



玩转视觉识别
小 MU 视觉传感器 - 深度开发指南



杭州摩图科技有限公司

版本 V0.1

2019.08.10

免责声明和版权公告

•本手册是摩图科技公司所生产的小 MU 视觉传感器(下称小 MU)MUVS-AB2 型号 V311 版本的配套手册,文中内容仅针对小 MU 而编写;

•本手册属于通用教程,主要介绍的是小 MU 的原理、思维与技巧,帮助用户更好地理解小 MU 和使用小 MU,本手册并不是完整的教材,文中只有部分代码或示意代码;

•手册中的图片及标识是为了更好的解释说明而绘制的,并不代表用户可以真实的看到这些图像内容;

•MORPX 是杭州摩图科技有限公司的注册商标, MU 是小 MU 视觉传感器的注册商标。文本或图片中涉及到的所有商标(名称与图案)归属于其持有者,文中照片的人物肖像权归其本人所属有,特此声明;

•本手册属于电子文档,非出版物,未经摩图科技有限公司许可,请勿擅自印刷出版或引用文中内容,造成法律上侵权行为的由侵权者承担;

•本手册将以 MicroBit 平台来介绍小 MU 的使用,本手册不是 MicroBit 的基础教程,对其不了解的请参照 MicroBit 的相关教程,本手册不提供其他开发平台的示意代码。

目录

免责声明和版权公告	1
1 前言	4
2 小 MU 简介	4
2.1 小 MU 工作原理	5
2.2 小 MU 使用步骤	5
2.2.1 设置通讯模式	5
2.2.2 设置小 MU 地址	6
2.2.3 线路连接	7
2.2.4 编写代码并烧录	8
2.2.5 运行测试	9
3 小 MU 参数设置	9
3.1 摄像头帧率	9
3.2 LED 颜色	10
3.3 数码变焦与视野关系	10
3.4 白平衡的作用	11
3.4.1 白平衡模式	11
3.4.2 锁定白平衡的校准方法	11
3.4.3 锁定白平衡模式的使用场景	12
3.5 算法性能设置	14
3.6 光源的影响	14
3.7 物体大小与检测距离的关系	15
4 视觉算法简介	16
4.1 图像坐标系与检测结果	16
4.2 色块检测与颜色识别的区别	18
4.2.1 颜色 Label 分类标签的定义	18
4.2.2 色块检测框的作用	19
4.2.3 色块检测的调用方式	19
4.2.4 颜色识别区域的作用	19
4.2.5 颜色识别的调用方式	20
4.3 球体算法与背景干扰	21
4.4 人体算法的有效特征	22
4.5 卡片算法的有效角度	23
5 数据处理方法	25
5.1 判断是否检测到目标物体	25
5.2 分类标签 label 的使用	25
5.2.1 分类标签的基本调用方式	25
5.2.2 无人驾驶小车	26
5.2.3 数字乐器演奏	26
5.3 同一平面空间判断物体方位	27

5.4 同一平面空间判断物体远近	28
5.5 垂直平面空间判断物体方位	29
5.6 垂直平面空间判断物体远近	30
5.7 视觉循线-单线	30
5.8 视觉循线-模拟道路	33
5.9 颜色识别实现循线	36
5.10 二维空间区域分割	37
5.10.1 区域分割的方法	37
5.10.2 轨迹识别	38
5.10.3 图案扫描	40
5.10.4 图案识别	41
5.11 三维空间坐标系	42
5.11.1 三维空间密码锁	42
5.11.2 三维空间乐器	43
5.11.3 三维空间绘画	43
5.12 同时开启多种算法	44
5.12.1 循线加上颜色识别	44
5.12.2 多类卡片同时识别	45
5.12.3 利用颜色识别物体	46
5.13 算法的衍生	46
5.13.1 利用颜色识别来实现加减乘除	46
5.13.2 利用小球实现任意物体跟踪	47
5.13.3 利用形状卡片表达一个篮球框	47
5.14 数据的特殊处理	47
5.14.1 提升检测结果的可靠性	47
5.14.2 消抖措施	48
5.14.3 快速跟踪	48
5.14.4 施密特触发	49
5.14.5 排除异常数据	50
6 服务与支持	50
修订历史	51

1 前言

对于人工智能 AI (Artificial Intelligence) 既熟悉又陌生，熟悉是因为 AI 已经逐渐的渗入到日常生活之中，例如手机拍照的人脸识别，移动支付的二维码识别，商场里的服务机器人，家庭中的智能音箱，或是无人驾驶汽车等。陌生却是因为人工智能技术太深奥，似乎只有那些专家或是学者才需要去关心其技术本身。

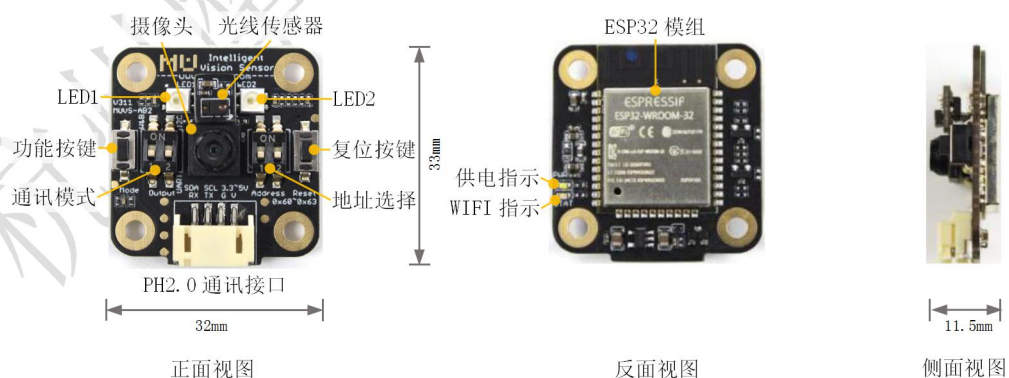
如今在国家的支持下，人工智能技术已经开始面向青少年教育领域发展，结合先进的 STEAM 教育理念，人工智能教育得到了很多行业机构或教育人士的关注，市面上也开始出现一些具有 AI 属性的产品和课程，同时在比赛方面也越来越多的融入人工智能的主题。青少年人工智能教育在未来几年内将与编程教育一样，成为下一个教育行业重要的发展方向。杭州摩图科技有限公司自主研发与生产的一些列产品专注于人工智能教育的青少年化。

小 MU (目) 系列视觉传感器面向嵌入式图像识别领域，内置多种实用的视觉算法，经多次产品迭代，小 MU 愈发成熟与强大，不仅识别率高，稳定性好，而且官方能持续提供升级与算法扩展。此外，在小 MU 使用过程中，用户无须关注算法的处理过程，更无需进行复杂的参数调教，结合可视化编程模块，可以将更多的精力和时间投入到作品和产品的设计之中。

小 MU 视觉传感器的本质是“传感器”，它的传感方式是通过“看”来实现的，输出的数据还是需要进行“人工处理”，如何应用这些数据设计出有创意的作品，甚至解决生活中的问题，这将会是很有意义的事情，也是青少年阶段人工智能教学的重点。对于视觉算法的基本原理我们也应当有所了解，但不必抱着“掌握”的态度，青少年对算法原理的学习，其目的不是为了再创造一个算法，而是为了让视觉算法更好更稳定的为算法本身服务，拓展思维，开阔眼界。

本手册将介绍小 MU 视觉传感器的一些基本的视觉算法知识，图像处理的知识，数据处理方法，以及一系列的应用案例，相信本手册能够让你更深入的理解小 MU 视觉传感器，解决使用过程中遇到的一些疑惑与问题。

2 小 MU 简介

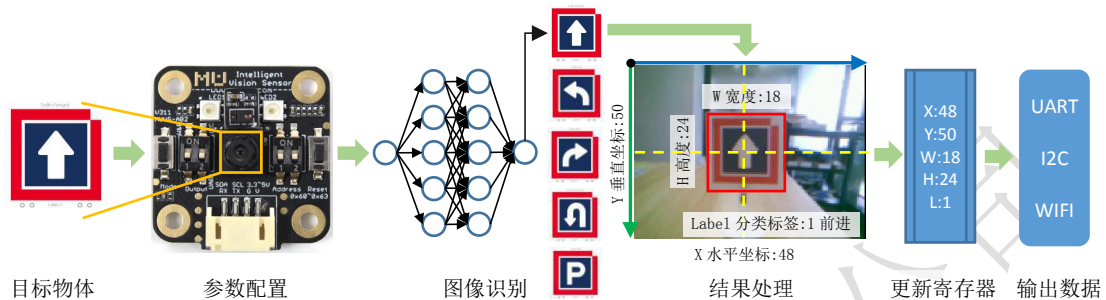


小 MU 是一个集成了多种视觉算法及传感功能的嵌入式图像识别模块，采用 ESP32 双核处理器，配有一颗 VGA85°无畸变广角镜头，具有速度快，视野广，体积小，功耗低等特点。小 MU 的视觉算法全部由摩图科技自主研发，包括色块检测，颜色识别，球体识别，人体检测，以及形状、交通、数字卡片识别，通过固件升级还可以获得更多算法的支持。摄像头上方具有一颗高度集成的光线传感器，可以实现环境光强度的检测，红外测距，手势方向识别功能。模块自身还具有 WIFI 功能，可以实现图传和无线通讯。小 MU 采用 UART 和 I2C 作为主要通讯方式，

可以在任何嵌入式平台上使用。针对创客与青少年教育，小 MU 全面支持 Arduino，MicroBit，Mixly，Mind+主流开发平台。

备注：*光线传感器和 WIFI 功能尚未对外开放，待后续固件升级

2.1 小 MU 工作原理



小 MU 作为嵌入式视觉识别模块，其算法全部在芯片内部完成运算处理，无须云端服务器的支持，通过一些简单的寄存器配置即可开启算法。算法开启后，小 MU 将开始不断的处理摄像头所拍摄到的图像，如果图像中出现了目标物体（与算法检测内容一致的物体）且大小，角度，清晰度等可以满足识别的要求，那么就会将该物体在图像中的相对位置，大小，分类编号等数据写入寄存器，每处理一幅图像这些数据就会被刷新，用户通过读取这些寄存器来获取图像识别的结果。

在 Arduino 平台，提供了完整的库函数来实现对小 MU 的操作，在 MicroBit，Mixly，Mind+ 平台中则提供了可视化编程模块。使用 C 代码编程将会更有利于发挥小 MU 的性能，而在可视化编程中，为了让用户操作简单易上手，对部分功能做了简化处理。

2.2 小 MU 使用步骤

以 MicroBit 人体检测算法为例进行介绍。

2.2.1 设置通讯模式

小 MU 支持 4 种通讯模式：UART，I2C，WIFI*，图传*模式。根据所需要的通讯方式，拨动小 MU 左侧的 Output 拨码开关。

输出模式	拨码开关	编号	LED 指示	简介
UART		00	闪烁红色	支持 9600(默认)~921600 波特率 ⁽¹⁾ ，8 位数据位，1 位停止位，无校验。基于 MU-Protocol 协议进行数据读写，波特率较低时会降低图像识别帧率
I2C		01	闪烁绿色	最高支持 400Kbps，直接对寄存器读写操作，使用较为灵活，效率高
WIFI ⁽²⁾		10	闪烁黄色	使用 WIFI-UDP 方式进行数据通讯，基于 MU-Simple-Protocol 协议以指令的方式来读写数据，可以用于物联网应用
图传 ⁽²⁾		11	闪烁紫色	通过浏览器来查看图传数据以及发送遥控指令，浏览器中输入地址 192.168.4.1 即可查看图像

备注：(1)因 MicroBit 通讯性能限制，MicroBit 仅支持 9600 波特率，建议使用 I2C 模式；

(2)WIFI 模式和图传模式目前版本尚不支持，待后续固件更新。

选择通讯方式后，程序中的通讯方式应与拨码开关保持一致。编程时应首先配置通讯方式，然后才可以进行其他的参数配置，使用过程中不可更改，每次切换通讯方式，需要重启小 MU。



初始化时配置通讯方式

2.2.2 设置小 MU 地址

小 MU 支持 4 个地址：0x60（默认），0x61，0x62，0x63。当小 MU 与其他传感器地址冲突时需要进行更改。I2C 模式下支持多个不同地址的小 MU 协同工作，可以给小 MU 分配不同的地址。一般情况下保持默认地址即可。

设备地址	拨码开关	编号	设备地址	拨码开关	编号
0x60		00	0x62		10
0x61		01	0x63		11

选择设备地址后，代码中的设备地址应与拨码开关保持一致。编程时应首先配置设备地址，然后才可以进行其他的参数配置，使用过程中不可更改，每次切换设备地址，需要重启小 MU。地址选择后，后续代码应与该地址保持一致。



初始化时配置设备地址



程序中的设备地址应保持一致

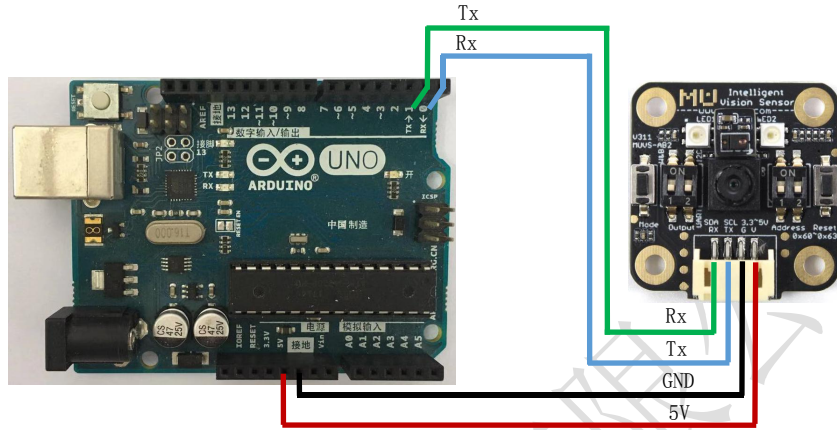
2.2.3 线路连接

线路连接以典型的 Arduino UNO 和 MicroBit 主控为例进行介绍，其他主控板连接方式与示例类似。

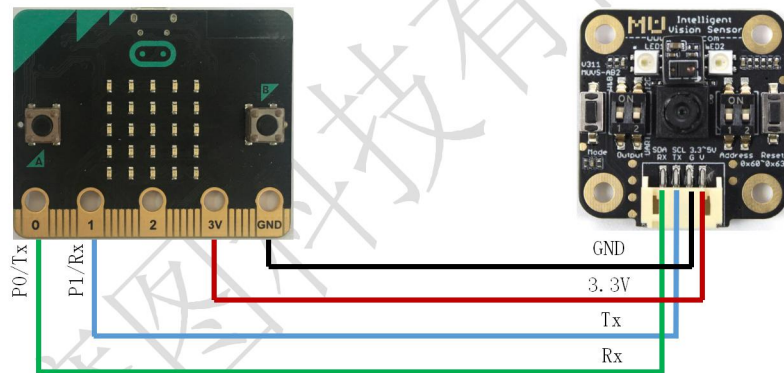
UART 模式：

小 MU-Rx 接主控板 Tx，小 MU-Tx 接主控板 Rx，（也可使用自定义的软串口连接）

小 MU-G 接主控板 GND，小 MU-V 接主控板 5V



小 MU 与 Arduino 连接 UART

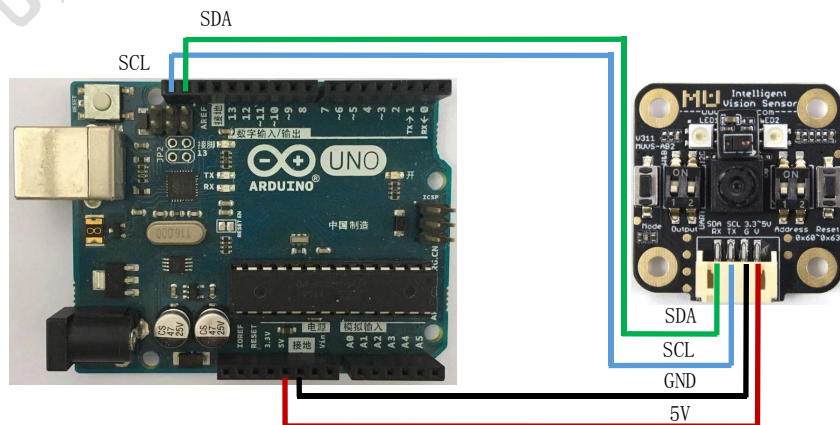


小 MU 与 MicroBit 连接 UART

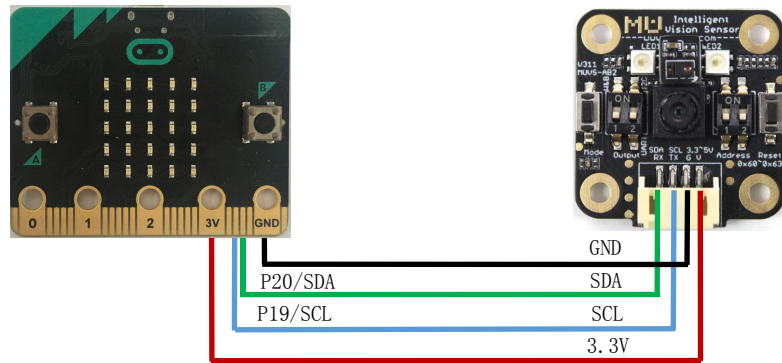
I2C 模式：

小 MU-SCL 接主控板 SCL（或 C，CLK），小 MU-SDA 接主控板 SDA（或 D，DATA）

小 MU-G 接主控板 GND，小 MU-V 接主控板 3.3V 或 5V



小 MU 与 MicroBit 连接 I2C



小 MU 与 MicroBit 连接 I2C

2.2.4 编写代码并烧录

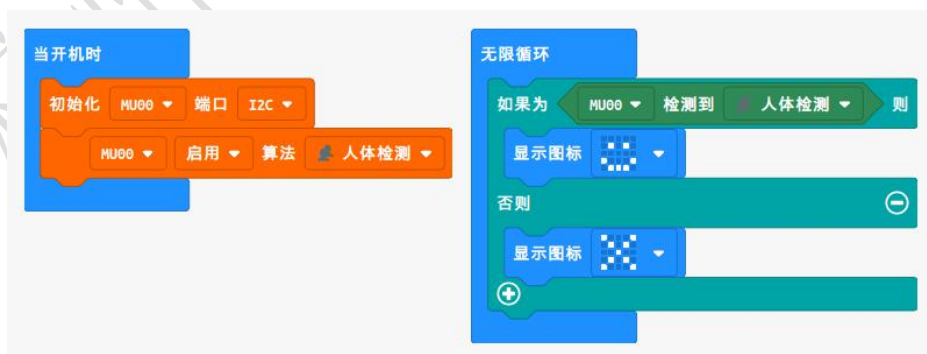
打开浏览器，进入 MicroBit 在线编辑页面 <https://makecode.microbit.org>

进入左侧边栏>高级>扩展，搜索项目“mu-opensource/pxt-MuVisionSensorIII”，加载 MUVisionSensor 库。



MicroBit 加载小 MU 库函数

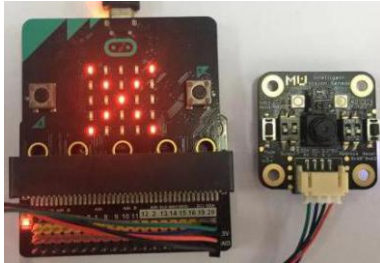
返回编辑页面并编写一个简单的人体检测程序（I2C 模式），代码如下：



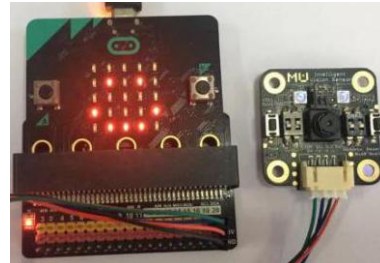
MicroBit:人体检测代码

2.2.5 运行测试

将生成的 Hex 文件烧录至 MicroBit 中并运行程序。程序开始后，小 MU 首先 LED 短暂闪烁一下白色，代表算法已开启，然后 LED 开始闪烁红色或蓝色，其中红色代表未检测到人体，蓝色代表检测到人体，与此同时，如果小 MU 检测到人体，则 MicroBit 屏幕上会显示笑脸，否则显示叉号。



未检测到人体时，显示叉号 X



检测到人体时，显示笑脸:)

3 小 MU 参数设置

小 MU 提供了简单而实用的参数设置功能：摄像头帧率，LED 颜色，数码变焦，白平衡，算法性能设置。合理设置这些参数，将会有利于小 MU 性能的提升。

3.1 摄像头帧率

摄像头帧率指摄像头采集图片的速度，分为高速模式和标准模式，高速模式为 50fps，标准模式为 25fps。提高摄像头帧率可以提高图像处理的速度，缓解图像拖影问题，但功耗会有一定增加。环境较暗时，标准模式比高速模式成像要亮一些。

备注：V112 之后的固件默认为高帧率模式。



高帧率模式，较暗的环境



标准帧率模式，较暗的环境



高帧率模式，非常暗的环境



标准帧率模式，非常暗的环境

高帧率与标准帧率模式下成像明暗对比

3.2 LED 颜色

小 MU 板载 2 颗 RGB-LED，左侧为 LED1，右侧为 LED2，可以用于指示小 MU 的识别状态，在默认配置下，检测到物体显示蓝色，未检测到物体显示红色。

备注：颜色识别算法不会开启灯光指示功能。



如果使用 Arduino 库编程，我们还提供了 LED 的高级控制功能，包括手动灯光模式，亮度调节，灯光保持等。我们可以在手动模式下把 LED 亮度提高，作为补光灯使用。



未开启 LED，非常暗的环境



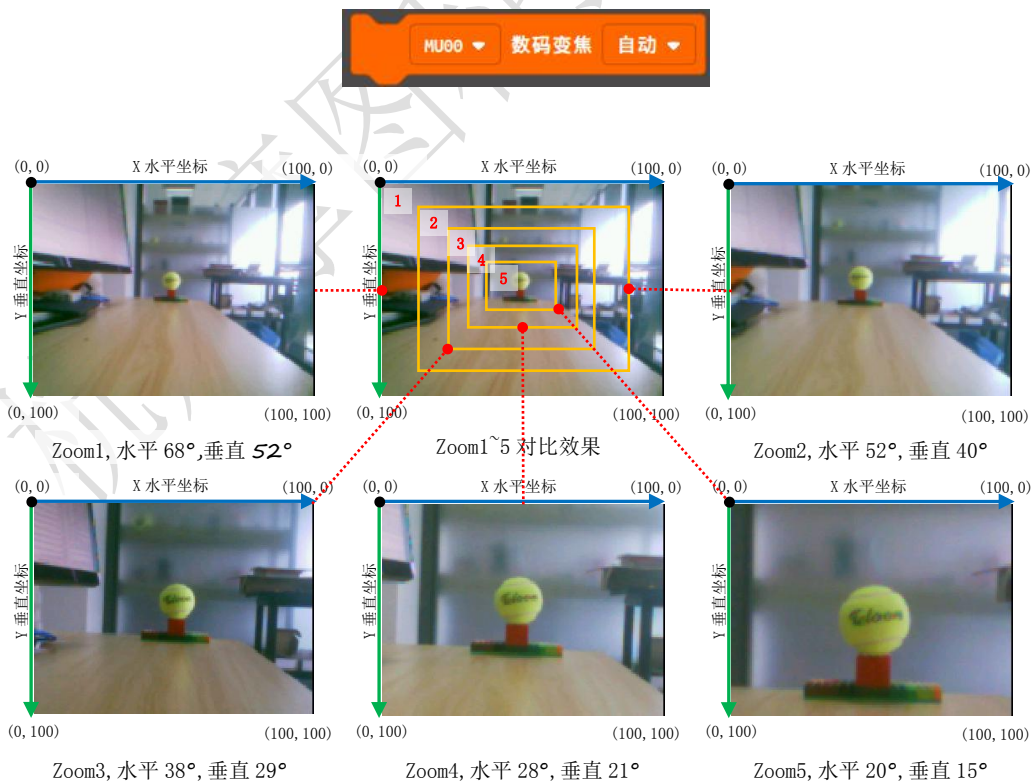
开启高亮白色 LED，非常暗的环境

LED 作为补光灯效果

3.3 数码变焦与视野关系

小 MU 提供了数码变焦功能，共有 5 个档位（Zoom 值）可以使用，利用镜头缩放功能可以看到远处的目标物体。Zoom 值小，则视野广，距离近，Zoom 值大，则视野窄，距离远。

备注：使用 Zoom3~5 时，图像帧率会比 Zoom1~2 低。



3.4 白平衡的作用

摄像头在拍摄照片时，由于外界光线或背景的色温不同，会对图像产生一定的色偏。为了还原真实的色彩，通过白平衡功能调节 R 红色，G 绿色，B 蓝色的增益来纠正这种色偏。但白平衡功能并不总是有利于色彩还原的，合理使用白平衡对于颜色相关的算法是十分关键。



3.4.1 白平衡模式

白平衡的调节分为自动调节方式和固定参数方式，对于一张构图颜色均衡的图像，其 R，G，B 三个通道的值应是分布均匀的，如果 R、G、B 通道分布出现了失衡，自动白平衡功能会适时的修改各通道的色彩增益值使其重新分布均衡，而固定白平衡是不会改变其色彩增益的。

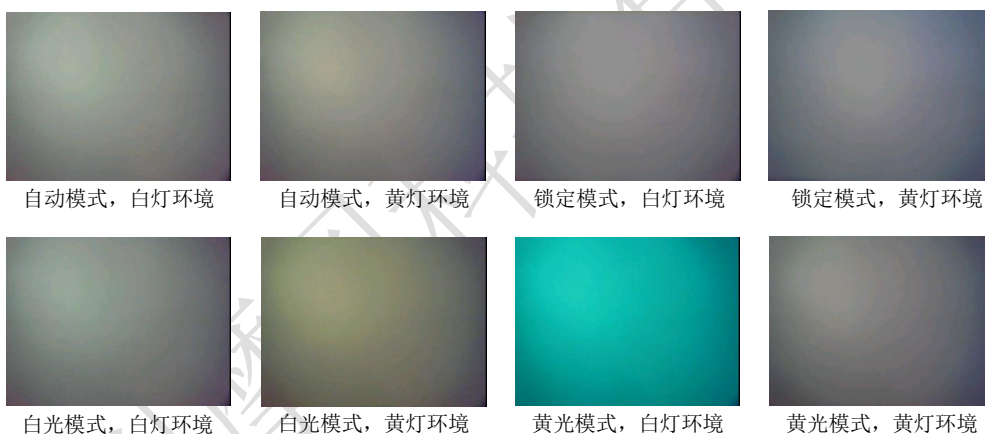
小 MU 视觉传感器提供了 4 种白平衡模式：自动模式，锁定模式，白光模式，黄光模式。

自动模式：适用于光照良好且对颜色要求不高的环境中使用；

锁定模式：适用于颜色还原要求较高的环境，让小 MU 面对白纸进行白平衡校准，然后锁定白平衡参数，锁定后颜色不会随环境的改变而变化；

白光模式：适合于白色灯光或者阴天环境中使用，该模式也属于自动白平衡模式；

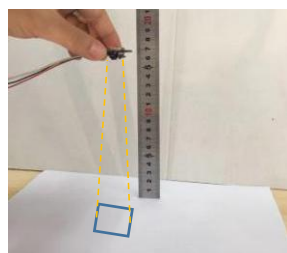
黄光模式：适合于黄色灯光或者阳光环境中使用，该模式也属于自动白平衡模式。



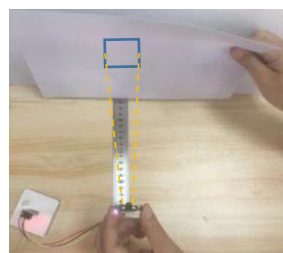
4 种白平衡模式分别在白色灯光和黄色灯光下的测试效果对比

3.4.2 锁定白平衡的校准方法

(1) 将小 MU 面向一张白纸 10~20cm 处，确保摄像头中心区域完全面向白纸，灯光照向白纸，但尽量不要有阴影遮挡；



小 MU 垂直向下面向白纸

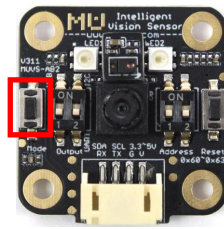


小 MU 水平向前面向白纸

(2) 调用锁定白平衡指令，或短按功能按键（小 MU 左侧 mode 键）；



方法 1：调用指令方式



方法 2：短按功能按键方式

(3) 小 MU 的 LED 会闪烁一下白色灯光，开始白平衡校准；

(4) 几秒钟后，当 LED 闪烁一下青色灯光后，表明白平衡校准完成；

(5) 如果 LED 闪烁为紫色灯光，则表明校准失败，可以再次重新校准。

备注 1：由于校准白平衡需要几秒钟的时间，所以使用软件锁定白平衡时应该位于小 MU 设置最后使用，并设定 2 秒左右的暂停时间等待校准完成。

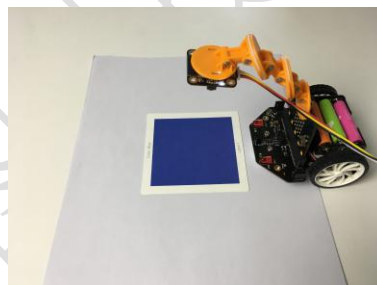
备注 2：按键方式可以在小 MU 运行的任何时刻去校准白平衡，而指令模式只适合于开机后的配置阶段使用。

备注 3：如果校准时使用的不是白纸，即便校准后显示为青色灯光，那么图像也可能是处于偏色状态的，因为该功能会将其他颜色尽量调节为白色。

3.4.3 锁定白平衡模式的使用场景

自动白平衡，白光模式，黄光模式对于大部分应用场景都是适用的，但是如果图像中存在大面积单色物体时，自动白平衡功能就失去了意义，不仅不能还原色彩，还会将色彩调节失衡，尤其是在使用颜色相关的算法时，需要使用锁定白平衡模式。

小 MU 视野中存在大面积单色物体时：



小 MU 视野中有一个大面积的色块
备注：图中小车为 DFRobot 的麦昆小车（下同）



自动白平衡
大片红色周围呈现淡青色



自动白平衡
大片绿色周围呈现淡紫色



自动白平衡
大片蓝色周围呈现淡黄色



校准并锁定白平衡
大片红色周围呈现白色



校准并锁定白平衡
大片绿色周围呈现白色

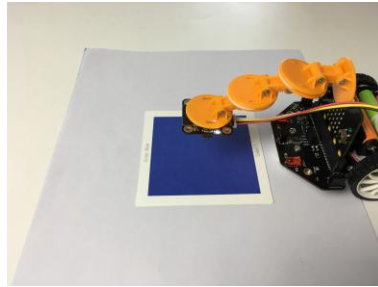


校准并锁定白平衡
大片蓝色周围呈现白色

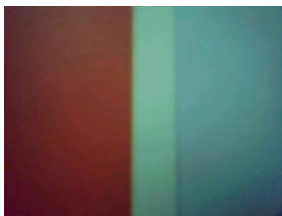
在上述图片的第一排图像中，当图像中出现大面积红色时，红色 R 通道值明显高于 G、B 两个通道，摄像头为了调节白平衡而将 G、B 增益提高，所以其周围的白色区域呈现出了淡青色。类似的，大面积绿色周围会呈现淡紫色，而大面积蓝色周围会呈现淡黄色。

经过校准并锁定白平衡后，三种颜色周围的白色被正确的还原，不存在偏色现象。

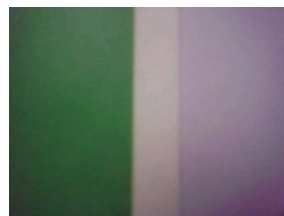
小 MU 近距离检测颜色时：



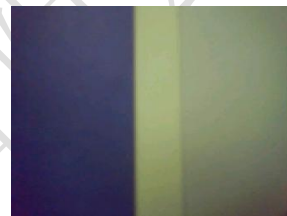
小 MU 近距离检测颜色



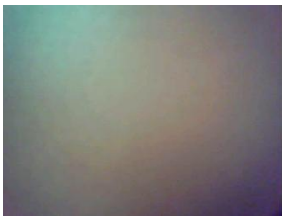
自动白平衡
近距离，图像左半边为红色



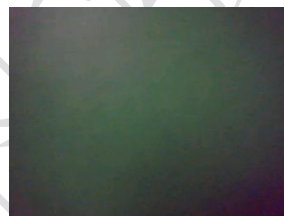
自动白平衡
近距离，图像左半边为绿色



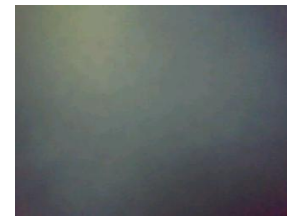
自动白平衡
近距离，图像左半边为蓝色



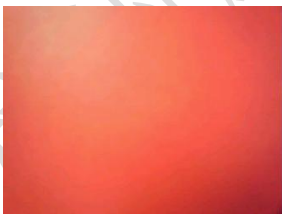
自动白平衡
近距离，图像全部为红色



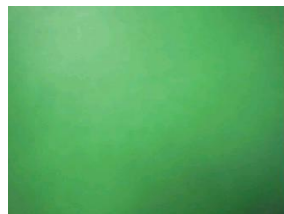
自动白平衡
近距离，图像全部为绿色



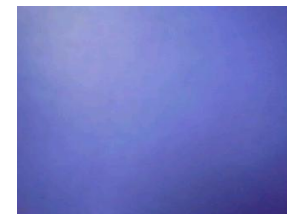
自动白平衡
近距离，图像全部为蓝色



校准并锁定白平衡
近距离，图像全部为红色



校准并锁定白平衡
近距离，图像全部为绿色



校准并锁定白平衡
近距离，图像全部为蓝色

在这一组测试中，摄像头十分靠近单色物体时，自动白平衡为了还原白色，将原本的红绿蓝卡片调节到完全失衡的状态，此时已经失去了原本的颜色信息，校准并锁定白平衡后，其原本颜色可以得到正确的还原。

3.5 算法性能设置

小 MU 视觉算法的运行分为两个阶段，一是算法处理，二是结果处理。这两个阶段对算法的性能和速度起到关键的作用。

算法处理阶段：算法处理时会有一些内部参数的设置，参数的不同会直接影响检测效果。一个算法也可能是由不同的小算法单元构成的，这些小的算法单元可能是负责图像预处理、检测、分类、校验、优化等等。如果这些小算法单元全部使用，那么检测结果的准确率是最高的，环境适应性也是最好的，但处理时间会变慢。如果我们放弃一些小算法单元，那么将会提高算法的运行速率，在可控的环境下也是可以得到较好的检测结果的。

结果处理阶段：算法处理后的数据是原始数据，中间可能夹杂着错误结果，偶尔的漏报，或是误报等问题，所以在检测结果正式的输出前，会对数据结果做一定的预处理。

根据这两个阶段的不同的处理方法，每类算法都提供了 3 种性能选项：**速度优先，均衡性能，准确率优先**。用户可以根据实际应用需求来选择最适合的性能：

性能选项	特点	适合场景
速度优先	通过去掉一定的算法步骤来提高处理速度，在可控环境下依然具有不错的检测效果，数据结果不做任何优化处理，直接输出原始数据	需要高速处理的应用，如实时跟踪某个物体；对某个物体需要做出快速的响应；场地或背景可以控制的环境
准确率优先	以最佳的性能来运行算法，具有最可靠的检测结果，输出结果经过处理，一些异常数据会被优化掉，具有一定平滑性，但有些算法的运行速度会明显降低	需要可靠的检测某个物体的结果；对速率要求不高的应用；场地或背景复杂且不可控的环境
均衡性能	介于速度优先和准确率优先之间的模式，也是小 MU 的默认模式	适合一般场景的应用需求

3.6 光源的影响

良好稳定的光源对于视觉识别是一个非常重要的因素，采用劣质光源，强光直射，昏暗环境，逆光环境等都不利于图像识别。



劣质光源的水波纹现象



强光直射造成局部过曝



昏暗环境造成物体颜色失真



逆光造成颜色信息丢失

劣质光源：摄像头在有些光源下会产生水波纹现象，当水波纹非常严重时就会影响到小 MU 的正常工作，可以用手机的相机功能来检测光源是否符合要求，不合格的光源在大部分手机上也会产生水波纹现象。但这个现象并不一定会发生，这个与摄像头的曝光时间有关，视野内越明亮，曝光时间短，水波纹也就越明显；

强光直射：当强光直射到目标物体上时，会造成其局部曝光过度或者反光的现象，此时会影响到图像的完整性，导致无法正常识别。但通常这个现象只发生在一些特定的角度，有时只需要换个角度就可以正常识别了；

昏暗环境：与人眼不同，摄像头对昏暗环境处理能力十分有限，不能依据人眼来判断是否为昏暗环境。在自动白平衡模式下，摄像头的增益功能可以提高暗环境一定的亮度，但同时会产生明显的噪点，加之环境中光线微弱，目标物体的颜色无法被正确的还原，导致图像识别效果变差，甚至无法识别，不建议在昏暗环境中使用，如必须使用，可以使用 LED 补光灯；

逆光环境：当摄像头从昏暗的屋内面向明亮的窗户时，就会处于明显的逆光环境，逆光会造成颜色信息的严重丢失，影响小 MU 的正常使用。但如果屋内是明亮的，那么逆光问题就会相对较弱，开启 LED 补光灯也可以缓解近距离的逆光问题。

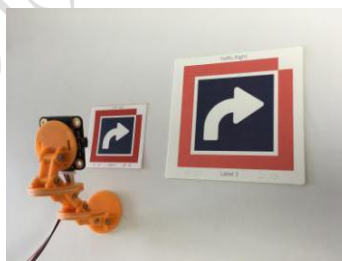
3.7 物体大小与检测距离的关系

目标物体在小 MU 中成像的大小对于是否能被检测到起到至关重要的作用，我们可以利用数码变焦的高 Zoom 值来检测远距离物体，也可以利用低 Zoom 值来检测近距离物体，但对于一些特殊情况还是无法满足我们的需求，此时可以通过放大或缩小目标物体的尺寸来解决。

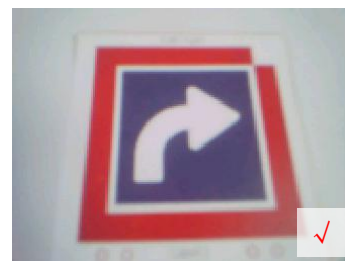
近距离需要缩小目标物体：当我们需要近距离检测物体时，如果物体尺寸太大，物体那么小 MU 中的图像可能已经超出了镜头范围，无法有效检测，此时我们可以将物体尺寸缩小，这样就可以在很近的距离内检测到物体了。



近距离面对大卡片时，图像显示超出范围，无法识别



近距离采用小卡片，图像显示正常，可以识别



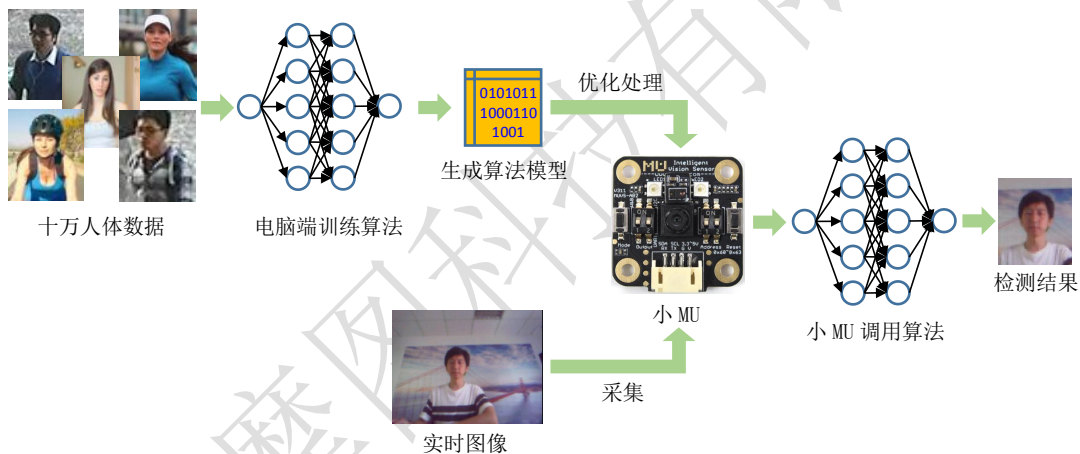
远距离需要放大目标物体：反之如果我们要检测的物体距离很远，为了能够看到远处的物体，我们可以适当放大其尺寸，这样即便距离比较远也是可以检测到的。



左侧的小卡片远距离无法检测到，右侧的大卡片远距离可以检测到

4 视觉算法简介

小 MU 采用先进的计算机视觉处理技术——深度学习。通过在电脑端对数万张图片数据样本进行学习训练，生成一个针对嵌入式系统而优化的算法模型，该算法模型可以在小 MU 的处理器上快速而稳定的运行。小 MU 通过摄像头实时的采集图像信息并调用算法模型进行视觉识别，返回检测到的数据结果。



小 MU 视觉算法工作流程示意图

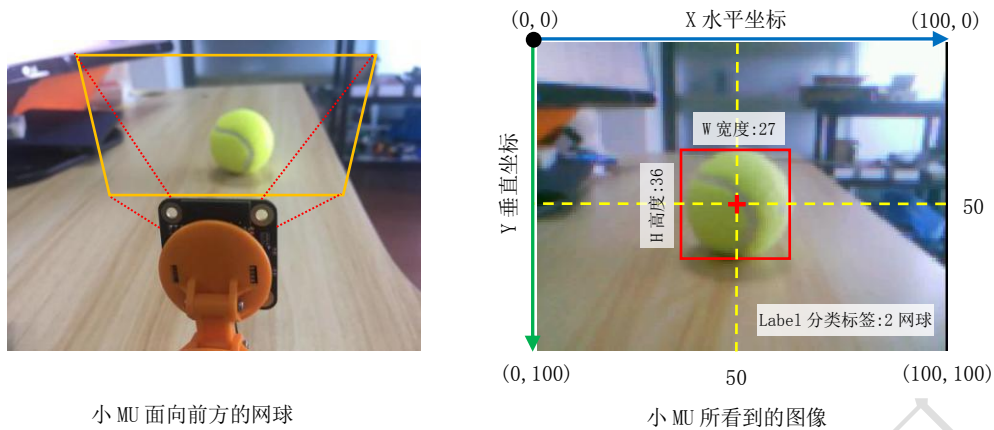
4.1 图像坐标系与检测结果

坐标系的概念大家并不陌生，生活中常用的有 1 维坐标系（线段），2 维坐标系（平面），3 维坐标系（空间），作为图像而言是一个 2 维坐标系，它的两个轴分别表示 X（水平坐标，横向坐标），Y（垂直坐标，纵向坐标）。

在数学教材中，我们常用的 2 维坐标系的原点为左下角，而在图像坐标系中，采用了左上角为坐标原点，这是因为在显示一幅图像时，像素点总是从左上角开始，从左至右，由上而下逐行的向下显示数据的。我们以球体算法为例，讲解坐标系的概念以及小 MU 各检测结果的含义。

当一个网球处于小 MU 前方时，小 MU 会检测到该网球并返回 Detected 是否检测到，X-Y 坐标，W-H 宽高，Label 分类标签。

备注：小 MU 还有检测结果数量的信息，但目前算法只支持单物体的检测或识别，所以这个值目前始终为 1。



Status 检测状态 :当小 MU 检测到目标物体时,该值输出为“真 True”,否则为“假 False”。在读取其他算法结果时,首先应该检查是否检测到目标物体,然后再去处理坐标数据或是宽高数据,否则就继续检测,这样可以提高不少的代码效率。

X-Y 坐标 :小 MU 摄像头输出图像的宽高比例为 4:3。在小 MU 图像坐标系中,水平与垂直坐标均被量化到了 0~100 的范围内,其含义是该坐标相对于完整图像的百分比位置。例如 X 为 30 时,即该物体的 X 坐标位于水平方向距离原点 30%的位置上。

W-H 宽高 :当小 MU 检测到物体后,还会返回物体的 Width 宽度值和 Height 高度值,其量程同样被量化到了 0~100 的范围内,其 W 宽度的含义是其在整个水平方向所占据的百分比值,H 高度的含义是其在整个垂直方向所占据的百分比值。需要注意的是,由于水平垂直的比例是 4:3,所以对于宽高相同的正方形物体而言,其输出的宽高值是存在一个 3:4 的比例关系的,而并不是相等的关系。

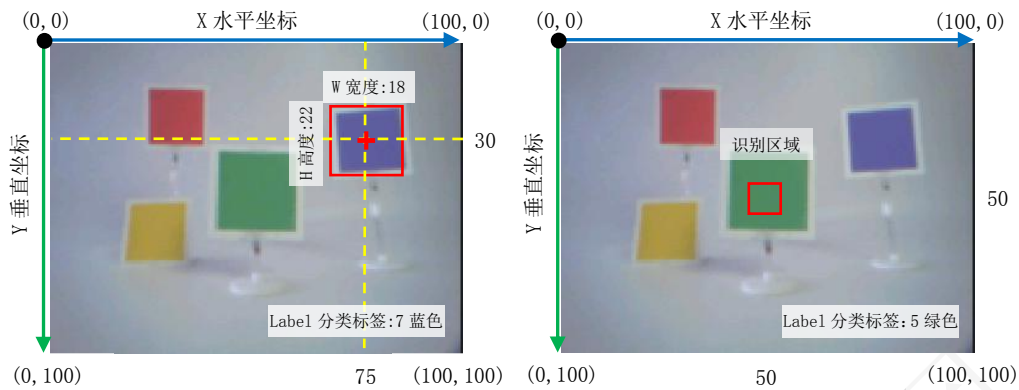
Label 分类标签 :对于具有分类识别功能的算法而言,还会根据所检测到的不同物体输出其对应的分类标签值,该值是小 MU 内部设定好的一系列数字,每一个数字对应一个特定的物体,以球体算法为例,当 Label 标签为 1 时,代表检测到了一个乒乓球,而如果 Label 标签输出为 2,则说明此刻检测到的是一个网球。Label 分类标签在颜色和卡片的算法中是一个十分关键的信息。

示意代码：



当检测到球体时,输出各检测结果的数值

4.2 色块检测与颜色识别的区别



色块检测: 设定为检测蓝色, 小 MU 会在图像中寻找蓝色色块方位

颜色识别: 设定识别区域位于图像中心, 图中识别结果为绿色, Label5

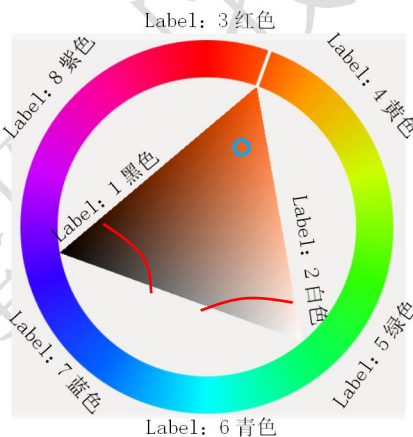
色块检测: 指定一种颜色和该颜色块尺寸的最小值, 然后小 MU 在图像中寻找符合该要求的一个色块区域, 返回其坐标和大小, 解决“是不是”的问题。

颜色识别 指定一个检测区域, 然后小 MU 统计该区域中像素的颜色, 并返回该颜色的 Label 值标签以及具体的红 R, 绿 G, 蓝 B 通道的值, 解决“是哪一个”的问题。

备注: “色块检测”旧称“颜色检测”, 二者是同一个算法。

4.2.1 颜色 Label 分类标签的定义

小 MU 共定义了 8 种颜色, 其分布区间如下图所示



颜色 Label 分类标签分布图

颜色算法分类标签:

分类标签	目标物体	图案	分类标签	目标物体	图案
1	黑色 (深灰色)		2	白色 (浅灰色)	
3	红色		4	黄色 (橙色)	
5	绿色		6	青色 (蓝绿色)	
7	蓝色		8	紫色	
0	未知	N/A			

4.2.2 色块检测框的作用

在使用色块检测算法时，需要设置一个色块检测框，只有符合检测框的颜色以及大于该检测框的色块才会被检测到，小于该检测框的色块将会被忽略。通过设置合适的大小的检测框可以排除背景中杂色的干扰；

色块检测框有 3 个参数可以设定：检测框最小宽度（默认 6），检测框最小高度（默认 8），颜色标签（默认 3：红色）。



四个色块对比

假设我们要检测宽 15，高 20 的红色色块，那么在上图的 4 个区域中

区域 1：可以检测到，满足色块检测框的需求

区域 2：无法检测到，色块尺寸太小，不符合检测框设置

区域 3：无法检测到，颜色不符

区域 4：无法检测到，颜色不符

备注：色块检测框的大小只可在使用 Arduino 库开发时进行更改，可视化编程平台只提供颜色 Label 的设置

4.2.3 色块检测的调用方式



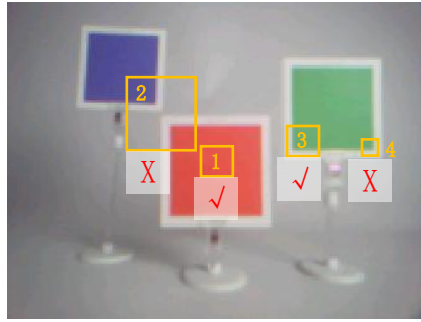
当检测到绿色色块时通过串口输出数据

色块检测在不更改设置的情况下只能检测一种选定的颜色，上图中检测绿色色块，那么小 MU 会寻找视野中符合绿色块的地方，返回结果最好的那一个，目前版本（V112）不支持多结果输出。如需检测其他颜色，只需更改颜色 label 标签即可，其余参数为默认值。

4.2.4 颜色识别区域的作用

在使用颜色识别算法时，需要指定一个颜色识别区域，算法运行后只返回该区域内的颜色信息。颜色识别区域小，像素少，速度快，但容易因各别像素的偏色而造成可信度的降低；颜色识别区域大，像素多，速度慢，但容易受背景的干扰。颜色识别区域应根据实际需求而合理设置。

颜色识别区域有 4 个参数可以设置：水平中心坐标（默认 50），垂直中心坐标（默认 50），识别区域宽度（默认 5），识别区域高度（默认 5）



四个检测区域对比

上图中有四个检测区域：

区域 1：结果可信，识别为红色

区域 2：结果不可信，区域中有多个颜色

区域 3：结果可信，虽然有白色边框干扰，但绿色占据主要部分，仍可识别为绿色

区域 4：结果不可信，识别区域很小，白色边框的干扰不可忽略

备注：颜色识别区域的大小只可在使用 Arduino 库开发时进行更改，可视化编程平台只提供坐标的设置。

4.2.5 颜色识别的调用方式

颜色识别输出结果并不是坐标和宽高，而是颜色的 label 标签以及 R 红，G 绿，B 蓝三个通道的值（范围 0~255）。在使用颜色识别时可以指定识别区域的坐标，其余参数为默认值。



识别图像中心区域颜色并通过串口输出



识别图像中心区域，判断颜色





当识别不同区域时，需要多次识别以确保结果可靠

4.3 球体算法与背景干扰

小 MU 的球体识别算法并不能识别任何球体，仅支持橙色乒乓球 (Label : 1) 和绿色网球 (Label : 2) 的识别。

球体算法分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	橙色乒乓球		2	绿色网球	

背景对球体识别的影响：任何算法的目标物体都容易受到背景的干扰影响，对于球体这种纯色的物体更是如此。在使用球体算法时，背景与球体应具有明显的对比度，光照应为白色光源，场地内不应出现与乒乓球和网球类似物体，以免误报。



高对比度，有利于识别



低对比度，不利于识别



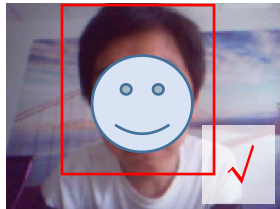
低对比度，不利于识别

球体算法的特殊性：球体算法分为搜索阶段和跟踪阶段，在球体未检测到时处于搜索阶段，此时处理速度相对比较慢，检测距离也会比较近。一旦检测到球体后，会进入跟踪阶段，处理速度会有所提升，检测距离也会变远。当球体丢失后，会再次进入搜索阶段。

4.4 人体算法的有效特征

人体检测实际是检测人体的上半身特征，上半身特征是指由头部（必须包含人脸）和胸部以上（肩膀）构成的特征组合，如果图像中具有符合人体上半身特征，那么会被认为是有人体被检测到。

当正对摄像头时具有最佳的检测角度：



近距离，Zoom1，30cm

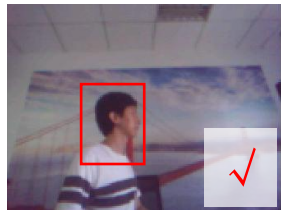


中距离，Zoom1，90cm



远距离，Zoom1，150cm

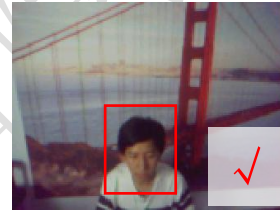
以下几种情况也可以检测到人体，但检测效果会受影响：



侧视



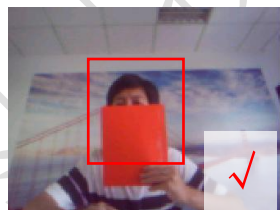
仰视



俯视



遮住衣物



遮住部分脸

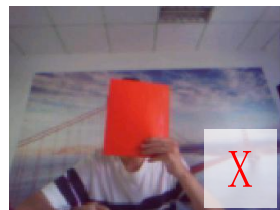


遮住头发

以下几种情况不满足上半身的特征要求，无法检测到人体：



背对摄像头



完全遮住脸



只有身体

4.5 卡片算法的有效角度

小 MU 具有 3 类卡片算法：形状卡片，交通卡片，数字卡片。小 MU 只能识别下述特定图案的卡片。在一幅图像中，同一个算法只能返回一个卡片的检测结果，但可以同时运行不同类型的卡片算法。

形状卡片分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	对号图案		2	叉号图案	
3	圆形图案		4	方形图案	
5	三角形图案		0	无效	

交通卡片分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	前进		2	左转	
3	右转		4	掉头	
5	停车		0	无效	

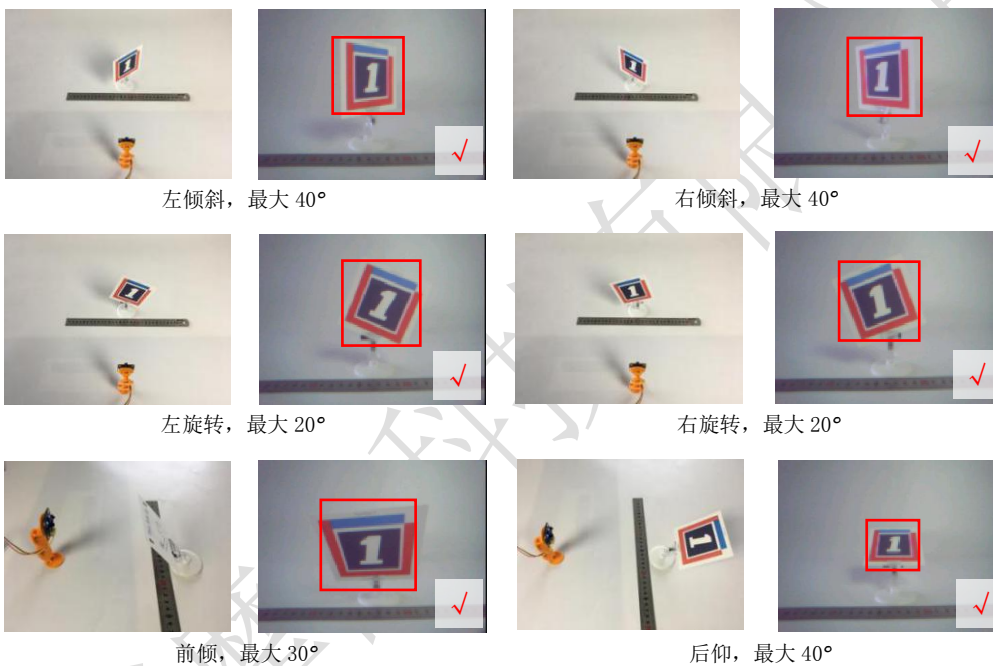
数字卡片分类标签：

分类标签	目标物体	图案	分类标签	目标物体	图案
1	数字 1		2	数字 2	
3	数字 3		4	数字 4	
5	数字 5		6	数字 6	
7	数字 7		8	数字 8	
9	数字 9		0	数字 0	

卡片识别角度：当卡片正对摄像头时具有最佳的检测视角，如果卡片倾斜，旋转、俯仰在一定角度范围内，也是可以识别的，但在这些情况下，所检测出来的数据会有一些的偏差。



最佳识别角度，正对摄像头



5 数据处理方法

原生数据：小 MU 直接输出的检测结果我们称之为原生数据。

利用原生数据只需经过简单的逻辑判断即可满足一般的应用需求，如果对这些原始数据通过数学方法进行一些简单的处理，那将产生更加丰富多彩的应用。本章节将深入介绍一些典型的小 MU 原生数据的处理方法。

5.1 判断是否检测到目标物体

最简单的小 MU 应用就是判断目标物体是否被检测到，当检测到目标物体时执行某个动作，否则执行另一个动作。

示意代码：



检测到人体时，串口输出“检测到人体”，否则输出“未检测到人体”

特殊技巧：



利用循环方式实现：直到检测到人体为止



利用循环方式实现：直到人体消失为止

5.2 分类标签 label 的使用

检测到目标物体后，可以通过读取分类标签 label 来判断具体属于哪一个分类。通过分类标签可以做更多细化的应用。

5.2.1 分类标签的基本调用方式



直接判断球体 Label 标签，如果为 1 表明是一个乒乓球，如果 Label 为 2 表明是一个网球

5.2.2 无人驾驶小车

交通卡片具有 5 个分类，分别是 1 前进，2 左转，3 右转，4 掉头，5 停车。通过调用可视化编程中的卡片模块可以很方便的进行判断，然后控制小车的行进。



利用交通卡片实现无人驾驶

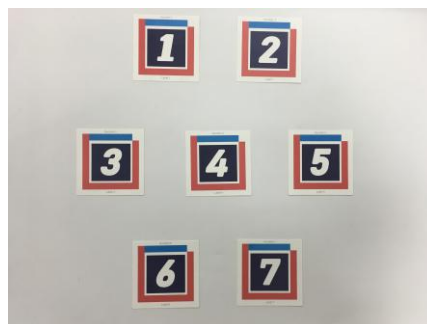
示意代码：



调用交通卡片模块，控制小车做出不同的行为

5.2.3 数字乐器演奏

利用数字 1~7 的卡片分别代表 7 个音符 Do,Re,Mi,Fa,Sol,La,Si 通过读取数字卡片的标签，然后直接赋值给播放音符的程序进行播放。



利用数字卡片实现乐器演奏

示意代码：

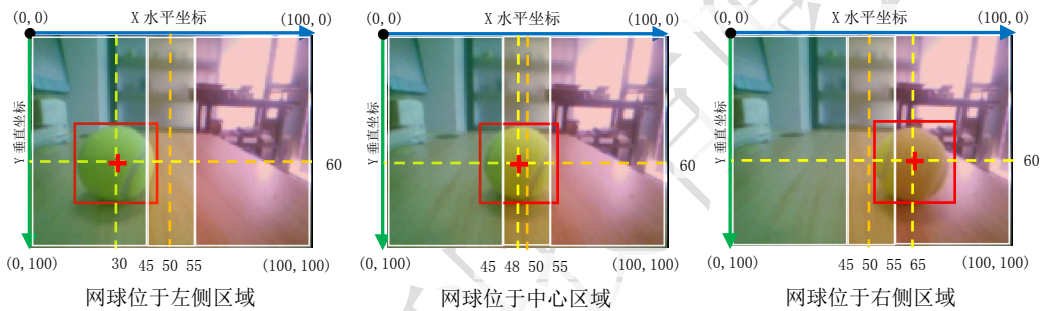


直接将标签结果赋值给函数使用

5.3 同一平面空间判断物体方位

同一平面空间：指被测物体与小 MU 处于相同的参考平面上，比如都处于桌面上，被测物体只在该平面上前后左右的运动，不会离开该平面进行上下的运动。

当目标物体的 X 水平坐标为 50 时，代表其位于小 MU 视野的正前方，当 X 水平坐标小于 50，位于 0~50 区间内时，代表其偏左，而 X 水平坐标大于 50，位于 50~100 区间内时，代表其偏右。

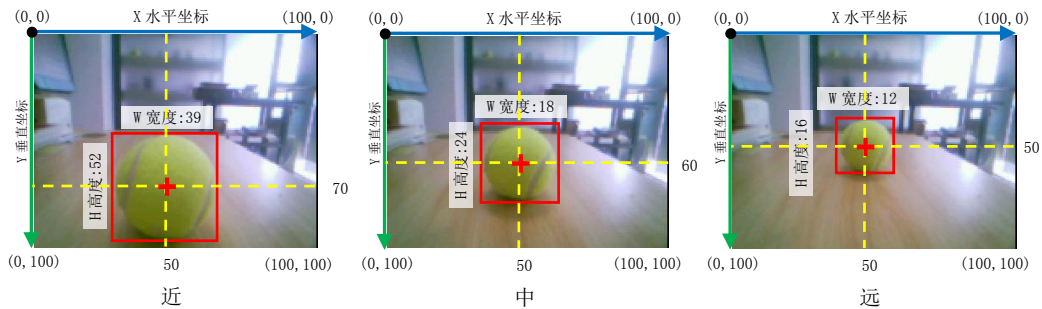


由于目标物体处于一个动态变化的过程，检测结果自身也可能存在一定的抖动，如果我们单纯的将 50 作为左右的分界点，那么其输出结果会忽左忽右来回振荡，为此需要做一个简单的消抖处理。我们可以将中心区域设定为一个区间，比如 45~55，当 X 坐标位于该区间内均认为是处于图像水平坐标的中心区域。相对应的左侧区域范围变为 0~45，右侧区域变为 55~100。



通过判断 X 水平坐标来控制左右转弯

5.4 同一平面空间判断物体远近



利用 W-H 宽高值判断远近：根据物体的透视原理，视野中的物体存在近大远小的现象，因此我们可以通过 W-H 宽高值来判断物体的远近关系，这个判断方法比较容易理解，也易于使用。该方法比较适合于小 MU 直视前方且物体较近的范围内使用，因为在此范围内物体的 W-H 宽高值变化会比较明显，但当物体较远时，W-H 宽高值的变化幅度就会变得比较小，难以准确分辨。

目标物体的远近关系需要根据其大小和应用需求而定，对于上图的网球而言，我们可以定义 $W < 35$ 为远， $W > 50$ 为近， $35 \leq W \leq 50$ 为适中距离。



通过判断 W 宽度来控制前进后退

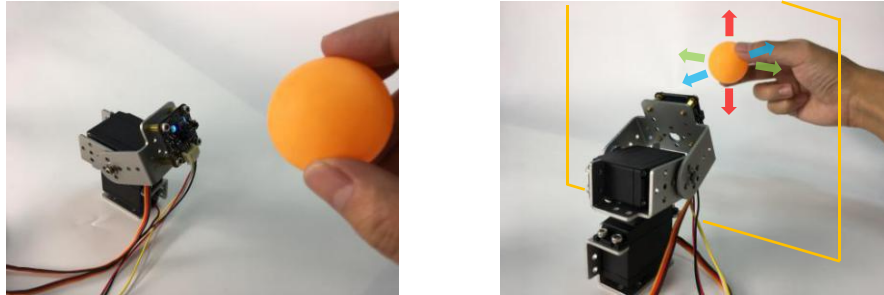
利用 Y 垂直坐标判断远近：可以根据 Y 垂直坐标来判断物体的远近，在上面的 3 个图中我们可以看出一个规律，物体靠近小 MU 时 Y 垂直坐标是偏下的（偏向 100 方向），反之远离小 MU 时，Y 垂直坐标会向图像上方移动（偏向 0 方向），该方法适合于小 MU 俯视角度使用。如果是仰视角度，情况相反，物体靠近小 MU 时 Y 垂直坐标是偏上的（偏向 0 方向），反之远离时偏下，此时适合于仰视角度。



俯视角度：通过判断 Y 垂直坐标来控制前进与后退

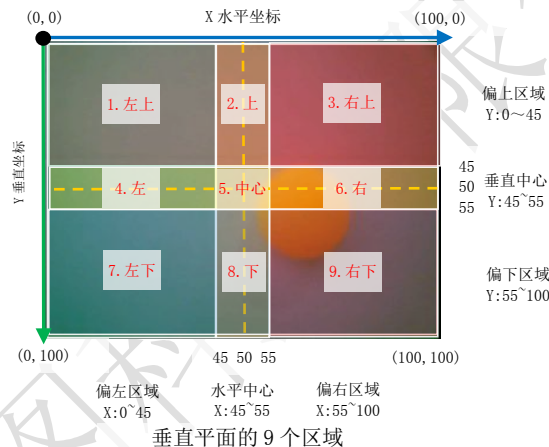
5.5 垂直平面空间判断物体方位

垂直平面空间：是指物体的运动平面与小 MU 呈垂直状态，即物体在小 MU 面前上下左右前后的运动，典型的应用就是舵机云台。



用小球控制舵机云台上下左右跟随

将水平方向和垂直方向各分割为 3 个区间，分别是水平方向的偏左区域(X:0 ~ 45)，水平中心区域(X:45 ~ 55)，偏右区域(X:55 ~ 100)，以及垂直方向的偏上区域(Y:0 ~ 45)，垂直中心区域(Y:45 ~ 55)，偏下区域(Y:55 ~ 100)，于是整个图像就被划分为了以下的 9 个区间：



控制一个双轴云台跟踪小球，其目的就是要将小球的中心坐标移动至“5 中心”位置。小球当前位置落于哪个区间，就控制舵机往那个区间方向移动，X 和 Y 方向分别每次移动一个小的角度，然后检测一次新的 X-Y 坐标，直至小球的 X-Y 坐标落于“5 中心”位置为止。

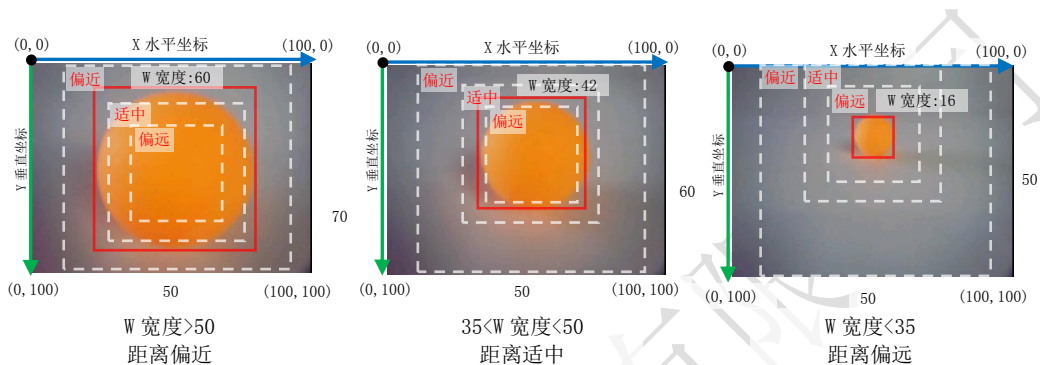


通过判断 X-Y 坐标来控制云台的转动

5.6 垂直平面空间判断物体远近

在垂直平面空间中，根据近大远小的透视原理，可以通过物体的宽度 W 或者高度 H 来判断远近，下面以 W 宽度为例，如果物体距离小 MU 比较近，那么物体在小 MU 中的成像就会比较大，检测到的 W 宽度值就比较高，反之如果物体比较远则成像较小 W 宽度值较小。

假设在一个装有小 MU 的小车上，如果我们想让小 MU 与一个乒乓球保持一定的距离，那么我们需要知道在这个距离上乒乓球在小 MU 视野中的 W 宽度。在乒乓球前后移动的过程中我们不断的检测乒乓球的 W 宽度值，如果宽度与该值一致，那么也就表明乒乓球移动到了我们所期待的一个距离上。



同样为了消除抖动，这个适当的 W 宽度值往往可以设定为一个区间，比如 $35 \sim 50$ ，在此区间内都被认为是合适的距离，而当 W 宽度小于 35 则认为距离较远，需要控制小车前进，如果 W 宽度大于 50 则任务距离太近，需要控制小车后退。

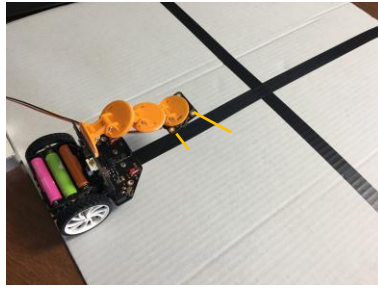


垂直平面空间中，通过判断 W 宽度来控制前进与后退

5.7 视觉循线-单线

利用色块检测可以方便的实现视觉循线功能，与传统红外循线相比，视觉循线除了跟随黑线外，还可以跟随彩色线，异形线等。下面以黑色线为例进行讲解。

(1) 小 MU 视角：为了实现较好的跟随效果，小 MU 视野中不宜出现其他外界干扰，如果线条比较细，则需要压低摄像头，如果线条比较粗，则可以适当抬高。小 MU 视角斜向下向前，不宜太垂直于地面，更不应往后看，目的是为了让小 MU 可以看到前方的路况做出预判。



小 MU 视角斜下向前看，视角比较低
备注：图中小车为 DFRobot 的麦昆小车（下同）

(2) 开启色块检测：开启色块检测算法时需要锁定白平衡，也可以使用 mode 按键来校准。为了避免小 MU 上 LED 灯光对道路的干扰，需要关闭 LED 显示。算法性能根据测试效果进行更改。为了提高检测帧率，我们需要使用 I2C 通讯，并开启高帧率模式。



用于单线循线的小 MU 配置

(3) 直线道路识别：当小 MU 视野中出现黑色道路时，会返回黑色块的坐标和大小，我们主要利用 X 水平坐标来判断小 MU 是否偏离了道路。



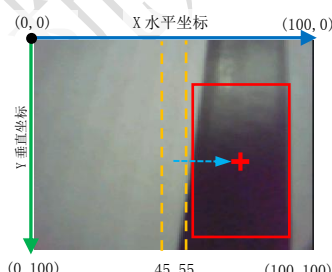
小车偏向左侧



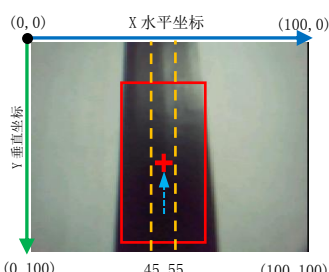
小车位于中心



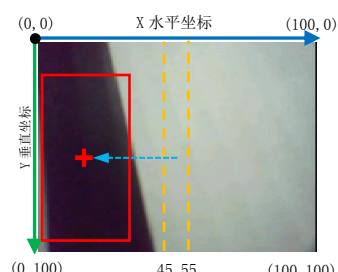
小车偏向右侧



黑色道路偏向右侧
控制小车右转



黑色道路位于中心
控制小车前进



黑色道路偏向左侧
控制小车左转

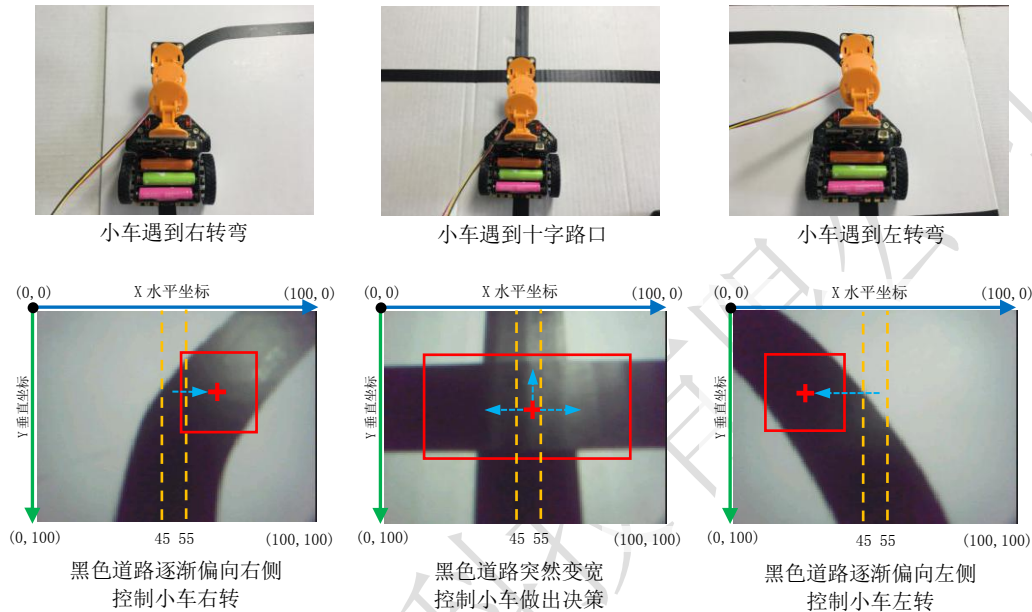
直行：我们首先需要规定哪里是小车需要跟随的道路中心，一般来说，我们会将图像的 X 水平坐标的中心值 $X : 50$ 作为小车的跟随中心。为了消除抖动，可以设置一个阈值范围 $X : 45 \sim 55$ ，也就是说，如果返回的黑色块的 X 坐标位于这个阈值之内，我们就认为道路是在我们的正

前方，此时我们可以控制小车向前直行；

直行偏左：在直行过程中，一旦小车偏向左侧，那么在小 MU 视野中，道路就跑到了图像的右边，返回的黑色块 X 坐标会位于道路中心的右侧，X 值将大于 55，所以当 $X > 55$ 时，我们就要控制小车右转一点角度，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内；

直行偏右：当小车偏向右侧时，X 值是小于 45 的，所以当 $X < 45$ 时，我们控制小车左转一点角度，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内。

(4) 转弯道路识别：同样主要依据黑色块的 X 水平坐标来判断转弯方向。



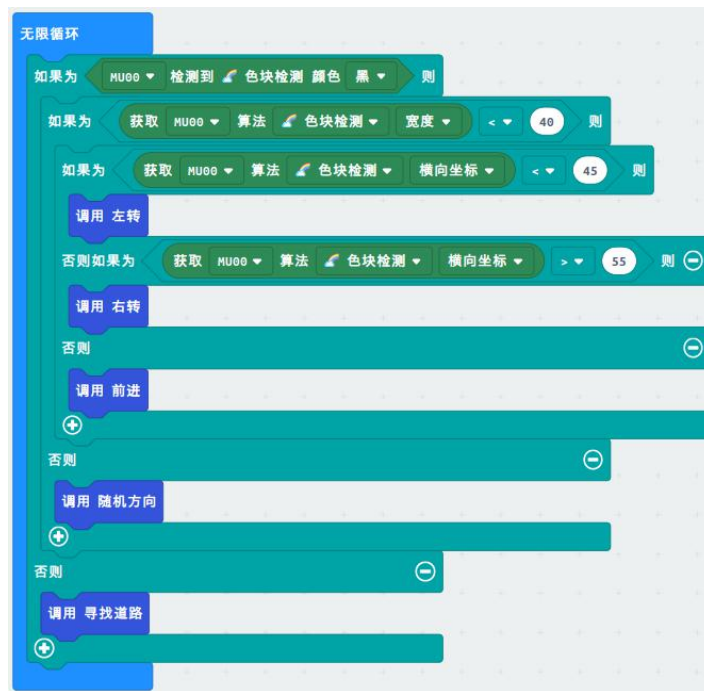
右转弯：当小车遇到右转弯时，小 MU 所检测到的黑色块 X 坐标会位于图像的右侧，当 $X > 55$ 时，控制小车右转，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内；

左转弯：当小车遇到左转弯时，小 MU 所检测到的黑色块 X 坐标会位于图像的左侧，当 $X < 45$ 时，控制小车左转，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内；

交叉路口：在十字路口，T 字路口等地方，在某一时刻所检测到的黑色块的 W 宽度会突然增加，这是因为检测到了横向延伸的色块，此时我们可以认为是遇到了一个交叉路口，至于小车该如何转向或是前行，则由用户自行控制。

(5) 示意代码：通过上述的分析，我们可以得出以下的控制方案

- 当检测到黑色块时，控制小车行进，否则寻找道路
- 当 $W < 40$ 时，小车在单条道路上（该参数需要根据实际调整）
- 当 $W \geq 40$ 时，遇到路口，控制小车做出行进决策
- 当 $X < 45$ 时，控制小车左转
- 当 $X > 55$ 时，控制小车右转
- 当 $45 \leq X \leq 55$ 时，控制小车前进



利用色块检测实现单线循线

5.8 视觉循线-模拟道路

对于模拟道路场景，小车在两个黑色边沿白色道路上行驶，利用小 MU 的色块识别功能，也可以方便的实现模拟道路的循线。我们以黑色边界，白色道路为例进行讲解。

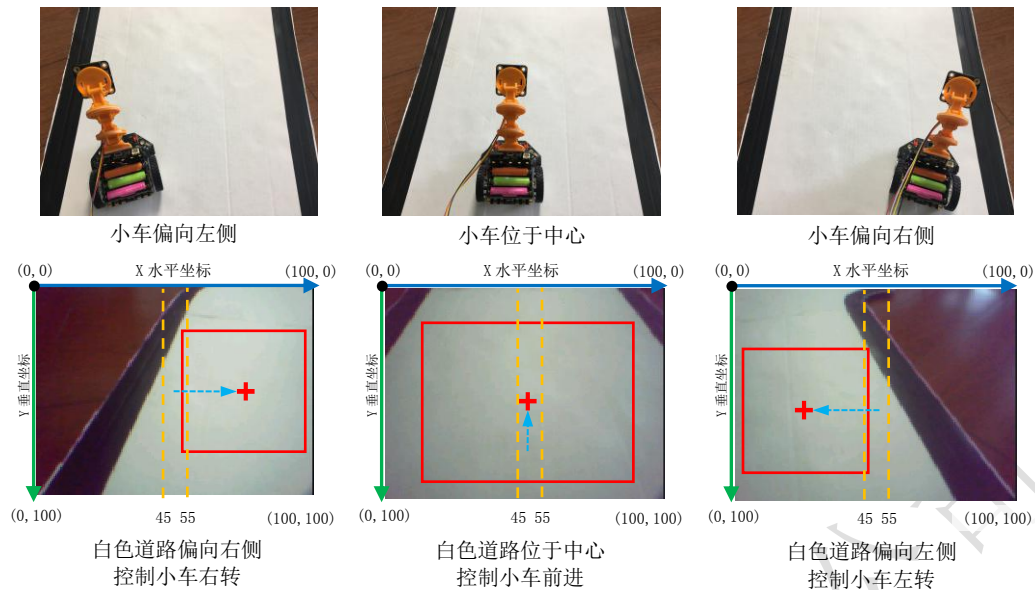
(1) 小 MU 视角：与单线循线不同之处，双线视角需要看到道路的全景，而模拟道路比单线要宽很多，所以小 MU 要抬高，斜下向前看，小车位于道路中心时，应该可以看到两侧的黑色边界。



小 MU 视角斜下向前看，视角比较高
备注：图中小车为 DFRobot 的麦昆小车（下同）

(2) 开启色块检测：与单线循线配置相同，开启色块检测算法时需要锁定白平衡，也可以使用 mode 按键来校准。为了避免小 MU 上 LED 灯光对道路的干扰，需要关闭 LED 显示。算法性能根据测试效果进行更改。为了提高检测帧率，我们需要使用 I2C 通讯，并开启高帧率模式。

(3) 直线道路识别：与单线循线原理一致，稍有不同的是，此时需要检测的是白色色块。当小 MU 视野中出现白色道路时，会返回白色块的坐标和大小，我们主要利用 X 水平坐标来判断小 MU 是否偏离了道路。

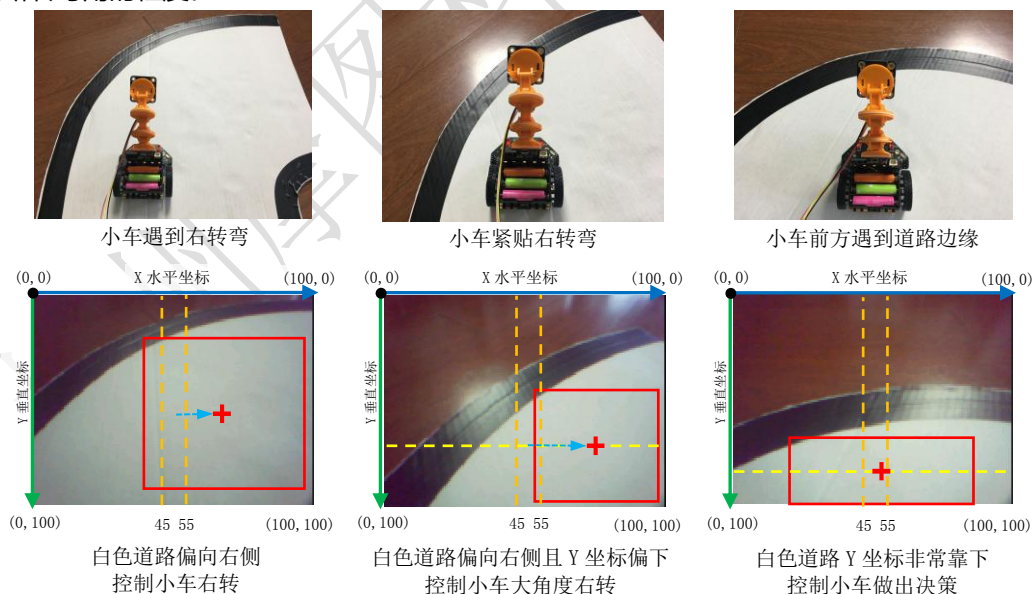


直行：设定中心区域为 $X : 45 \sim 55$ ，如果返回的白色块的 X 坐标位于这个阈值之内，我们就认为道路是在我们的正前方，此时我们可以控制小车向前直行；

直行偏左：在直行过程中，当小车偏向左侧时，白色块的 X 值将大于 55，所以当 $X > 55$ 时，我们就要控制小车右转一点角度，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内；

直行偏右：当小车偏向右侧时，白色块 X 值是小于 45 的，所以当 $X < 45$ 时，我们控制小车左转一点角度，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内。

(4) 转弯道路识别：主要依据白色块的 X 水平坐标来判断转弯方向，通过 Y 垂直坐标可以判断转弯角的程度。



右转弯 当小车遇到右转弯时，小 MU 所检测到的白色块 X 坐标会位于图像的右侧，当 $X > 55$ 时，控制小车右转，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内；

左转弯：与右转弯处理类似，小 MU 所检测到的白色块 X 坐标会位于图像的左侧，当 $X < 45$ 时，控制小车左转，直到 X 重新回到中心区域 $X45 \sim 55$ 范围内

贴近转弯边缘：当小车贴近右转弯边缘时，小 MU 所检测到的白色块会偏向图像的右下方，

此时除了 X 水平坐标大于 55 外，其 Y 垂直坐标也会下移（Y 值增大），代表距离道路边缘已经很近了，需要加大转弯角度，需速做出调整；

正前方遇到边缘：当小车正前方遇到边缘时，其 X 水平坐标虽然位于 45 ~ 55 的中心区域内，但是 Y 垂直坐标却十分靠下（Y 值很大），这是因为在边缘处白色区域都被压到了图像的下方。此时需要控制小车做出决策。

（5）示意代码：通过上述的分析，我们可以得出以下的控制方案

当检测到白色块时，控制小车行进，否则寻找道路

当 $X < 45$ ， $Y < 60$ 时，控制小车左转

当 $X < 45$ ， $Y \geq 60$ 时，控制小车加速左转

当 $X > 55$ ， $Y < 60$ 时，控制小车右转

当 $X > 55$ ， $Y \geq 60$ 时，控制小车加速右转

当 $45 \leq X \leq 55$ 时，且 $Y < 60$ 时，控制小车前进

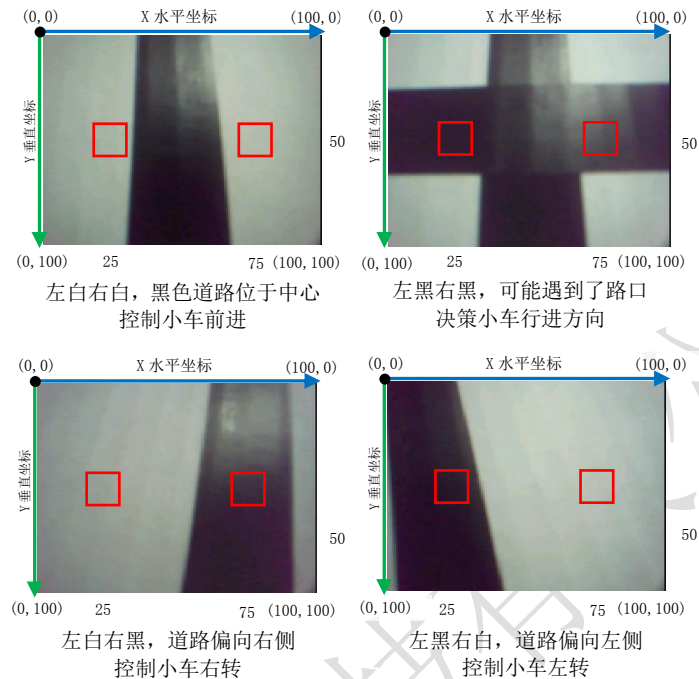
当 $45 \leq X \leq 55$ 时，且 $Y \geq 60$ 时，小车遇到边缘，控制小车做出决策



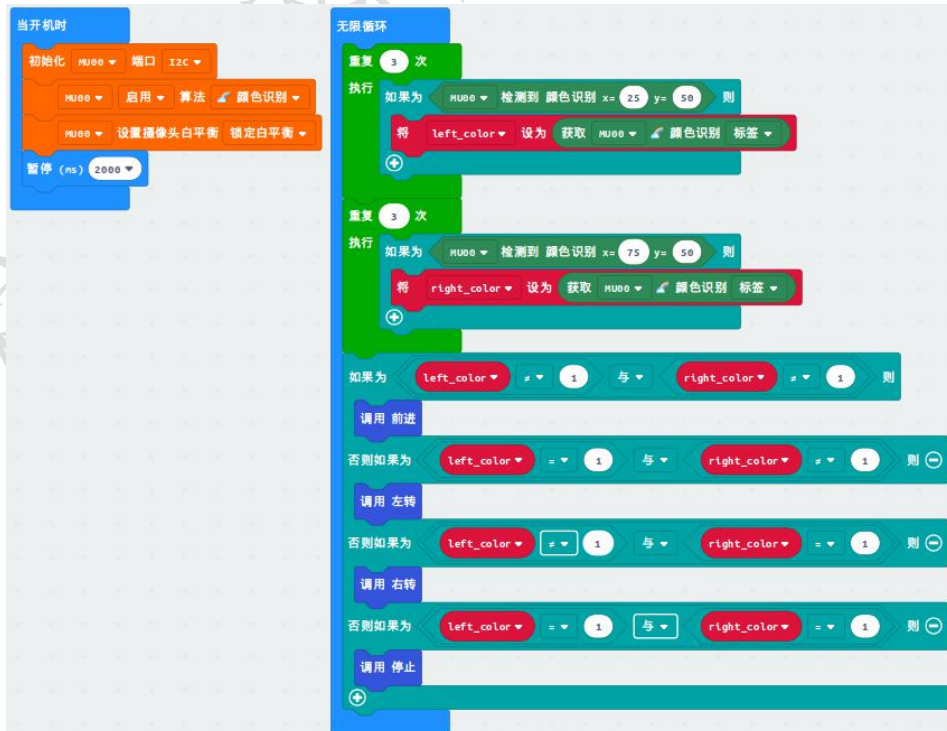
利用色块检测实现模拟道路循线

5.9 颜色识别实现循线

可以利用颜色识别算法进行循线，其原理与红外传感器循线原理一样，检测图像中的多个点位来判断道路是否偏移，下面以最简单的双点检测为例来说明其原理，实际中还可以使用多个点位来完成更复杂的道路判断。



- 示意代码：**
- (1) 根据道路的粗细程度，在图像左右两侧设置合适的位置设置两个虚拟的检测点；
 - (2) 开启颜色识别算法，并将检测区域坐标设置到左侧点位，识别颜色并记录；
 - (3) 将检测区域设置到右侧点位，识别颜色并记录，完成一次采样过程；
 - (4) 如果为左白右白：前进，左黑右黑：路口，左白右黑：右转，左黑右白：左转



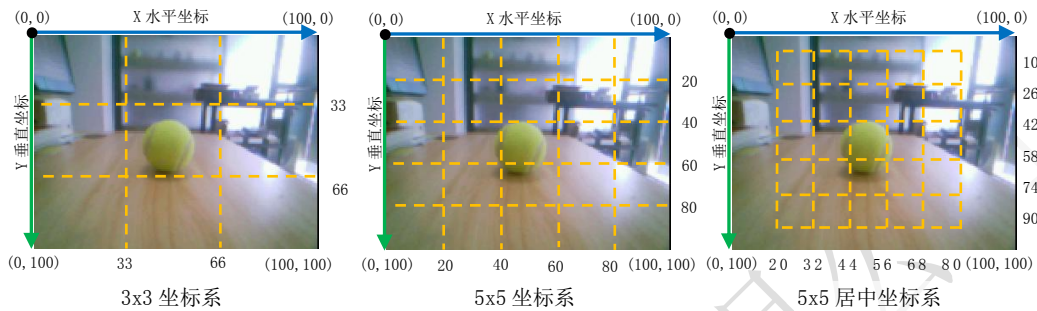
利用颜色识别左右 2 个点位，实现循线功能

5.10 二维空间区域分割

有时我们对于设计需求并不需要精确的参数(如 X12, Y41), 而只需要一个大概的趋势(位于图像左侧), 利用二维空间区域分割可以更好更快速的使用和理解小 MU 的检测结果。

5.10.1 区域分割的方法

二维空间区域分割就是将原本 100x100 的坐标系简化到一个 3x3, 5x5 或是其它的低分辨率坐标系中, 在计算过程中, 数值需要取整数:



3x3 坐标系: 将 X-Y 的值分别除以 33 取整即可, 坐标为 100 时可以忽略



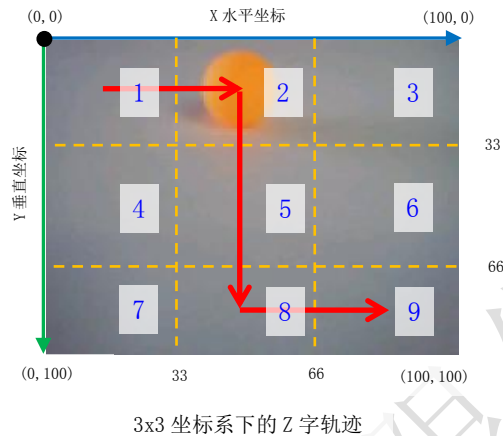
5x5 坐标系: 将 X-Y 的值分别除以 20 取整即可



5x5 居中坐标系: 将 X 值先减去 20 再除以 12 取整, 将 Y 值先减去 10 再除以 16 取整

5.10.2 轨迹识别

轨迹识别是利用二维空间区域分割来实现的，我们以 3x3 坐标系为例，讲解一个通过视觉轨迹实现密码锁的应用。假设我们要用小球的移动来匹配图中的“Z”型密码，类似于智能手机中的图案解锁功能，小球的移动轨迹与图案一致则解锁成功，否则失败。

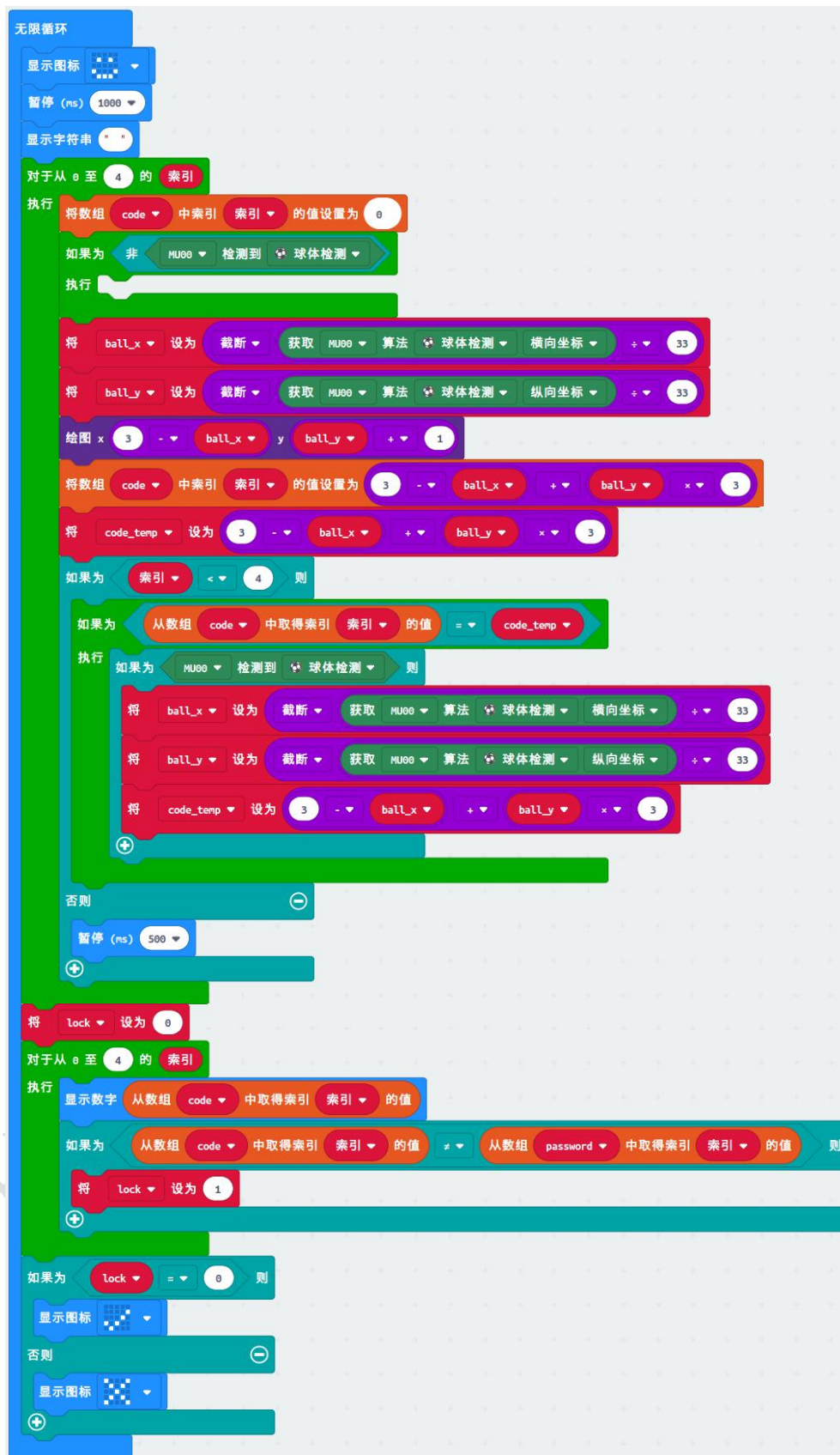


示意代码：

- (1) 按 3x3 将图像分割为图中的 9 个区域，区域 1~9 的坐标分别为 1(X0,Y0), 2(X1,Y0), 3(X2,Y0), 4(X0,Y1), 5(X1,Y1), 6(X2,Y1), 7(X0,Y2), 8(X1,Y2), 9(X2,Y2)；
- (2) 当首次检测到小球坐标时，则记录下当前的区域编号，此编号为起始编号；
- (3) 如果小球坐标始终位于同一个区域，则不重复记录编号，直到小球的坐标区域发生变动；
- (4) 小球坐标区域变动后，则记录第二个区域编号，依此类推，直到检测到 5 个编号为止；
- (5) 将这 5 个编号与我们所需求的“Z”字编号 (1, 2, 5, 8, 9) 进行匹配，如果顺序和编号完全一致则识别成功，否则失败。



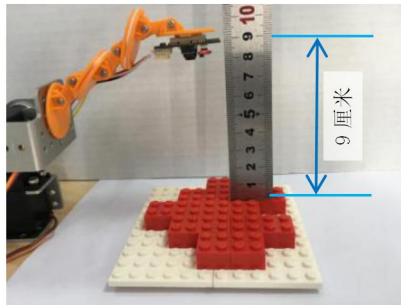
设置 Z 字轨迹密码 “12589”



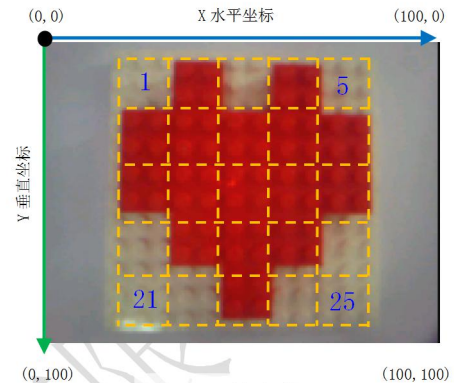
识别轨迹，显示数值，判断是否与密码一致

5.10.3 图案扫描

图案扫描是利用二维空间区域分割来实现的，我们以 5x5 居中坐标系为例，讲解一个心形图案的扫描的应用。为了实现较好的识别效果，我们需要将小 MU 固定在一个位置保持不动，而且要有一些标准的参照物。本案例用了标准的乐高积木块来做演示，利用积木块拼出一个心形和二维码造型的图案，然后将小 MU 固定在图案的上方，小 MU 的 Pcb 边缘距离乐高积木块表面约 9 厘米的位置（可以根据实际检测效果去调整），此时乐高图案在小 MU 中的成像如图所示。

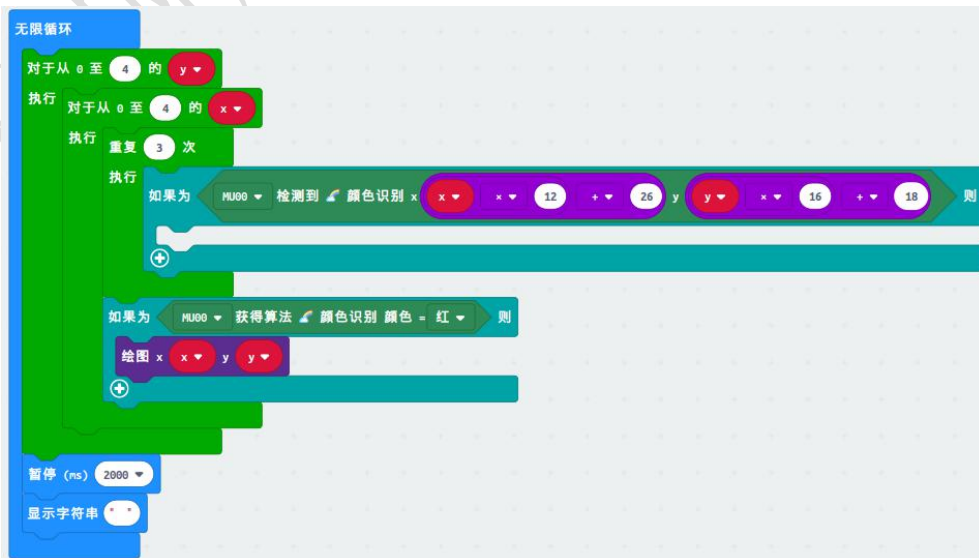


小 MU 与乐高图案的相对位置



心形图案扫描

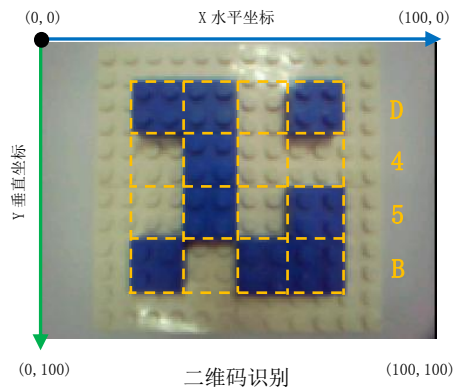
- (1) 开启颜色识别算法，校准并锁定白平衡；
- (2) 从左上角的第一个格子区域 1 开始，设置颜色识别区域的坐标至该区域中心；
- (3) 识别该区域颜色 label 分类标签号，为了提高可靠性连续检测 3 次，记录下颜色 label 分类标签；
- (4) 设置颜色识别区域的坐标至右侧的区域 2，并按照步骤 3 记录结果，依此类推，直到 25 个区域全部统计完毕，最后会得到一系列的 label 数组；
- (5) 对于该心形图案，有效数据是红色 label3，所以我们可以串口打印一个 5x5 的矩阵在屏幕上，其中在 label3 的位置打印*号，其他 label 输出空格；
- (6) 如果使用 MicroBit，还可以将图案在屏幕上显示出来，其中 label3 的地方点亮对应位置的 LED，其余 label 关闭对应位置的 LED；
- (7) 该方法也可以扫描手绘的图案，也可以用 10x10 的矩阵去扫描。



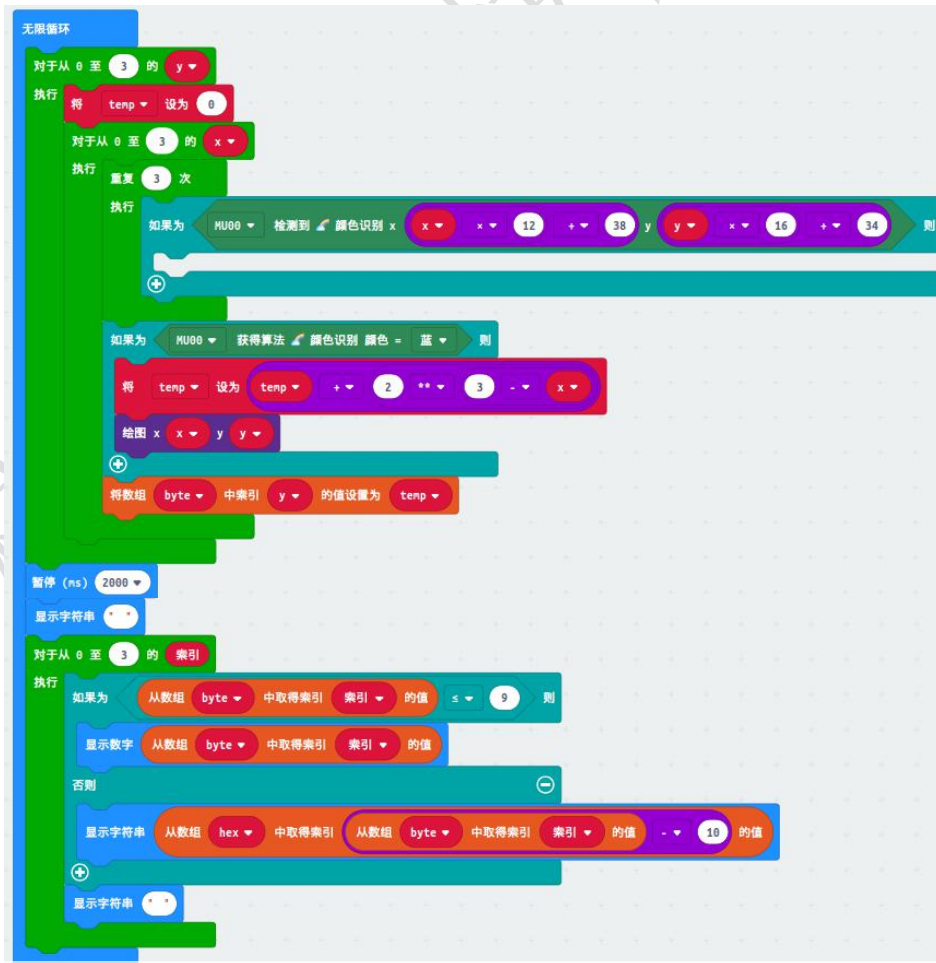
在 5x5 矩阵内检测红色色块，并在 LED 屏幕对应位置上显示

5.10.4 图案识别

图案识别是利用二维空间区域分割来实现的，我们以 4x4 居中坐标系为例，讲解一个自定义二维码识别的应用。小 MU 的安放与上一节图案扫描的方法一致。



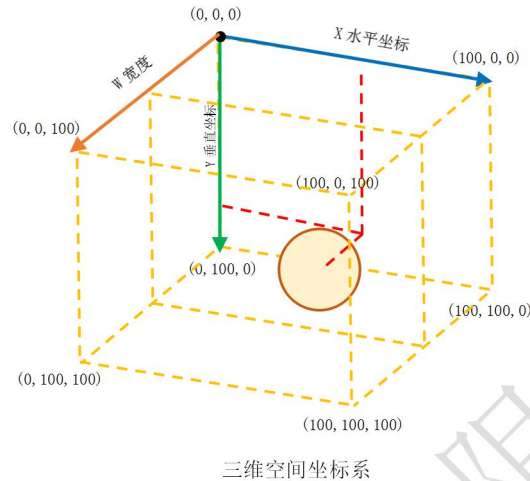
- (1) 开启颜色识别算法，校准并锁定白平衡；
- (2) 从左上角的第一个格子区域 1 开始，设置颜色识别区域的坐标至该区域中心；
- (3) 识别该区域颜色 label 分类标签号，如果为蓝色记录为 1，其他颜色则记录为 0；
- (4) 设置颜色识别区域的坐标至右侧的区域 2，并按照步骤 3 记录结果，依此类推统计完第一行的 4 个数据，会得到“1101”的数据，将该数据转换为 16 进制表示，即为“D”；
- (6) 解码其余的 3 行数据，那么这个简单的二维码数据为“D45B”；
- (7) 我们可以定义这串字符的含义是“开门”，于是我们有了一个门禁卡片系统。



在 4x4 矩阵内检测蓝色色块，每行结果转换为十六进制并显示出来

5.11 三维空间坐标系

通过 X-Y 坐标和 W 宽度值即可构建一个三维空间坐标系，在三维空间，会比二维空间多一个维度的信息表达，可以创造出很多奇妙的应用，比如三维空间密码锁，三维空间乐器，三维空间绘画等。



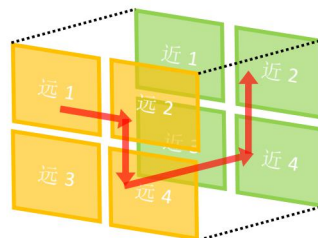
三维空间坐标系

三维空间可以采用二维空间区域分割的方法，将三维空间分割出不同的区域块，通用的示意代码如下：



三维空间坐标系的通用示意代码

5.11.1 三维空间密码锁



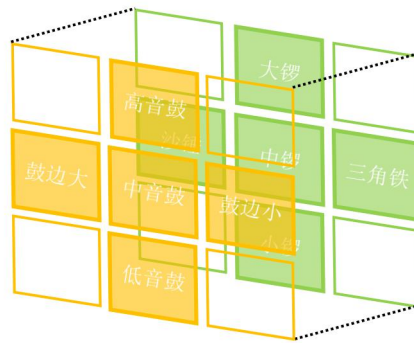
在三维空间中的密码轨迹

(1) 将 X-Y 坐标系进行 2x2 区域分割，将 W 宽度以 50 为分界点分为远(51~100)，近(0~50)两个区间，于是三维空间中存在 8 个象限区间：远 1，远 2，远 3，远 4，近 1，近 2，近 3，近 4；

(2) 可以利用小球或者卡片，按照轨迹识别的方法在三维空间中划过 5 个点位并采样；

(3) 如果划过的轨迹与图中一致(远 1>远 2>远 4>近 4>近 2)则表明密码正确，可以开锁。

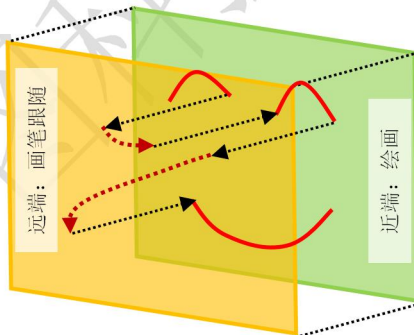
5.11.2 三维空间乐器



三维空间乐器的定义

- (1) 将 X-Y 坐标系进行 3x3 区域分割, 将 W 宽度以 50 为分界点分为远(51 ~ 100), 近(0 ~ 50)两个区间, 于是三维空间中存在 18 个象限区间;
- (2) 远端和近端只使用“上下左右中”五个区间, 每个区间放置一种乐器音乐;
- (3) 可以利用小球或者卡片, 当坐标出现在相对应的区间上时则发出对应的音乐;
- (4) 如果持续检测到, 则只发出 1 次音乐, 如果区域发生变化或是目标物体消失后又重新检测到, 且连续检测到稳定的 3 帧, 则播放对应的音乐;
- (5) 还可以根据目标物体的 label 不同, 增加额外的属性, 比如根据数字卡片的数字来表达音量的大小或音域的高低。

5.11.3 三维空间绘画



在三维空间中绘画

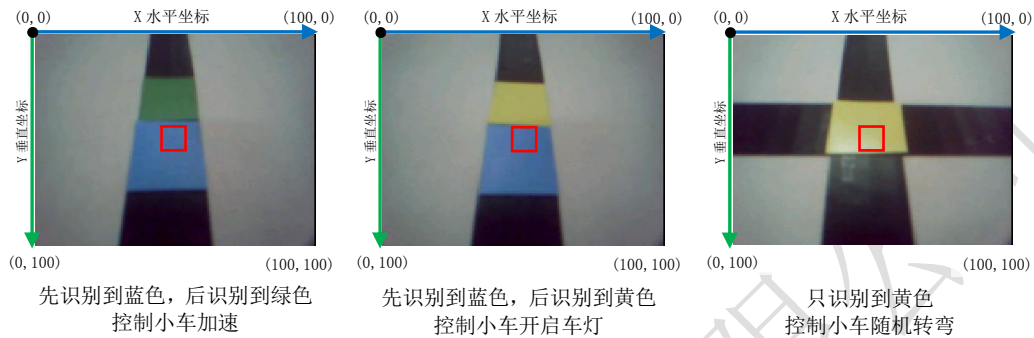
- (1) 将 W 宽度以 50 为分界点分为远(51 ~ 100), 近(0 ~ 50)两个区间, 于是三维空间中存在近端平面和远端平面;
- (2) 用小球作为画笔, 在远端平面上只跟随画笔, 但并不在屏幕上绘画;
- (3) 当小球移到近端平面时, 表明画笔被按下, 在屏幕上同步显示坐标轨迹。

5.12 同时开启多种算法

小 MU 支持同时运行多个算法，与运行单个算法相比，算法的运行时间会变长，如果算法不是必须同时工作的，那么可以灵活的调用算法的开启和关闭。开启多种算法可以创造更多精彩的应用。

5.12.1 循线加上颜色识别

在色块检测算法的基础上增加颜色识别算法，于是可以利用颜色贴纸来实现小车的高级控制。当小车前进过程中遇到不同的颜色或颜色组合时，作出不同的反应，例如：



识别颜色，如果是先蓝后绿则加速，如果是先蓝后黄则开启车灯

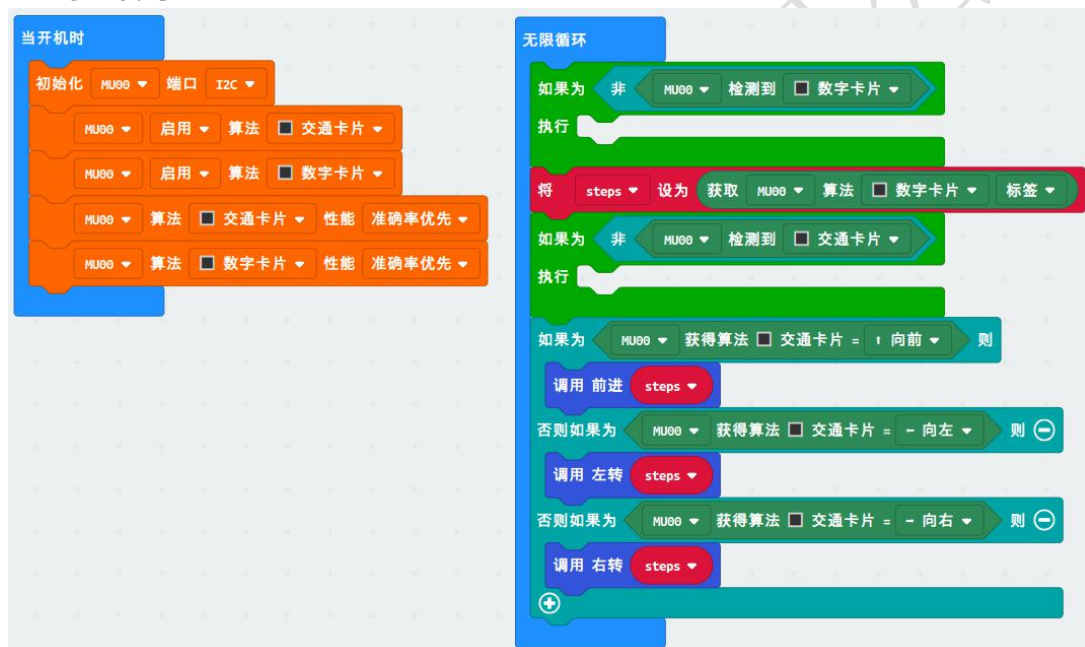
5.12.2 多类卡片同时识别

卡片在准确率优先模式下，可以实现不同类型卡片的同时识别，例如同时识别交通卡片和数字卡片：



利用“前进”和“数字5”表达前进5步的含义

示意代码：

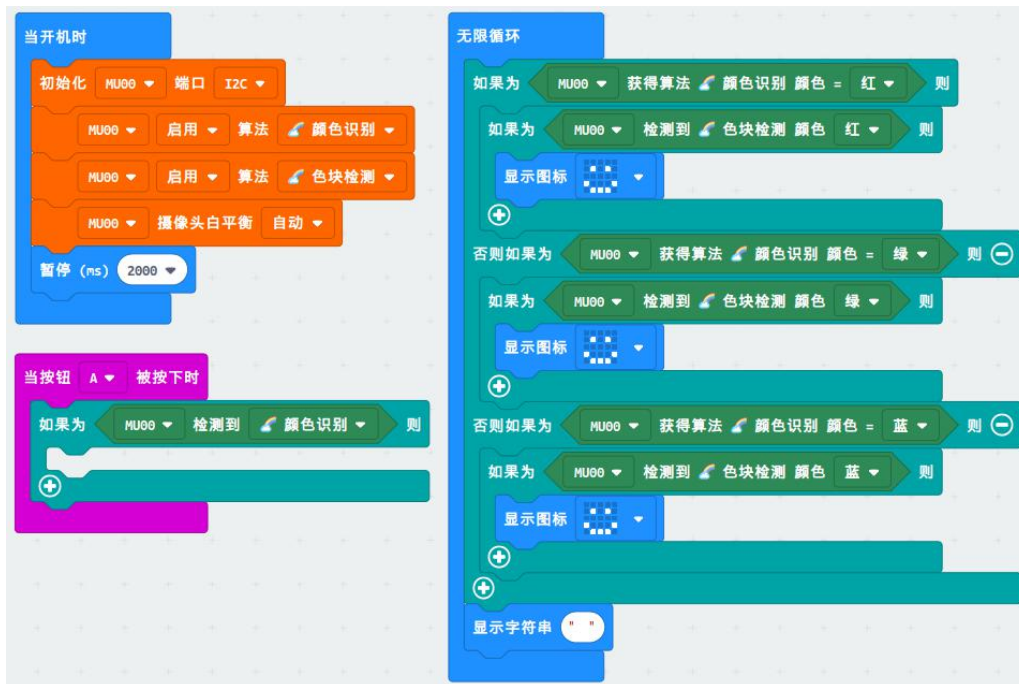


同时识别两种卡片

5.12.3 利用颜色识别物体

对于具有单一颜色的物体，可以先用颜色识别，后用色块检测的方法来实现对这个物体的识别或跟踪，比如黄色的香蕉，红色的袋子，蓝色的瓶子等等，当然，背景中不能有相同颜色的干扰。

示意代码：



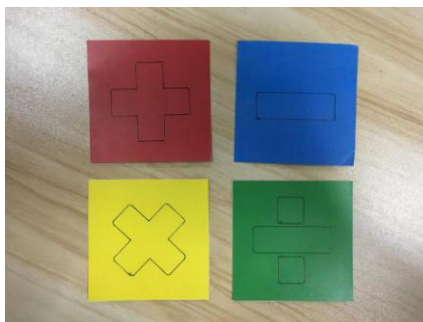
当按下 A 键，则识别颜色，然后在主循环中开始检测相应的颜色物体

5.13 算法的衍生

小 MU 无法提供所有满足用户需求的算法，但可以利用现有算法间接的创建一个“新算法”。

5.13.1 利用颜色识别来实现加减乘除

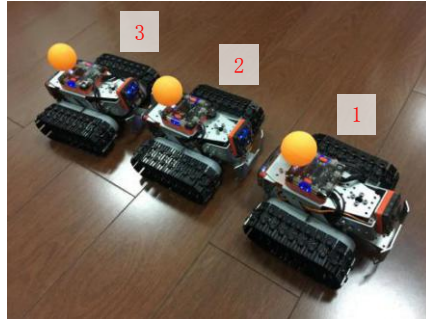
小 MU 本身并不会识别“加减乘除”的运算符号，但可以利用颜色卡片来代替这四种卡片，只需要在这四种颜色卡片上绘制上相应的图案即可，然后利用颜色识别来判断是哪一个运算符号。下面的图片中，利用红色来表示“加号”，蓝色表示“减号”，黄色表示“乘号”，绿色表示“除号”。类似的，我们也可以表达“石头，剪刀，布”。



利用颜色卡片制造的“加减乘除”运算卡片

5.13.2 利用小球实现任意物体跟踪

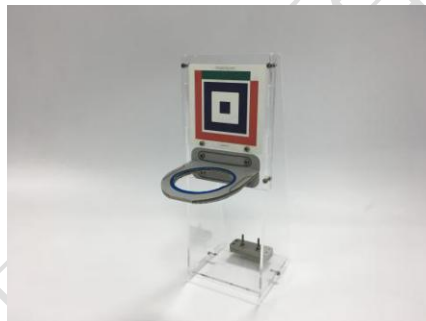
如何让小 MU 跟踪另外一个小车子行进，可以在前面的小车上放一个小球，后面的小车去识别和跟踪这个球即可。在下面这个图中，1 号小车在前面通过遥控或者程序控制其行进路线，后面的 2 号小车通过识别 1 号小车后面的小球来进行跟踪，3 号小车则跟踪 2 号小车后面的小球，于是三台小车就组成了一个自动驾驶的车队，间接的实现了小车的跟踪。



利用乒乓球作为标识物，实现对任意物体的跟踪

5.13.3 利用形状卡片表达一个篮球框

想要做一个篮球机器人，但是没有篮球框的识别算法，那么可以利用一个形状卡片来代替一个篮球框的位置，控制机器人根据这个形状卡片来判断投篮的位置。



利用形状卡片定义一个篮球框

5.14 数据的特殊处理

图像检测的数据结果不一定是稳定不变的，哪怕小 MU 和目标物体的相对位置都没有发生变化，也可能存在小范围内的数据抖动，这都是图像检测中不可避免的现象。小 MU 在数据输出前做过一些数据处理，但如果仍无法满足用户的使用需求，可以尝试下面的一些简单而实用的数据处理方法来提高数据的稳定性与可靠性。

5.14.1 提升检测结果的可靠性

为了消除偶然的误报或是漏报问题，我们可以对检测结果进行数据统计，当统计结果达到某个阈值时才更新相应的结果。以人体检测为例：

(1) 统计人体的检测到人体的次数和未检测到人体的次数，分别通过变量 `detected_count` 和 `undetected_count` 变量来统计；

(2) 当 `detected_count` 次数大于 5 次时，则表明确实检测到了人体，此时将检测标志变量 `detect_flag` 置 1 并清空 `undetected_count` 数据；

(3) 反之则当 `undetected_count` 次数大于 5 次时，则表明人体确实消失了，此时将检测

标志变量 detect_flag 置 0 并清空 detected_count 数据；

(4) 通过 detect_flag 来判断串口输出内容。



利用数据统计来提高检测结果可靠性

5.14.2 消抖措施

检测数据会存在抖动，比如 X 水平坐标期望值为 50，而实际检测结果可能为：49，50，48，51，50，50，49，51……。假设一个舵机在根据这样数值在做出不断的调整，那么就会产生左右抖动现象，此时可以设置一个阈值，只有当 X 水平坐标与上次的结果大于该阈值时，才会动作，而小于这个阈值的数据维持不变。



利用阈值来消除抖动

5.14.3 快速跟踪

在做跟踪时，有时 X 水平坐标会距离中心点 X50 很近，有时很远，如果我们在转向跟踪时采用相同的处理速度，那么对于近距离会出现转向过大，而远距离会出现转向不足的现象，此时可以根据 X 距离中心的差值来分别处理。

下例中，当 $X < 30$ 时快速左转，当 $30 \leq X < 45$ 时正常左转，而当 $X > 70$ 时快速右转，当 $70 \leq X < 85$ 时正常右转，位于 $45 \leq X \leq 70$ 之间则保持不变。



根据 X 的区间执行不同的程序

5.14.4 施密特触发

一般来说，触发条件是“0”和“1”的区别，例如当人体 W 宽度 > 50 则开门， $W \leq 50$ 则关门。对于这样的逻辑看似是没有问题的，但在实际使用时，由于 W 在 50 周围的抖动原因，导致当 W 在 50 附近时，门会处于频繁的开启和关闭切换状态，系统变得不可靠。

施密特触发具有两个独立的触发条件，一个是开启条件，一个是关闭条件，在这两个参数之间是一个缓冲区间，状态维持不变。例如将 $W > 50$ 作为开门条件，而关门条件设为 $W < 30$ ，30 ~ 50 之间是缓冲区。这样一来，即便 W 在 50 或是 30 附近有抖动，也不会造成门的频繁开启和关闭，因为他们的抖动区间达不到改变门状态的触发条件。



施密特触发原理



利用施密特触发控制门的开启与关闭

5.14.5 排除异常数据

对于平稳运动的物体而言，理论上是不应该存在数据跳变的，如果新的结果与老的结果发生了较大的数据跳动，则表明是一个异常数据，用下面的方法可以简单的排除异常数据。下面的例子中，当人体的 X 水平坐标与上一次的检测结果 `old_x` 相差在 10 以内，则认为是合理的数据并更新 `old_x`，反之则是异常数据，不做处理。



简单的排除异常数据

6 服务与支持

至此，相信您对小 MU 视觉传感器有了一个更深层的认识，脑海中也诞生出无数奇妙的想法，迫不及待的想去尝试一番。如果手册中仍有不足之处，希望您能谅解，也希望得到您的反馈，意见与建议。更多的资料和信息请关注：

摩图科技官网：

<http://www.morpx.com>

小 MU 服务与支持页面：

<http://mai.morpx.com/page.php?a=sensor-support>

Github：

<https://github.com/mu-opensource/>

技术与支持邮箱：

support@morpx.com

摩图科技电话：

(0571) 8195 8588



官方技术支持微信号



官方技术支持 QQ 号

修订历史

日期	版本	发布说明
2019.08.10	V0.1	起草
/	/	/

杭州摩图科技有限公司