

实验四 目标颜色识别

1 实验目的:

- ① 掌握色度学基础知识及颜色空间转换的基本原理
- ② 掌握轮廓检测算法实现对目标进行检测和分割
- ③ 掌握基于色度模型和阈值分割的颜色识别技术

2 实验仪器

- ① 硬件: 计算机, EAI 机器人
- ② 软件: ROS、OpenCV 库

3 实验原理

3.1 色度模型

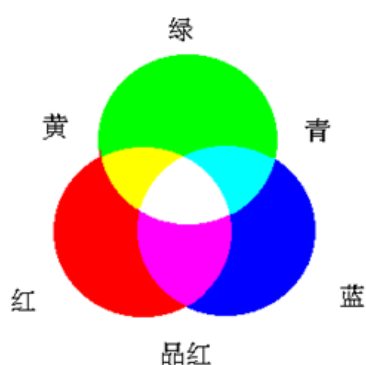


Fig 1 RGB 模型

RGB 模型与人眼强烈感知红绿蓝三色的事实能够很好地匹配,但是不能适应实际上人们对颜色的解释,当人们观察一个彩色物体时,用色调 (Hue)、饱和度 (Saturation) 和亮度 (Intensity) 描述,色调是描述一种纯色的颜色属性,饱和度是一种纯色被白光稀释的程度的度量,亮度是一个主观描述子,体现了无色的强度概念,这三者组成了 HSI 色度模型。该模型能够在彩色图像中从携带的彩色信息中消去强度分量的影响,因此是开发基于彩色描述的图像处理算法的理想工具。

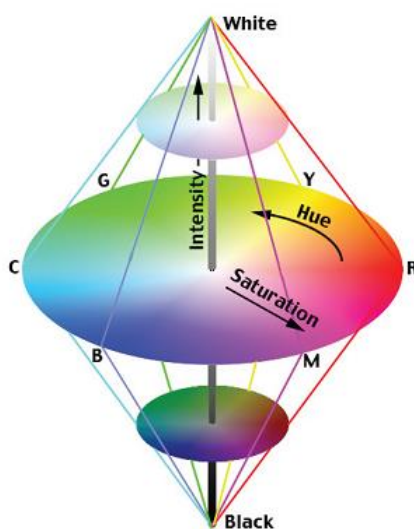


Fig 2 HSI 色度模型

给定一幅 RGB 彩色格式的图像,转换到 HSI 空间的公式如下:

$$\begin{cases} I = \frac{1}{3}(R+B+G) \\ S = 1 - \frac{3}{R+B+G}[\min(R+B+G)] \\ H = \arccos \left\{ \frac{[(R-G) + (R-B)]/2}{[(R-G)^2 + (R-B)(R-B)]^{1/2}} \right\} \end{cases} \quad (1-1)$$

与 HSI 空间类似, HSV 空间也是一种常用的色彩描述空间, 是根据颜色的直观特性由 A. R. Smith 在 1978 年创建的一种颜色空间, 也称六角锥体模型。

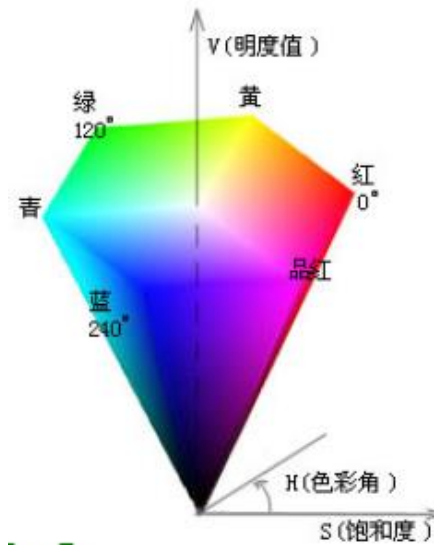


Fig 3 HSV 色度模型

给定一幅 RGB 彩色格式的图像, 转换到 HSV 空间的公式如下:

$$\begin{aligned} v &= \max(r, g, b) \\ s &= \frac{v - \min(r, g, b)}{v} \\ 6h &= \begin{cases} 5 + b' & \text{when: } r = \max(r, g, b), g = \min(r, g, b) \\ 1 - g' & \text{when: } r = \max(r, g, b), g \neq \min(r, g, b) \\ 1 + r' & \text{when: } g = \max(r, g, b), b = \min(r, g, b) \\ 3 - b' & \text{when: } g = \max(r, g, b), b \neq \min(r, g, b) \\ 3 + g' & \text{when: } b = \max(r, g, b), r = \min(r, g, b) \\ 5 - r' & \text{else} \end{cases} \end{aligned} \quad (1-2)$$

3.2 阈值分割

利用色度模型将 RGB 图像转换到其他空间后, 获取了原图的彩色信息, 由图 1 和图 2 可知, 不同颜色占据着不同的颜色空间, 因此我们需要对图像中的像素进行取舍和决策, 直接剔除一些低于或者高于一定值的像素。若给定上下界 thresh_max 和 thresh_min 的条件下, 阈值分割后的像素值 $dst(x, y)$ 满足:

$$dst(x, y) = \begin{cases} 255 & thresh_min < dst(x, y) < thresh_max \\ 0 & else \end{cases}$$

对转换后的三通道进行阈值分割，以 HSV 空间为例，不同颜色对应的三通道阈值如表 1-1 所示：

表 1-1 HSV 空间颜色分割阈值参考表

	黑	灰	白	红		橙	黄	绿	青	蓝	紫
H_min	0	0	0	0	156	11	26	35	78	100	125
H_max	180	180	180	10	180	25	34	77	99	124	155
S_min	0	0	0	43		43	43	43	43	43	43
S_max	255	43	30	255		25	255	255	255	255	255
V_min	0	46	221	46		46	46	46	46	46	46
V_max	46	220	255	255		255	255	255	255	255	255

通过设置合适的阈值对图像进行处理，能够分别得到不同的颜色在原图中的对应区域，如下图所示。

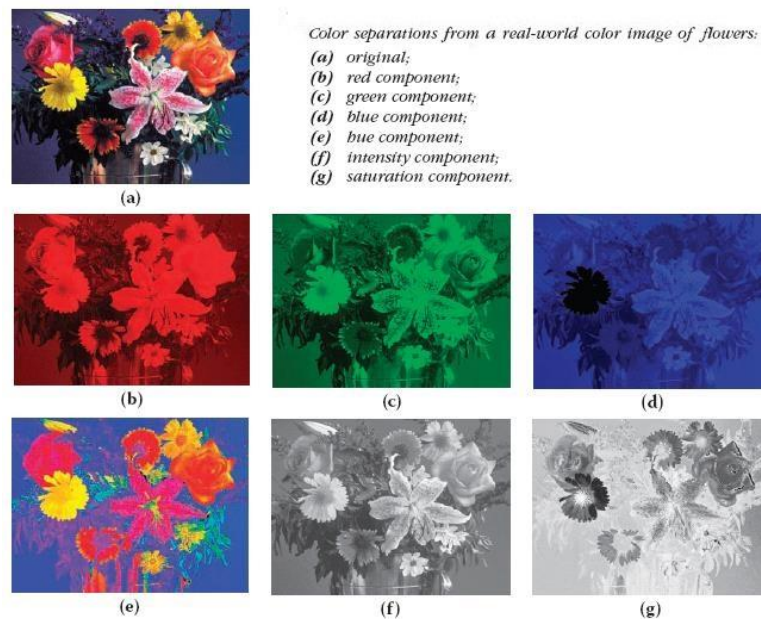


Fig 4 色彩分割结果示意图

3.3 轮廓检测

边缘检测虽然能够根据像素之间的差异，检测出轮廓边界的像素，但是它并没有将轮廓作为一个整体，而霍夫变换虽然能够实现直线、圆等简单轮廓的检测，但是功能单一。1985 年，satoshi suzuki 发表了一篇论文《Topological structural analysis of digitized binary images by border following》，通过分析数字二值化图像的拓扑结构，实现了对二值图的轮廓检测，基于该文章的思想，Opencv 中提供了 findContours()函数来寻找图像中物体的轮廓，并结合 drawContours()函数将找到的轮廓绘制出。

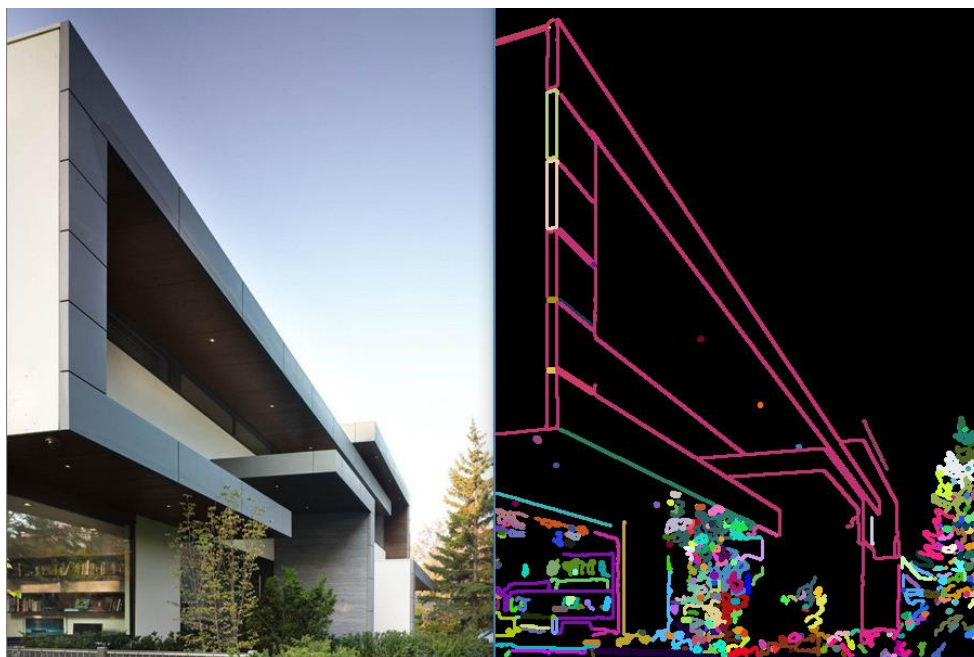


Fig 5 轮廓检测结果示意图

实际应用中,为了对轮廓进行分割,需要将检测到的轮廓用多边形表示出来,既根据轮廓提取出图像的多边形区域,常见的函数包括:

- 1) `boundingRect()`: 返回外部矩形边界;
- 2) `minAreaRect()`: 寻找最小包围矩形;
- 3) `minEnclosingCircle()`: 寻找最小包围圆形;
- 4) `approxPolyDP()`: 逼近多边形曲线

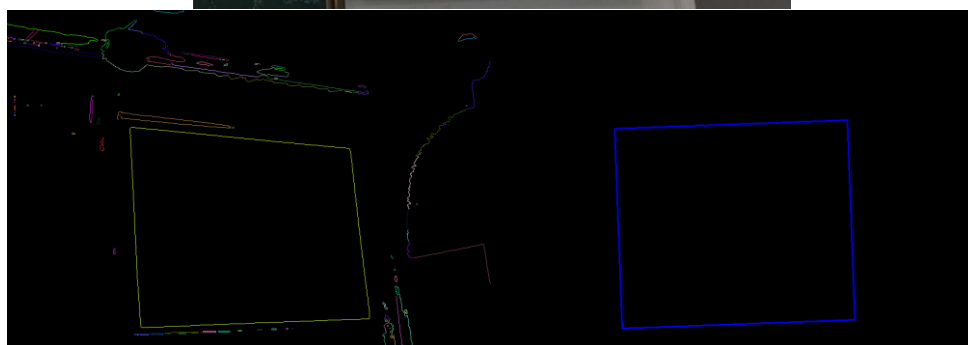
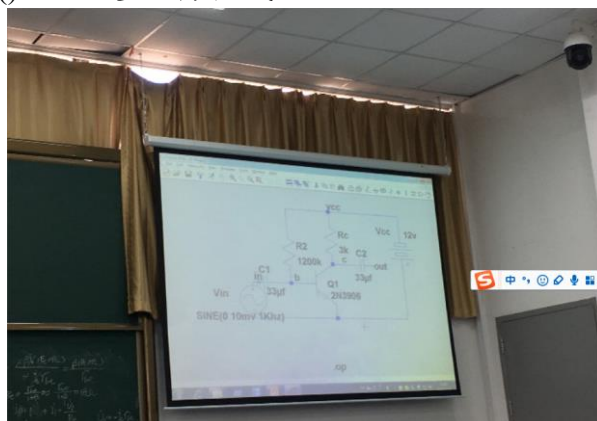


Fig 6 轮廓检测多边形表示示意图

4 实验内容

4.1 色度空间转换

1. 利用 `Imread` 函数或者从 ROS 话题中获取目标图片
2. 进行滤波去噪
3. 遍历原图的 RGB 三通道像素，并将其转换至特定色度空间
4. `Imshow` 显示结果图



Fig 7 HSV 空间转换结果图

4.2 颜色分割

1. 利用 `createTrackbar` 函数建立滑动条，对颜色空间转换后的各通道进行阈值分割
2. 根据阈值分割的结果，判断各种颜色的对应阈值
3. 针对不同颜色，分别对图像进行阈值分割
4. 对阈值分割结果进行数学统计，判断图像的颜色并输出分类结果

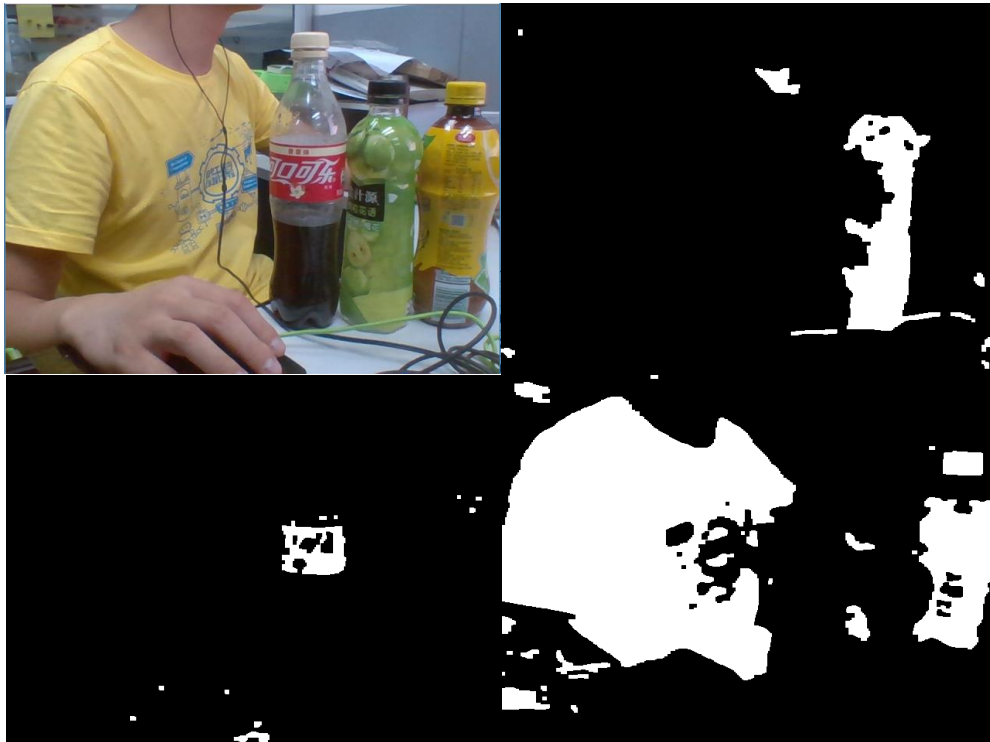


Fig 8 绿红黄三色的分割结果

4.3 目标颜色检测

1. 对图像进行预处理，消除噪声并获取二值化图

2. 对二值图进行轮廓检测;
3. 根据任务目标选择合适的多边形描述轮廓;
4. 获取多边形区域后, 从原图中截取该区域图像;
5. 对多边形区域的图像进行颜色分割, 对分割结果进行统计, 判断图像的颜色并输出分类结果;

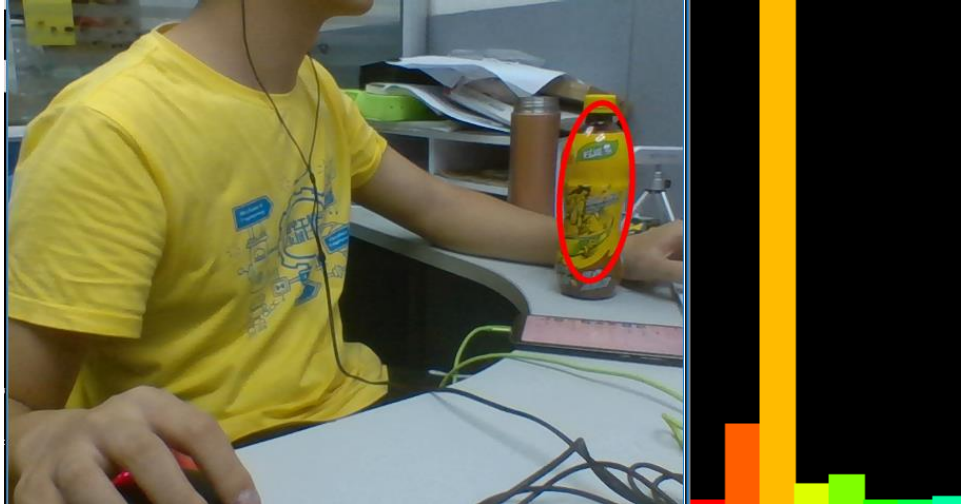


Fig 9 目标颜色检测结果

4.4 机器人控制

1. 机器人通过摄像头获取当前图像;
2. 检测指示牌, 并识别其颜色
3. 根据识别出的指示牌颜色, 对机器人发出控制指令, 其中:
 - 红色: 前进;
 - 绿色: 后退;
 - 蓝色: 左转;
 - 黄色: 右转。