

# Design of Positioning System for Indoor Mobile Multi-Robot Platform Based on Image Features

Guo-sheng Wang, Fan Xia, Feng Guo, Qiang Lv

Department of Control Engineering, Academy of Armored Forces Engineering, Beijing 100072 (gswang@126.com)

**Abstract**—A design problem of positioning system for indoor mobile multi-robot platform is investigated in this paper. Based on knowledge of image features, color features and shape features are used to distinguish indoor targets and obtain the positions of mobile multi-targets. The image processing library *OpenCV* and Microsoft Foundation Classes multithreading technology are used to extract image features of robot mark. By extracting image features of robot mark, a design method of positioning system for indoor multi-mobile robot platform is proposed. Finally, an experiment and its results of four indoor robots show that the proposed design method of positioning system for indoor multi-mobile robot platform is effective.

**Keywords**—multi-mobile robot, image features, *OpenCV*, multithreading technology, position

## 基于图像特征的室内多移动机器人定位系统设计

王国胜 夏凡 郭峰 吕强

装甲兵工程学院控制工程系, 北京, 中国

**摘 要** 本文考虑了室内多移动机器人系统平台的定位系统设计问题。基于图像特征知识, 利用其颜色和形状等特征区分室内移动目标并对移动多目标进行定位。本文利用图像处理库 *OpenCV* 以及微软基础类(Microsoft Foundation Classes, MFC)多线程技术提取机器人标记图像特征, 提出了一种室内多移动机器人定位系统的设计方法。最后, 4 个移动机器人平台实验及其实验结果表明本文所提室内多移动机器人定位系统的设计方法的有效性。

**关键词** 多移动机器人, 图像特征, *OpenCV*, 多线程技术, 定位

### 1. 引言

作为机器人研究的重要领域之一, 近年多移动机器人系统在许多工程领域得到了应用。在多移动机器人系统的实验研究中, 多机器人定位系统的研究是至关重要的。在室内环境下, 多机器人实验平台常用的定位技术主要有里程计、惯性导航、GPS 等。但是上述方法都有局限性, 如里程计技术由于其积分特性存在较大的误差累积、惯导技术会随着时间的增长存在漂移使得误差随时间增长、GPS 精度问题并且不宜室内使用等。近年随着图像技术的发展, 由于图像获取的信息丰富, 图像采样周期短且受磁场等干扰因素干扰小, 其在机器人领域的应用逐步加大。通过图像技术得到的信息更加准确快速且实时性良好, 能够有效克服积分等因素带来的误差。

本文利用图像处理技术对通过提取机器人的颜色、形状等图像特征信息完成机器人定位。图像处理使用图像处

理库 *OpenCV*, 它是 Intel 资助开发的开源计算机视觉库<sup>[1]</sup>。*OpenCV* 主要用于开发实时的图像处理程序。*OpenCV* 有强大的图像和矩阵运算能力, 为数字图像处理和计算机视觉应用提供了极大的方便<sup>[2]</sup>。由于机器人定位系统的实时性要求, 本文选择 *OpenCV* 完成图像处理。

本文以自制的四轮全向移动机器人<sup>[3]</sup>为基础, 在 Microsoft Visual 9.0 环境下, 利用图像处理库 *OpenCV* 及 MFC 多线程技术, 获取特定特征的标记信息, 经过换算得到全向移动机器人的定位信息, 从而利用全向移动机器人的运动模型、动力模型等<sup>[3,4]</sup>实现控制。

### 2. 颜色与形态特征提取

本文主要提取图像的颜色和形态两个主要特征。

#### 2.1 颜色空间变换

图像的捕获是利用摄像头获取实时视频信息，通过截取视频中的一帧得到信息图像，该图像是 RGB 颜色空间的图像，但是由于 RGB 模型对于光照强度变化敏感所以将其转化至 HSV 模型。

RGB 模型由红(R)、绿(G)、蓝(B)三原色组成，RGB 模型中产生色彩的原理是利用不同强度的三原色混合成所需颜色，每个原色的范围均是 0 至 255<sup>[5]</sup>。为方便起见可以将 RGB 模型看成一个三维空间并简化为单位立方体即以比例的形式表示原色的强度如图 1 所示。在单位立方体的主对角线上，R、G、B 强度相等，产生由暗到明的白色。

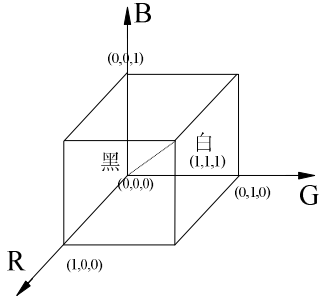


图 1 RGB 空间

在 HSV 模型中，H 代表色度，S 代表色度饱和，V 代表明亮值，这里 H 的范围是 0~180，S 和 V 的范围是 0~255<sup>[5]</sup>。HSV 模型中，亮度值被单独提取出来，可以有效的减弱亮度变化对图像颜色特征提取的影响，因此选择 HSV 模型为图像模型。HSV 模型可用图 2(1)的倒置圆锥体表示，其中 H 为围绕长轴的角度，S 为距离长轴的距离，V 为长轴<sup>[5]</sup>。

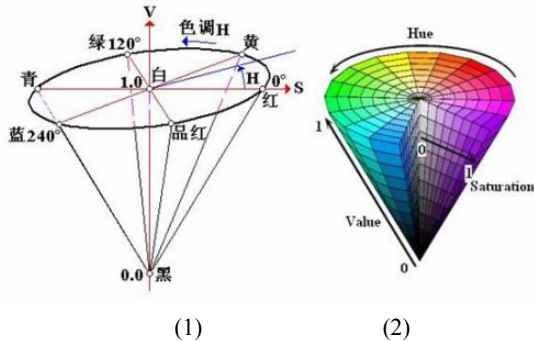


图 2 HSV 模型

2.2 阈值法图像分割

在得到 HSV 彩色图后，对其分割，便于轮廓提取。图像分割的目的是通过分割方法将图像中感兴趣的目标分离出来，便于后续图像处理。本文采用阈值法将 HSV 颜色空间的图像处理为灰度图。

阈值法，最重要的是判断 H、S、V 的阈值范围。本文采用红、绿、蓝、黄为标记色，查阅资料得到各色的阈值，顺序读取图像中各个像素的 HSV 值与阈值比较，符合阈值

条件的像素设定为 255 即白色，不符合的设为 0 即黑色。

2.3 数学形态学去噪

利用阈值法分割法得到的灰度图将要检测的目标分割了出来，但是存在一些不必要的像素点，对于这些像素点可以用数学形态学消除。形态学的基本方法有腐蚀、膨胀、开运算和闭运算<sup>[6]</sup>。腐蚀的目的是收缩细化图像，用一个结构元素 B 与图像 A 中的每一个像素做比较，比较元素的大小，若小于结构元素就消除掉以达到缩小目标消除噪声的目的<sup>[7]</sup>。腐蚀图如图 3 所示。

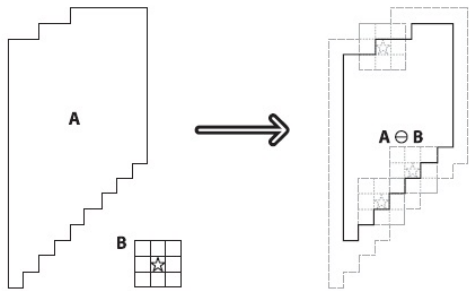


图 3 腐蚀图

膨胀的目的是加长加粗图像，与腐蚀一样利用一个结构元素 B 与每一个像素比较。膨胀的作用与腐蚀相反，将像素周围于其接触的并且小于结构元素的像素点合并到目标中，使得目标增大，填补目标中细小的空洞<sup>[7]</sup>。膨胀图如图 4 所示。

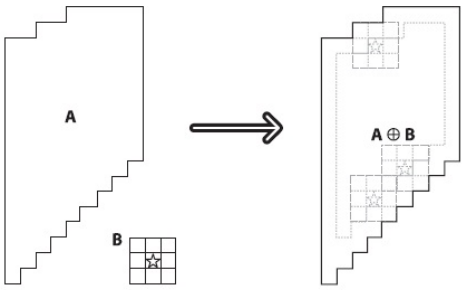


图 4 膨胀图

开闭运算是将腐蚀和膨胀结合使用的方法<sup>[7]</sup>。开运算将图像先腐蚀后膨胀，闭运算则是将图像先膨胀后腐蚀<sup>[7]</sup>。开运算与闭运算的腐蚀膨胀顺序相反得到的结果也大不相同，开运算可以达到消除噪声的目的，并平滑目标的边界，闭运算可以弥补目标中的空洞，同时平滑目标的边界。

2.4 边缘检测

在去除噪音后，在图像中边缘检测提取目标的形状特征，通过图像中目标的轮廓，可以简化图像分析的难度，便于图像信息的提取。

Canny 算子是应用较为广泛的边缘检测算子之一，是一种最优算法。首先对 x 和 y 方向求一阶导数，然后组合为 4 个方向的导数，在每个方向导数中找到局部最大的

点组成候选点<sup>[7]</sup>。利用双阈值法检测在候选点中选择边缘点。

Canny 算子的步骤为：

Step 1 Canny 算子首先高斯滤波平滑图像，用二维高斯函数的导数对图像进行平滑<sup>[8]</sup>。

Step 2 在高斯平滑后计算梯度幅值和方向。

Step 3 在计算梯度幅值和方向后，利用非极大值抑制法得到候选边界点。

Canny 在  $(i, j)$  处的  $3 \times 3$  邻域内选择与其  $\theta(i, j)$  相邻的两个点并比较梯度幅值的大小，如其幅值在两者中最大，则为候选边缘点<sup>[8]</sup>。

Step 4 利用双阈值法在候选边缘点中判断出边缘点。利用上限阈值和下限阈值，当像素的幅值大于阈值时，为边缘点，小于则抛弃，当介于上下限之间时如果边缘点连接时为边缘点<sup>[7]</sup>。

### 3. 多机器人定位

#### 3.1 MFC 的多线程支持

多线程技术是一种在硬件或软件上并发执行多个线程的技术。一个程序可以被看成一个进程，线程是进程中的可以独立调度和执行的代码单元，可以理解为代码的上下文。一个进程有多个线程，在一个进程中可以同时执行多个线程完成不同的任务，称为多线程。

MFC是Microsoft Visual中的一种应用框架。MFC拥有强大的界面开发功能，同时提供了多线程编程的支持。在MFC中线程可分为工作者线程和用户界面线程<sup>[9]</sup>。MFC中使用函数AfxBeginThread()创建线程，设置参数并在进程创建后即可运行。通过创建多个线程并发的运行，可得到多机器人定位系统。

#### 3.2 多机器人定位的实现

多移动机器人定位系统的设计流程可用如下图：

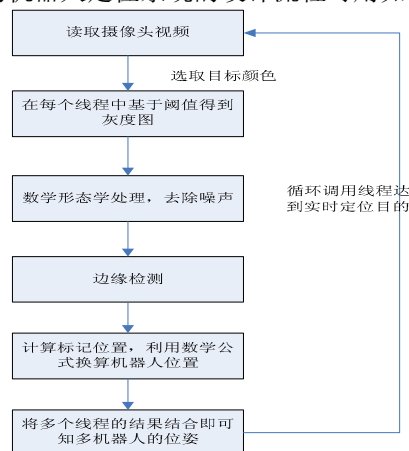


图 5 定位系统设计流程

根据所要检测的机器人数量设计多个线程，每个线程

分别完成对一个机器人标记的检测和定位。

多机器人定位系统的实现步骤：

Step 1 获取摄像头视频图像

利用cvCaptureFromCAM(int index)函数获取摄像头信息index为摄像头的索引根据实际设定。获取的仅为视频文件并不能直接获取视频中的一帧，OpenCV提供了cvQueryFrame()可以得到视频中的一帧。代码为：

```
IplImage* image;
CvCapture* capture;//视频获取结构
capture=cvCaptureFromCAM(1);//index根据实际定
image=cvQueryFrame(capture);//image即为得到的图像
//获取的图像为RGB模型，使用cvCvtColor()函数转换至
//HSV模型。实验证明HSV模型相对于RGB模型效果更好，
//对光照变化更不敏感。代码为：
cvCvtColor(image,image,CV_RGB2HSV);
//CV_RGB2HSV表示由RGB到HSV，imgae即为输入图像
//也为输出图像
```

Step 2 分割图像

在获取HSV彩色图像后，利用OpenCV中CV\_IMAGE\_ELEM()函数获取HSV图像每个像素的H、S、V分量的值，并设定阈值比较大小。CV\_IMAGE\_ELEM()函数声明为：CV\_IMAGE\_ELEM( image, elemtype, row, col)，其中若像素为(x,y),则row为像素的行坐标x，col的值为y\*image->nChannels+h，其中image->nChannels为图像的通道数，h取值为0,1,2。0代表H值，1代表S值，2代表V值。

阈值的设定在一定程度上决定了程序的运行速度，因为阈值分割决定了最后边缘处理得到的画面的效果，好的效果目标位置的计算简单，坏的效果则会使得定位不准并且使得计算变大，在实际中会使系统崩溃。

也可将HSV图像分解为3个单通道图像后，分别作阈值比较，但是为了运行速度考虑，使用提取H、S、V值的方法更快更高效。

Step 3 去除噪音

阈值分割得到灰度图中存在噪音。OpenCV提供了数学形态学处理函数cvErode(), cvDilate()。cvErode()为腐蚀函数，cvDilate()为膨胀函数。两种函数联合使用组成开运算或闭运算。cvErode(), cvDilate()的函数声明为：cvErode(const CvArr\* src, CvArr\* dst, IplConvKernel\* element=NULL, int iterations=1), cvDilate(const CvArr\* src, CvArr\* dst, IplConvKernel\* element=NULL, int iterations=1),其中src为输入图像，dst为输出图像，element为结构元素，根据实际确定。

Step 4 边缘检测

去噪后得到较为光滑的图像，检测图像边缘得到目标

边缘，OpenCV中提供了Canny算法的实现<sup>[10]</sup>。边缘检测的实现：利用cvCanny()检测去噪后所得图中目标的边缘，而后使用cvFindContours()函数提取边缘像素序列，利用cvFitEllipse2()实现椭圆拟合边缘得到近似圆形的边缘图像。cvCanny的函数声明为：cvCanny( const CvArr\* image, CvArr\* edges, double threshold1,double threshold2, int aperture\_size=3 )，其中image为原图，edges为边界，threshold1和threshold2分别为上下限。

Step 5 形态学特征提取

得到边缘后，利用形态特征提取，根据形态特性得到所需信息。

本文所用坐标系及方向角定义如图6。根据形态特征提取，得到的两个图形标记的中心，利用几何学得到机器人的中心坐标及机器人的方向。机器人的中心即为两圆圆心连线的中点，角度利用反三角函数可得角度。

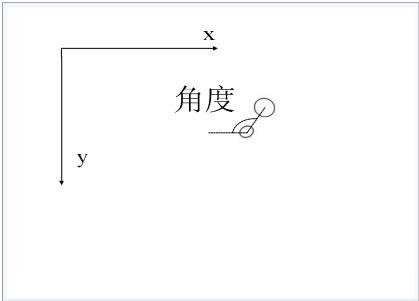


图6 坐标系

4 实验测试

本文以 4 个四轮全向机器人为基础，分别在全向机器人上加上一大一小两个圆形标记，颜色分别为红、蓝、绿、黄。利用置于棚顶的摄像头获取视频信息。摄像头的分辨率为 640\*480，距离地面 2.62m。

4.1 摄像头图像获取结果

本文 cvCaptureFromCAM(int index)函数的摄像头的索引 index 为 1。使用 cvQueryFrame()可以得到视频中的一帧后得到的图像如图 7。使用 cvCvtColor()函数将获取的 RGB 模型图像转换至 HSV 模型。

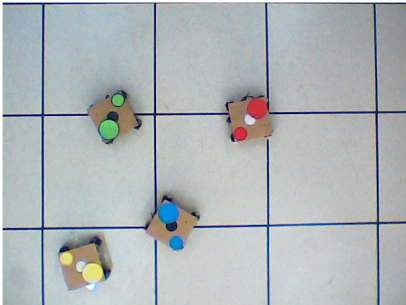


图 7 摄像头图像

4.2 MFC 用户界面框架

在 MFC 框架中设计用户界面，设定按钮及显示功能。界面如图 8 所示。



图 8 用户界面

其中在启动按钮函数中加入线程的初始化及运行程序，停止按钮停止程序。利用 AfxBeginThread()创建并运行线程。

4.3 分割目标结果

根据实际测定，本文实验条件下的各色 HSV 阈值为表 1。

表 1 各色 HSV 阈值

	红	黄	蓝	绿
Hmin	110	75	11	40
Hmax	130	99	25	70
Smin	130	100	220	100
Smax	255	255	255	150
Vmin	150	200	200	165
Vmax	255	255	255	255

利用阈值分割法分割图 7 后所得灰度图如图 9。

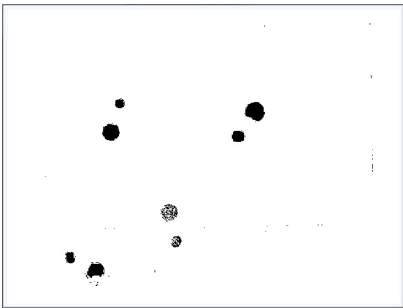


图 9 阈值分割图

4.4 形态学去噪结果

根据情况使用开运算或闭运算，本文对图 9 使用开运算的方法，消除图像中的噪声，减小边缘检测时的误差。开运算后得到的图像如图 10。比较图 9 和图 10，发现图 10 中细小噪声明显减少但是蓝和黄色检测有较多空洞，对其闭运算消除空洞后得图 11。

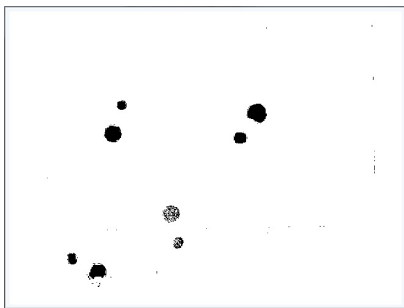


图 10 开运算处理图



图 11 闭运算处理图

#### 4.5 边缘检测结果

对图 11 利用 Canny 函数边缘检测，得到图像如图 12。从图 12 可以看到的得出了较准确的圆形边缘。

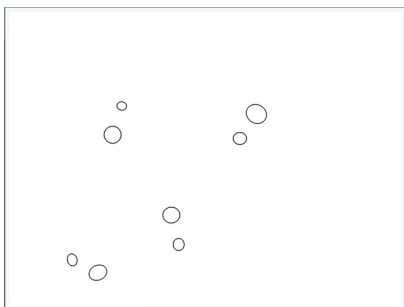


图 12 边缘检测图

在用户图形界面输出如图 13。图 13 中的数值为十六进制数。



图 13 用户界面显示

#### 4.6 多机器人运动跟踪定位

对运动中的机器人跟踪定位，多机器人运动轨迹如图

14。

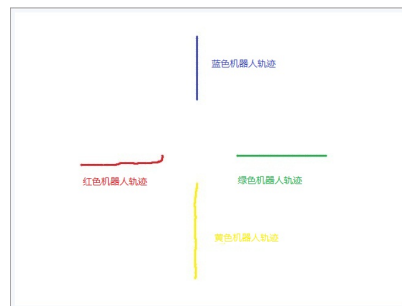


图 14 多机器人运动轨迹

不同的颜色分别代表其相对颜色的目标中心运动轨迹。截取三组值比较其与实际的误差。

截取图如图 15，图 16，图 17，实验结果及测量结果如表 2，表 3，表 4。

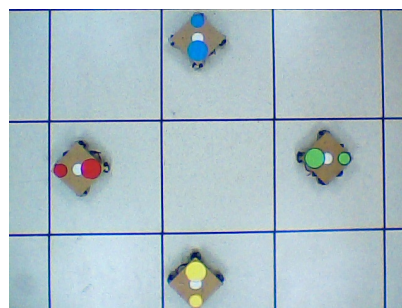


图 15 第一组截图

表 2 第一组截图对应实验结果

	实验结果		实际测量		定位误差	
	(x,y)/cm	角度	(x,y)/cm	角度	距离/m	角度
绿	(156,78)	0°	(150,71)	3°	0.09	3°
黄	(98,125)	90°	(90,130)	90°	0.09	0°
蓝	(99,13)	270°	(90,13)	270°	0.09	0°
红	(36,83)	177°	(32,76)	178°	0.08	1°

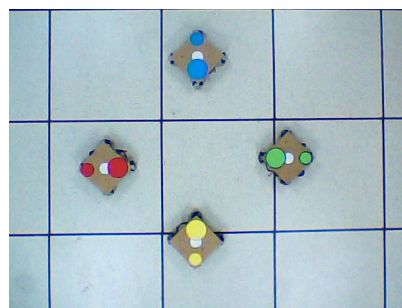


图 16 第二组截图



表 3 第二组截图对应实验结果

	实验结果		实际测量		定位误差	
	(x,y)/cm	角度	(x,y)/cm	角度	距离/m	角度
绿	(151,78)	0°	(147,72)	3°	0.07	3°
黄	(98,120)	90°	(92,113)	90°	0.09	0°
蓝	(99,24)	270°	(90,24)	270°	0.09	0°
红	(50,83)	177°	(48,75)	178°	0.08	1°

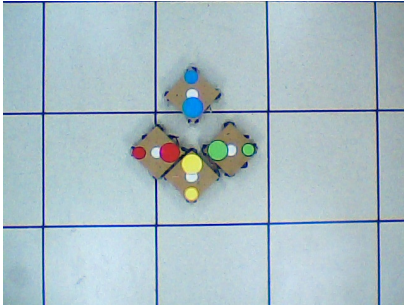


图 17 第三组截图

表 4 第三组截图对应实验结果

	实验结果		实际测量		定位误差	
	(x,y)/cm	角度	(x,y)/cm	角度	距离/m	角度
绿	(113,78)	0°	(108,70)	-4°	0.09	4°
黄	(94,86)	90°	(90,84)	90°	0.04	0°
蓝	(94,47)	261°	(90,45)	265°	0.04	4°
红	(89,78)	180°	(82,72)	180°	0.09	0°

由三组实验结果可以看出，定位距离误差最多不超过 9cm，角度误差不超过 5°，可以认为在室内环境下定位较准确。

## 5 结论

本文利用多线程方法在 MFC 和 OpenCV 的环境下实现了室内多移动机器人定位，设计了多移动机器人的定位系统。通过四个移动机器人平台实验验证了本文所提定位方法，得到了良好的定位精度。本文使用的形状特征比较单一，均是使用的圆形标记，可以改变标记的形状来提高目标定位的准确率。

## 参考文献

- [1] 刘慧英, 王小波, “基于 OpenCV 的车辆轮廓检测,” 科学技术与工程, vol. 10, no. 12, pp. 2987-2991, 2010.
- [2] 刘江, 张岑, “基于 OpenCV 的圆形标记点的提取,” 组合机床与自动化加工技术, pp. 27-29, 2013.
- [3] 王国胜, 吕红涛, 吕强, “四轮全向移动机器人的双闭环速度

控制系统设计,” 装甲兵工程学院学报, vol. 27, no. 1, pp. 59-64, 2013.

- [4] 吕红涛, 王国胜, 吕强, 刘峰, “基于 TMS302F28069 的自平衡机器人控制系统设计,” 单片机与嵌入式系统应用, vol. 12, no. 9, pp. 42-45, 2012.
- [5] S. P. Zhang and C. T. Zhang, “Image analysis for wave swash using color feature extraction,” in *2nd International Congress of Image and Signal Processing*, Tianjin, China, Oct 2009, pp. 1-4.
- [6] 赵红雨, 吴乐华, 史燕军, 王志中, “基于 HSV 颜色空间的运动目标检测方法,” 现代电子技术, vol. 36, no. 12, pp. 45-48, 2013.
- [7] G. Bradski and A. Kaehler, *Learning OpenCV*. Sebastopol, CA: O'Reilly Media, Inc., 2008.
- [8] 李俊山, 马颖, 赵方舟, 郭莉莎, “改进的 Canny 图像边缘检测算法,” 光子学报, vol. 40, no. S1, pp. 50-54, 2011.
- [9] 王伟, “基于 MFC 的多线程与串口通信,” 电子世界, pp. 106, 2013.
- [10] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, “A brief introducton to OpenCV,” in *MIPRO 2012*, Opatija, Croatia, May 2012, pp. 1725-1730.