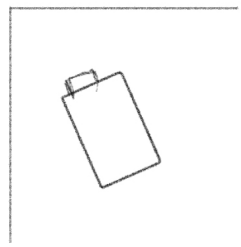
Chapter19 墙角自适应转向

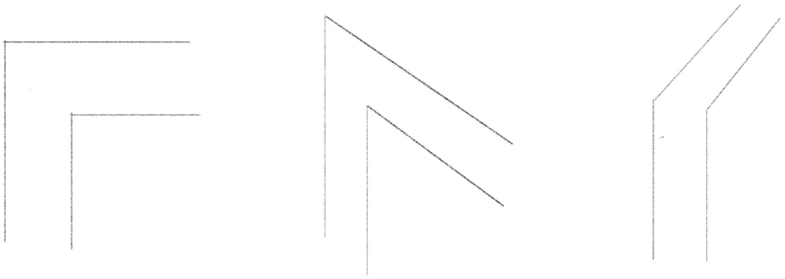
实验场景

本次实验模拟小车在墙角处的转向，通过采用超声波传感器来优化自动驾驶中的墙角转向，用最少的动作摆脱墙角。

墙角：



窄道：



解决方案

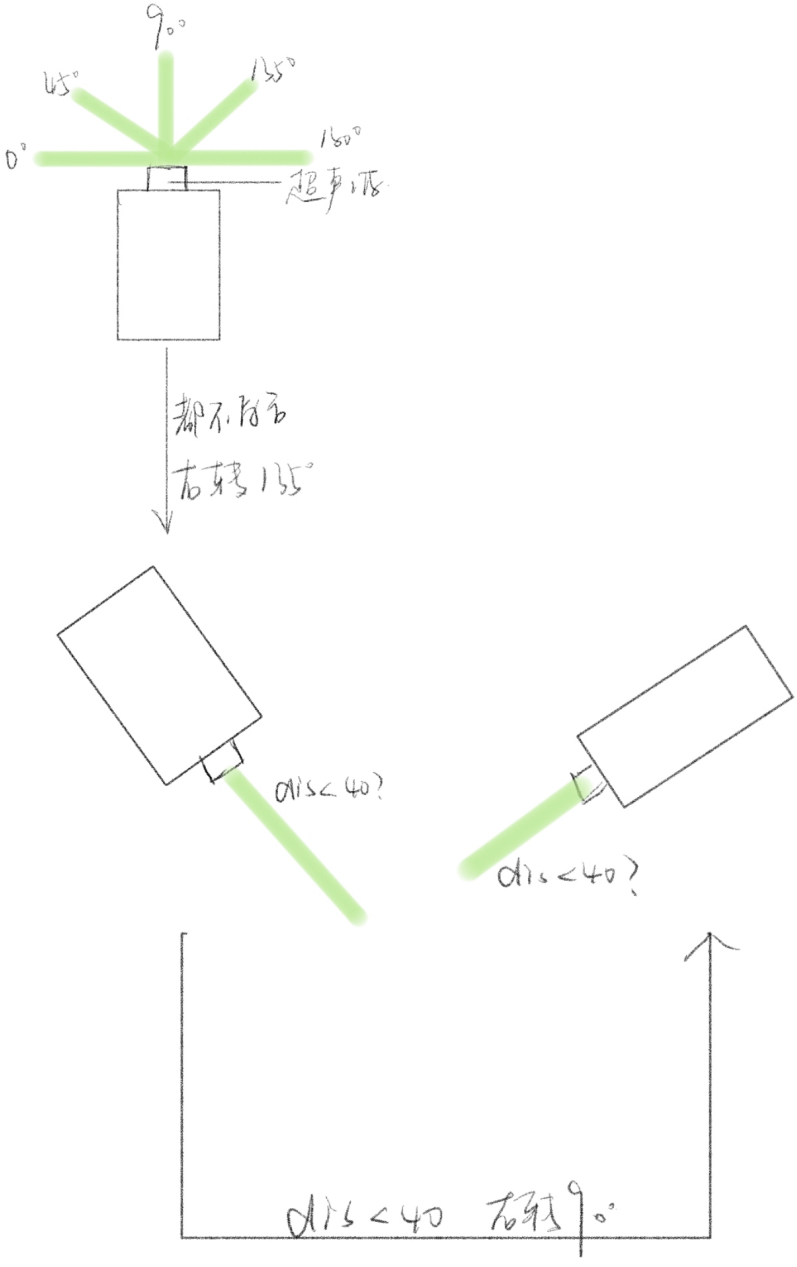
当小车靠近墙角时，用超声波传感器获取五个方向的距离，用一个列表来储存。

通过对五个距离的比较，来选择前进方向。

```
if d_180 - d_90>20:          #右转90
    spin_right_angle(90)
elif d_0 - d_90 > 20:        #左转90
    spin_left_angle(90)
elif d_135 - d_90>20:        #右转45
    spin_right_angle(45)
elif d_45 - d_90>20:         #左转45
    spin_left_angle(45)
```

如果依旧判断不出来，则右转135°，去判断是否可以通行。若依旧不行，再右转90，查看是否可以通行。若还是不行，则原路返回。

```
elif d_0 - d_90 <20 and d_180 - d_90 < 20:  #右后方和左后方
    spin_right_angle(135)          #右后方
    dis=get_distance()
    if dis<40:
        spin_right_angle(90)       #左后方
        dis=get_distance()
        if dis<40:
            spin_left_angle(45)    #掉头
            print("掉头")
        else:
            print("左转135")       #左后方前行
    else:
        print("右转135")           #右后方前行
else:
    spin_right_angle(90)
```



代码实现

```
# -*- coding:UTF-8 -*-
import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

from car_run import *
from infrared_distance import *
from infrared_light import *
from key import *
from led import *
from servo import *
from ultrasonic import *

# 按下开关开始启动，再按一下熄火
on_off = 0
dir_list=[]

def key_pressed_callback(pin):
    global on_off
    if not on_off:
        on_off = 1
    else:
        brake()
        car_run_clean()
        servo_clean()
        led_clean()
        exit(0)


def run_sides_space(run_speed, spin_time=0.2, spin_speed=SPIN_MID_SPEED):
    '''
    遇到障碍物,红外避障模块的指示灯亮,端口电平为LOW
    未遇到障碍物,红外避障模块的指示灯灭,端口电平为HIGH
    '''
    left_sensor = leftSensorValue_distance()
    right_sensor = rightSensorValue_distance()
    if left_sensor == GPIO.HIGH and right_sensor == GPIO.HIGH:
        print("左右都没障碍")  # 左右都没有
        run(run_speed, run_speed)
    elif left_sensor == GPIO.LOW and right_sensor == GPIO.HIGH:  # 左边有障碍
        print("左边有障碍")
        right(spin_speed)
        time.sleep(spin_time)
    elif left_sensor == GPIO.HIGH and right_sensor == GPIO.LOW:  # 右边有障碍
        print("右边有障碍")
        left(spin_speed)
        time.sleep(spin_time)
    else:
```

```
        print("左右都有障碍")  # 左右都有障碍
        spin_right_angle(90)


def choose_dir():
    '''
    dir_list:0  1    2   3    4
             前 左45 左90 右45 右90
    :return:
    '''
    dir_list.clear()
    brake()
    def get_dis_list(pos,sleep_time):
        servo_appointed_detection(pos)
        time.sleep(sleep_time)
        dir_list.append(get_distance())
    get_dis_list(75,0.1)
    get_dis_list(37,0.4)
    get_dis_list(0,0.4)
    get_dis_list(117,1.2)
    get_dis_list(160,0.4)
    servo_appointed_detection(75)
    time.sleep(0.8)
    d_0=dir_list[2]
    d_45=dir_list[1]
    d_90=dir_list[0]
    d_135=dir_list[3]
    d_180=dir_list[4]

    if d_180 - d_90>20:          #右转90
        spin_right_angle(90)
    elif d_0 - d_90 > 20:        #左转90
        spin_left_angle(90)
    elif d_135 - d_90>20:        #右转45
        spin_right_angle(45)
    elif d_45 - d_90>20:         #左转45
        spin_left_angle(45)
    elif d_0 - d_90 <20 and d_180 - d_90 < 20:  #右后方和左后方
        spin_right_angle(135)         #右后方
        dis=get_distance()
        if dis<40:
            spin_right_angle(90)      #左后方
            dis=get_distance()
            if dis<40:
                spin_left_angle(45)   #掉头
                print("掉头")
            else:
                print("左转135")      #左后方前行
        else:
            print("右转135")          #右后方前行
    else:
        spin_right_angle(90)
    brake()
    time.sleep(0.1)
```

```
if __name__ == "__main__":
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)

    car_run_init()
    infrared_distance_init()
    key_init()
    led_init()
    servo_init()
    ultrasonic_init()

    GPIO.add_event_detect(key, GPIO.RISING, key_pressed_callback, bouncetime=15)
    time.sleep(2)
    print("Press key2 to poweron!")
    while 1:
        if not on_off:
            brake()
            led_pause()
            servo_pause()
            while not on_off:
                pass

        distance = get_distance()
        if distance > 50:
            print("distance>50")
            run(MID_SPEED,MID_SPEED)
            LED_GREEN()
        elif distance > 20 and distance <= 50:
            print("20<distance<=50")
            LED_RGB(1, 0, 1)
            run(LOW_SPEED,LOW_SPEED)
        else :
            brake()
            choose_dir()

    car_run_cleanup()
    servo_cleanup()
    led_cleanup()
```

实验成果

本次实验模拟了小车在墙角处的转向，将小车测距转化为五个方向距离的比较，将墙角处进行建模，通过采用超声波传感器来优化自动驾驶中的墙角转向，用最少的动作摆脱墙角。