0)}
```