

文章编号: 1002-0446(2003)01-0010-06

移动机器人多目标彩色视觉跟踪系统*

汤思维¹ 陈卫东¹ 曹其新²

(1. 上海交通大学自动化研究所 上海 200030; 2. 上海交通大学机器人研究所 上海 200030)

摘要: 机器人视觉系统利用颜色、形状等信息来识别环境目标,但是难点在于识别的鲁棒性和实时性的保证。本文采用全自主移动机器人平台,提出了一种硬件成本低廉的、基于颜色学习的实时多目标视觉跟踪系统,并提出了一种新颖的目标颜色学习和跟踪算法。该视觉系统已成功集成在自主移动机器人系统中,非结构环境下的动态目标跟踪实验表明了系统的实时性和鲁棒性。

关键词: 移动机器人; 颜色学习; 颜色向量集; 色块(blob); 游程长度编码

中图分类号: TP24 **文献标识码:** B

MULTI-OBJECT COLOR TRACKING VISION SYSTEM FOR AUTONOMOUS ROBOT

TANG Siwei¹ CHEN Weidong¹ CAO Qixin²

(1. Institute of Automation, Shanghai Jiaotong University 200030;

2. Research Institute of Robotic, Shanghai Jiaotong University 200030)

Abstract: Color provides powerful information for object recognition. But it's difficult for robot vision system to recognize targets quickly and robustly using such information. To solve this problem, this paper describes a color training robot vision system capable of real-time tracking multi-objects quickly by very low-cost commodity hardware and presents a novel but easy color training and object tracking algorithm. The vision system has been successfully integrated with our autonomous robot system and shown its quickness and robustness in dynamic object tracking experiments in non-structured environments.

Keywords: autonomous robot, color training, color vector set, blob, run length code

1 引言(Introduction)

颜色是物体表面的固有特征,在目标识别和图像分割中有着无法替代的作用,但是难点在于识别的鲁棒性和实时性的保证。常用的彩色图像目标识别方法大多基于固定或线性颜色阈值^[1,2]。这类方法预先在颜色空间中定义了一些颜色多面体作为阈值,颜色值位于这些阈值多面体内部的像素就是目标像素,非常简单。但不足之处是很难找到完全或基本匹配目标颜色的多面体,极易造成误分类,鲁棒性较差。这种方法在机器人足球比赛中使用较广泛^[2],在实际比赛中,一旦光照条件变化比较大,很多队伍便会感到不适应而导致比赛失利。

针对固定或线性阈值的不足,人们提出了基于色彩学习的彩色图像分割方法^[3]。这类方法的优点是

通过事先的色彩学习,可以较好地适应光线条件的变化。一种方法是将需要学习的目标颜色值进行归一化处理,并求取平均值和标准差,将其作为目标颜色特征加以学习^[4]。该方法对于颜色均匀的目标分割效果较好,但是当处于复杂环境中时,目标表面颜色通常呈现出复杂分布,识别效果很难保证。而且随着环境的变化,表面颜色分布也会变化,因此识别的鲁棒性也是一个问题。

本文提出了一种基于颜色学习的多目标视觉跟踪系统及其算法实现。通过预先的颜色学习,可以对多种目标颜色同时进行跟踪和信息提取,而且学习过程非常简单,只需拉动鼠标即可。实验证明该系统实时性好,快速准确,并在全自主机器人足球比赛和机器人定位等实验中得到了很好的应用。

* 基金项目: 国家 863 计划(2001AA422140, 2001AA422200), 国家自然科学基金(60105005)资助项目。

收稿日期: 2002-08-07

2 系统结构(System setup)

2.1 视觉系统硬件结构

该视觉系统由一个标准的工业彩色 CCD 摄像机和图像采集卡所组成. 摄像机的水平清晰度是 480 电视线, 标称视角是 110 度. 考虑到机器人的成本, 图像采集卡采用的是市面上非常普遍的“天敏开发 2000”. 该采集卡基于 BT878 芯片, 采集速度为每秒 30 帧.

2.2 视觉系统软件功能

该软件单独作为一个进程工作于客户机-服务器模式的客户端. 软件中定义了 8 个通道, 每个通道对

应某一种颜色, 因此可以对 8 种目标颜色进行分通道在线学习并分别记忆. 目前每个通道可以同时跟踪 3 个目标, 各通道互不干扰. 经过学习后, 被跟踪的目标以蓝色色块(blob)的形式表示, 色块外部则以紫色方框的形式示出. 所有紫色方框在图像中的位置和面积以及方框中被跟踪色块的重心坐标和像素面积等信息被实时地(即每 33ms)刷新, 并等待服务器端的连接以便发送数据. 服务器端的机器人决策程序利用这些信息, 可以在一定程度上由图像中被跟踪目标的二维投影得出三维信息. 视觉系统模型如图 1 所示.

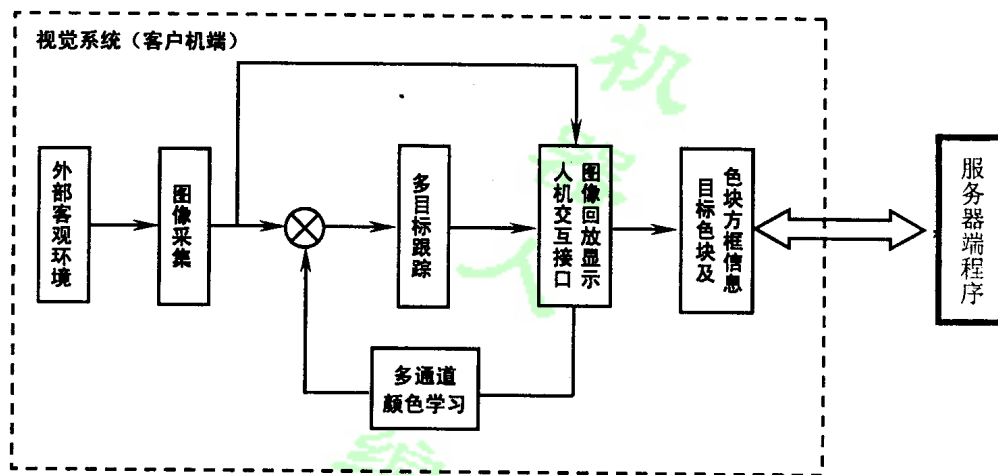


图 1 视觉系统模型

Fig. 1 Vision system model

3 彩色目标的学习和跟踪算法(Training and tracking algorithm)

3.1 色彩空间的选择

在图像处理中广泛使用的色彩空间主要有 HSI、YUV 和 RGB 等等. 其中 HSI 和 YUV 模型反映了人眼观察彩色的视觉规律, 从而在实际中应用较多. 但是由于图像采集卡的输出模式通常为 RGB, 而从 RGB 到 HSI 或 YUV 的转换要涉及到大量的浮点运算, 势必对系统的实时性产生很大影响, 因此采用 RGB 颜色模型可以保证系统的实时性, 而且使用方便. 但是, 它的缺点也同样明显, 主要在于 RGB 三个分量之间的高相关性^[5]以及空间上的高分散性, 这使得在不同的光照条件下, 同一颜色物体表面的 RGB 值会很分散地在整个 RGB 空间中变化, 为识别带来很大的困难.

虽然如此, 我们仍然采用了 RGB 作为目标识别

的颜色空间. 这主要是为了降低运算量, 提高系统实时性. 因此, 为了能在一定程度上弥补 RGB 模型的缺陷, 在确定目标的颜色特征时, 我们并没有采用固定阈值的方法, 而是采用了人为颜色学习从而构造颜色向量集的方法. 在不同的光照条件下, 还可以对目标物体进行多次学习, 最后构造出目标表面颜色的 RGB 向量集. 将图像中所有像素的 RGB 向量同该向量集相匹配, 即可找出所有的目标像素. 这样, 就有效避免了 RGB 模型的高分散性和高相关性所带来的阈值划分问题, 既保证了视觉系统的实时性, 又增加了在 RGB 模型中进行颜色目标识别的鲁棒性.

3.2 颜色学习算法

RGB 是一个非常庞大的三维空间, RGB 图像的每个像素在内存中占有 24 位, R、G、B 各占 8 位, 因此, 对单独一个通道来说, 需要保留将近一千七百万种颜色索引, 如果要 8 个通道同时处理的话, 还需要

乘以 8. 所以如果不经过一定的变换, 就需要占据极大的存储空间和计算时间. 因此, 我们利用了一个很简单的投影变换, 将图像中每个像素在 RGB 空间中所对应的颜色向量分别投影到红/蓝、红/绿和蓝/绿三个平面上, 从而降低系统维数和数据存储量. 我们首先定义三个全部为零值的二维数组 $RG[256][256]$ 、 $RB[256][256]$ 和 $GB[256][256]$. 进行颜色学习的时候, 用鼠标在图像中需要学习的目标颜色上拉方框, 系统就会将方框内所有像素对应的颜色向量进行投影, 并以投影坐标值作为索引分别将三个二维数组的相应元素置 1. 学习好后将最终的三个数组中置 1 的元素索引合成, 就构成了目标颜色的颜色向量集.

颜色向量集形成以后, 我们要对整个图像进行扫描以确定所有目标像素. 扫描的过程就是将每个像素颜色值经过投影后的坐标作为数组索引输入, 把对应的数组中的三个元素进行“按位与”操作, 即:

$$\text{interest_pixel} = RG[\text{greenindex}][\text{redindex}] \text{ BITWISE_AND}$$

$$RB[\text{blueindex}][\text{redindex}] \text{ BITWISE_AND}$$

$$GB[\text{blueindex}][\text{greenindex}]$$

从而确定该像素是否为目标像素.

为了能够实现 8 种颜色同时跟踪, 同时又能减小数据量和计算量, 我们将数组元素定义为字节型变量, 字节的每一位代表一个通道, 这样可以同时计算 8 个通道.

3.3 颜色区域分割和跟踪算法

在分通道找出所有目标像素之后, 要将它们在不同的通道按照四连通关系分别加以组合以形成区域(当然也可以采用八连通关系). 我们通过三个阶段进行融合计算.

第一阶段是对所有目标像素进行游程长度编码^[1,6]. 游程长度编码用图像像素值连续为 1 的个数来描述图像, 是在图像处理中较常使用的一种二值图像表示方法. 在该编码中经常运用两种方法, 一种是使用 1 的起始位置和 1 的游程长度, 另一种是仅仅使用游程长度. 举例来说, 假如有一幅用如下表格表示的简单二值图像, 就可以分别写出两种方法的游程长度编码, 非常简单. 在算法中, 我们采用了第一种表示方法, 并对该方法作了改动, 多记录了两项内容, 即通道标记和起始像素内存地址.

上面简单二值图像的第一种游程编码方法: 1 的游程是(2, 2)(6, 3)(4, 5)

0	1	1	0	0	1	1	1	0	0
0	0	0	1	1	1	1	1	1	0

第二种游程编码方法: 1 和 0 的游程长度是

0, 1, 2, 2, 3, 2

0, 3, 6, 1

在图像处理的实践中, 大多数区域融合是基于连通标记或是区域生长算法的^[1], 其中连通标记算法需要多次对图像中所有像素扫描, 另外在多目标跟踪时, 需要用大量的空间存储标记和标记后的像素, 导致计算量和数据存储量迅速增加. 而区域生长算法同样计算复杂费时, 很少用于对实时性要求高的场合^[7]. 因此游程长度编码使用后的优点显而易见, 区域融合计算量由像素级变为游程级, 而且只需要考虑图像纵向的邻接关系. 这样, 降低了处理的维数(二维变为一维), 有效地减少了运算量, 使系统实时性得到了保证.

第二阶段是从上至下纵向扫描游程, 判断处于同一通道的相邻两行的游程是否邻接, 以构成游程的有向图结构^[8], 每个游程就是图的一个顶点. 并根据顶点的地址形成有向邻接表.

第三阶段是再一次从上至下扫描游程, 按照邻接表使所有相连的顶点指向与它们相连的第一个顶点, 当指向操作全部完成后, 区域融合变告结束, 每个区域以它的第一个顶点的地址作为自己的区域标识. 整个融合过程见图 2.

在区域融合过程中同时进行了区域各项特征的计算. 区域特征主要包括色块的面积(以像素数表示)、重心、外部方框在图像中的位置坐标和面积等等. 将区域中所有游程长度相加即可得出色块面积 A . 色块重心坐标 (\bar{x}, \bar{y}) 可以根据下面两式计算:

$$\bar{x} = \frac{\sum_{i=1}^n \frac{l}{2}(2 \times a + l - 1)}{A}$$

和

$$\bar{y} = \frac{\sum_{i=1}^n h \times l}{A} \quad (1)$$

式(1)中 l 为每个游程的长度, a 为游程起始像素横坐标, h 为每个游程的纵坐标, n 是区域中游程的个数. 而外部方框在图像中的位置坐标则可以由色块中横坐标最大、最小和纵坐标最大、最小的四个像素得到. 之后, 将每个通道的所有区域按照面积从大到小进行排列, 取出面积最大的 3 个并以紫色方框显示. 各通道分别计算和显示, 互不干扰.

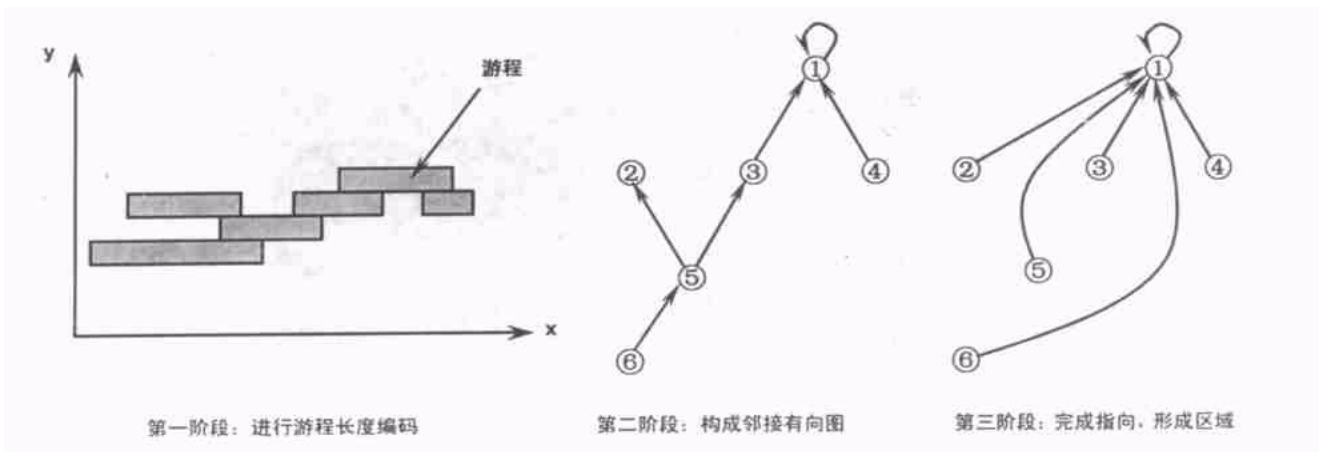


图 2 使用游程长度编码进行区域融合的三个阶段

Fig. 2 Three stages of region merging using run length encode

4 实验结果及系统应用(Experiment results and applications)

该视觉系统使用的机器人平台是自主开发的“交龙”中型全自主移动机器人(见图 3).

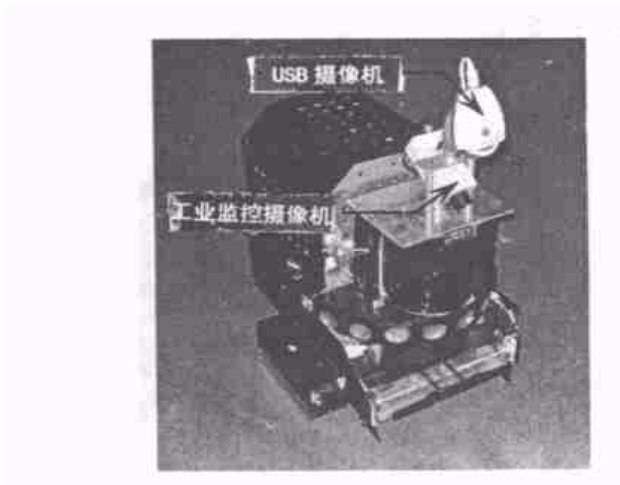


图 3 “交龙”全自主机器人

Fig. 3 “Jiaolong” autonomous

我们另外安装了一台 USB 接口彩色摄像机, 专门用来观察机器人前方近身处. 而工业摄像机则用来识别远处的目标(两台摄像机均固定). 机器人中央处理单元基于 Intel Pentium III 800 处理器, 操作系统为 Windows 2000. 本文所介绍的算法在该平台上对典型的 320× 240 像素图像进行处理时, CPU 利用率仅为 25% 左右, 使视觉系统能以每秒 30 帧的全速运行. 这完全能够满足系统实时性的需要.

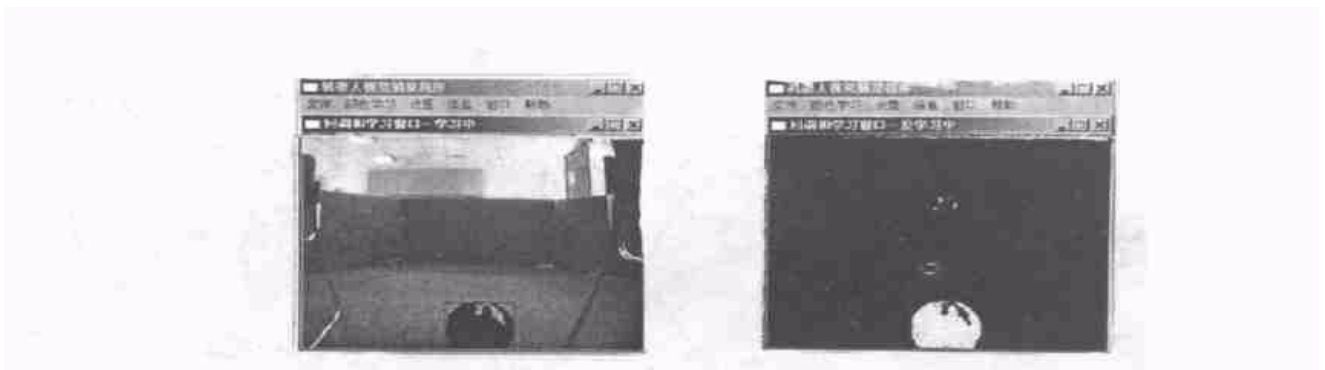
图 4 显示了在一般复杂环境下的学习和跟踪结果, 实验中人为地将环境作了一定的改动. 其中 (a) 为环境改动前的原图, (b) 为经过学习后的识别结果, (c) 为 (b) 图的二值化显示, (d) 图是未经重新学习但人为加入了围栏, 使得光照条件有了较大的变化后的结果, (e) 为 (d) 的二值化图像. 由于选择的场景中很多物体与球的颜色相接近, 而且环境的变化导致图像背景亮度条件变化较大, 因此增加了识别和跟踪的难度. 但是由于学习的灵活性及系统的鲁棒性, 系统仍然能够较好地识别和跟踪球.



(a) (b) (c)

图 4 一般环境和环境变化下的学习和跟踪结果

Fig. 4 Camera view of training and tracking results in complex environments



(d) (e)

图 4 一般环境和环境变化下的学习和跟踪结果

Fig. 4 Camera view of training and tracking results in complex environments

该视觉系统的一个成功应用是在全自主机器人足球赛中. 机器人足球赛中对场地和场地中所有物体的颜色均有规定. 在开始比赛前几分钟, 在场地现有光照条件下将所需的场上目标颜色分通道学习并保存. 比赛开始后, 视觉系统向机器人决策程序实时传送目标信息. 使用该系统, 由“交龙”机器人组成的球队获得了 2002 年中国机器人竞赛中型全自主机

器人组 1 对 1 和 2 对 2 项目比赛两项冠军. 图 5 说明了系统在场分通道工作的情况. 为简单起见, 图中显示了两个通道, 分别用来跟踪球和对方球门. (a) 为原图; (b) 为用其中一个通道学习球的情况; (c) 是 (b) 的二值化图像; (d) 是使用另一个通道学习对方球门的情况; (e) 是 (d) 图的二值化显示. 可以看出, 各通道并不干涉.



(a) (b) (c)



(d) (e)

图 5 在足球赛中多通道学习和跟踪情况

Fig. 5 Camera view of multi-objects training and tracking results in two different channels