

基于视觉的机器人导航算法研究^{*}

谭晓军, 郭志豪

(中山大学电子与通信工程系, 广东 广州 510275)

摘 要: 针对近年来国内外组织的大学生机器人比赛的某类题目, 提出了两种导航线识别的算法。经实践检验, 此两种算法都具有较高的可靠性。与传统的图像处理方法相比, 具有更强的针对性以及更好的实时性。

关键词: 计算机视觉; 机器人导航; 自动导航车

中图分类号: TP242 6 **文献标识码:** A **文章编号:** 0529-6579 (2004) S1-0127-05

近年国内外组织的大学生机器人比赛题目中有一类是提供了导航白线的, 自主机器人可以寻白线行走定位。针对此类题目, 一般可以选择普通色感器来寻线, 也可以采用计算机视觉技术来进行寻线定位。采用计算机视觉技术主要有以下优点:

(1) 结构简单: 只需要一个摄像头和一部笔记本电脑即可, 笔记本电脑可以通过串口或者并口对电机等执行机构进行控制, 在机器人不使用的时候, 笔记本电脑可以用作其他的科研工作, 这样可以提高资源的重复利用率;

(2) 有较好的鲁棒性: 很多采用色感器的机器人, 在运作之前都要精心放置色感器的位置, 并需进行仔细的自检过程, 因为色感器的数目一般为 10 个以上, 若比赛中某一色感器失效, 则有可能引起致命的错误;

(3) 有较好的可扩展性: 近年的机器人比赛题目越来越要求机器人具有一定的智能化, 而不是机械的执行完一个流程就可以完成比赛了。采用计算机视觉技术, 可以方便地与人工智能技术结合在一起, 让机器人真正具有“智慧”。

关于如何在机器人上构造计算机视觉的软硬件

系统, 笔者将在另一文章中讲述。本文的主要内容是讨论如何利用摄像机摄取到的图像进行导航。文章中所谈到的算法先后被用于若干全国大学生机器人比赛, 具有较高的效率及稳定性。

1 导航图像以及从中提取的信息

图 1 所示为一种典型的摄像机摆放的位置, 箭头方向为前进方向。图 2 所示为摄像机在实际场地所拍摄到的较典型的导航图像。我们的工作就是要结合机器人行走机构所采用的导航模型提取相应的方位信息。

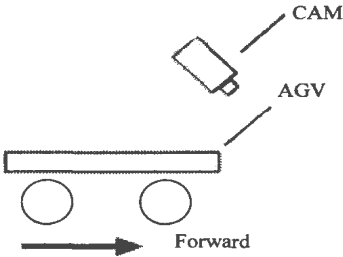


图 1 摄像机在 AGV 上的位置

Fig 1 A typical placement of the camera

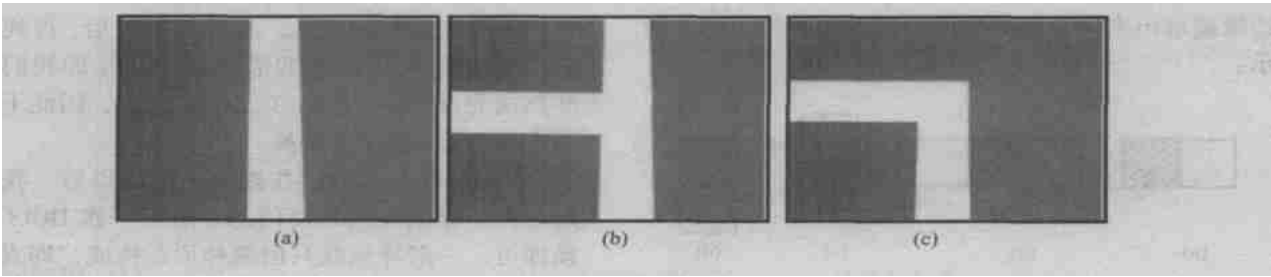


图 2 几种典型的导航图像

Fig 2 Some typical images taken on-line

^{*} 收稿日期: 2003-11-07

基金项目: 中山大学科技基金资助项目 (2001020504209213)

作者简介: 谭晓军 (1977 年生), 男, 博士研究生; E-mail: zsutxj@163.com

较为典型的一种导航模型就是所谓的 d, θ 模型, 其中 d 和 θ 分别反映了底盘和白色导引线的距离误差和角度误差 (图 3)。

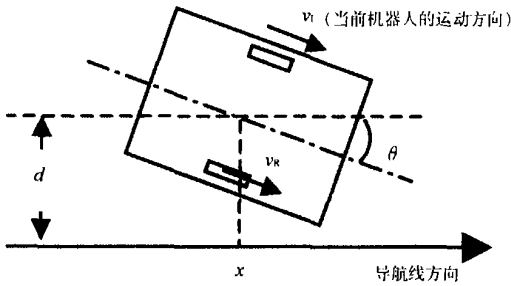


图 3 机器人导航模型

Fig 3 Navigation Model of the Robot

2 两种特征提取算法

2.1 导航图像的特点分析

分析导航图像, 可以发现如下特点:

(1) 图像色彩比较单一: 采集到的图像一般只包含了地板胶的颜色与导航线的颜色。

(2) 图像的边缘信息明显: 导航线与地板胶有着明显的颜色反差, 二者相接处产生了明显的边缘信息, 可以利用此边缘信息提取方位信息。

(3) 导航线具有连续性: 如下图所示, 在机器人正常行走过程中, 若图象中的 A 区域出现有导航线, 那么图像的 B、C、D 区域中至少有一个区域会出现导航线。

也就是根据以上分析, 提出了以下两种的导航线识别的方法。

2.2 边缘提取结合 HOUGH 变换的方法

2.2.1 对图像进行边缘提取 如前所述, 图像的边缘信息非常明显, 因此可以在图像中方便的找出导航线的边缘。一般来说就图像的横坐标方向和纵坐标方向分别做一次梯度扫描, 就可以把导航线的边缘提取出来。常用的梯度扫描的模板如下图 4 所示:

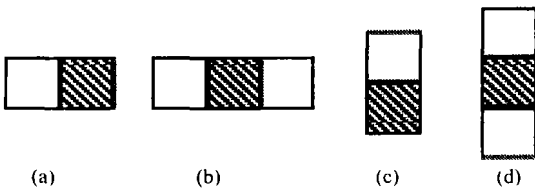


图 4 常用的梯度扫描模板

Fig 4 Templates used in scanning gradient

图中带斜纹的格子为待处理的像素点, 而白色的为其相邻的点。其中模板 (a)、(b) 主要用于横坐标方向, 而其中模板 (c)、(d) 主要用于纵坐标

方向。由于两个方向在原理上没有本质差异, 因此我们只需比较 (a)、(b) 两个模板的差别。

假设我们要处理的象素位于图像的第 i 列, 第 j 行, 即图像坐标为 (i, j) , 其灰度值为 $g(i, j)$, 使用 (a) 模板的逻辑是:

如果 $[g(i, j) - g(i - 1, j)] > \text{阈值}$, 那么判决数组的值 $p(i, j) = 1$, 否则, $p(i, j) = 0$ 。

而使用 (b) 模板的判断逻辑是:

如果 $[g(i, j) - g(i - 1, j)] > \text{阈值}$, 或者 $[g(i, j) - g(i + 1, j)] > \text{阈值}$, 那么判决数组的值 $p(i, j) = 1$, 否则, $p(i, j) = 0$ 。

应用以上两个模板对图 2 (a) 的处理结果如图 5 所示。

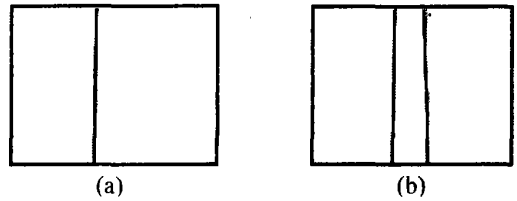


图 5 两种梯度模板的处理结果

Fig 5 Results obtained by different templates

从以上结果可以看出, 使用 2 个模板的结果都是可以获得单象素的边界, 而图 4 (a) 模板的好处是可以直接获得单边界, 原理简单, 便于从后续的 HOUGH 变换结果中提取直线信息; 而图 4 (b) 模板的好处是获得双边界, 虽然在 HOUGH 变换结果中提取直线信息时比较麻烦, 但可以根据两条直线的象素距离判断是否符合导航线的真实宽度, 提高算法的抗噪性, 减小误判的几率。因此, 实际使用时可以根据需要选择合适的模板。

值得注意的一点是, 经过扫描以后, 得到的一个二维函数 $p(i, j)$ 它的值域是 $\{0, 1\}$, 即我们在梯度扫描的过程中已经作了二值化处理, 因此下面可以直接进行 HOUGH 变换。

2.2.2 HOUGH 变换 在提取了边缘以后, 我们利用已知的导航线的几何信息, 进行一次 HOUGH 变换即可。一般导航线只由两种形态构成, 即直线和弧线。对于弧线的形态, 我们可以采用圆的参数方程进行 HOUGH 变换。因为目前大部分的导航线都是以直线或者相交的直线所给出的, 因此我们这里以直线为例, 讨论 HOUGH 变换的算法。

常用的直线的参数方程有以下两种, 一是斜率、截距方式 $y = ax + b$ (“斜截式”); 另一种是极

坐标方式 $x = \rho \cdot \cos \theta; y = \rho \cdot \sin \theta$ 。考虑到我们的直线有可能与横坐标轴平行, 则用斜率要取无限大, 用“斜截式”不方便, 因此我们选择用极坐标的方式。具体过程如下。

首先, 按照一般的习惯建立横、纵坐标(i 为横坐标, j 为纵坐标)。

需要注意的是图像的各个像素点都处于直角坐标系的第一象限中, 这在提取特征值的时候可以利用的一个特点

第二, θ 为极角, 因而其取值范围是 $[0^\circ, 180^\circ]$ 。计算所有 ρ 的最大值 $\rho_{\max}$, $\rho_{\max} = \sqrt{i_{\max}^2 + j_{\max}^2}$, 其中 $i_{\max}, j_{\max}$ 恰好就是图像的宽度和高度, 以 (ρ, θ) 为坐标建立一个新的平面, 从而把所有像素点映射到新的参数平面中去。

第三, 建立一个初始值为零的二维数组 $q(\rho, \theta)$, 用于统计, 对于每一个像素点 (i, j) , 变换到参数 (ρ, θ) 平面的一条曲线, 同时进行统计, 方法如下:

```
θ = 0;
循环开始:
如果 θ < 180°, 则继续, 否则, 跳出循环;
对于每一个的值, 计算出 ρ, ρ = i · cos θ + j · sin θ;
二维数组的相应位置加 1, 即 q(ρ, θ) = q(ρ, θ) + 1;
θ = θ + 1;
返回循环开始;
.....
```

循环结束后, 获得了一个统计量 $q(\rho, \theta)$, 检查所有 $q(\rho, \theta)$ 的值, 就可以提取有关的直线信息。

2.2.3 直线信息的提取 根据 HOUGH 变换的原理, 原理图像平面的直线实际就是 (ρ, θ) 平面多条曲线的交点, 那么 (ρ_0, θ_0) 若是这么一个“交点”, 那么其对应的统计值 $q(\rho_0, \theta_0)$ 就应该比较大, 因此我们设定一个阈值, 找出大于阈值的 $q(\rho, \theta)$ 及其对应的坐标 (ρ, θ) , 就是所求的直线的参数。这里

所提到的“阈值”, 可以根据平常试验的经验值找出, 也可以在每次统计的过程中动态地生成, 我们可以根据导航线的复杂性选择相应的合适的阈值产生的方法。

2.3 带滤波窗的区域扫描的方法

根据导航图像中, 导航线是连续的这一特点, 我们可以选择特定的区域进行扫描。扫描区域减小了, 则处理时间可以得到相应的提高。

如图6虚线部分所示, 在图像的上下左右靠近边界处建立扫描区域, 并通过边界扫描的方法在“扫描区域”内找出导航线的边界点。根据这些点确定两个信息: 导航线的形态、机器人航向与导航线的偏差。其中导航线的形态用于判断机器人是否需要转弯, 而航向的偏差反馈到机器人的运动控制模块, 对电机进行控制。

以上处理中, 虽然扫描区域减小了 (一般地, 扫描区域的面积不到原来图像面积的 $1/10$, 执行算法的时间减少了一个数量级), 而且不需进行 HOUGH 变换, 导航线识别效率提高了, 但是识别的可靠性却降低了。存在的问题主要表现为抗噪性能差: 一方面, 若光照条件不好, 导航图像存在较大的噪声 (主要为白噪声), 那么如果用前面所述的普通的梯度模板就会检出许多的伪边缘点; 另一方面, 是我们更不愿意碰到的情况, 就是路面上存在又一些反光点、气泡、甚至纸屑等“奇异点”。在以上两种情况下, 不加预处理而进行梯度边缘检测出现误检的几率很高, 检出导航线的可靠性很低。

针对以上情况, 我们可以利用图像处理的手段对检测区域的像素点进行预处理, 并在边缘检测之后对结果进行判断。常用的图像处理方法可以通过中值滤波, 滤掉图像中的白噪声, 在通过边缘增强的手段对检测区域进行一次锐化。然后再通过区域扫描, 检出区域中灰度值出现上跳变的点和下跳变的点。求出检出的上跳变点和下跳变点的距离,

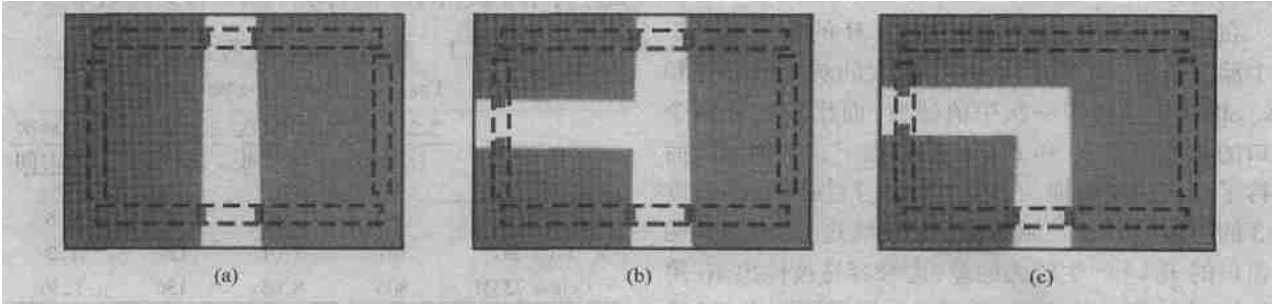


图6 带滤波窗的区域扫描
Fig 6 Region scanning with filters

并与上一帧的结果以及经验值比较，再决定是否接受本次区域检测的结果。然而，以上过程可以结合起来做，即通过一次扫描完成两个功能，而不需要分步做，这就是所谓的“带滤波窗的区域扫描的方法”。具体的算法如下：

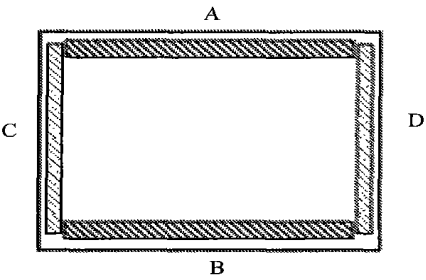


图 7 扫描区域的分步

Fig. 7 Distribution of the scanning regions

- (1) 如图 7 所示建立扫描区域 A，扫描区域的长度为 3 的倍数，记为 $3K$ ，扫描区域的宽度为 M ；
- (2) 令 k 从 1 变化到 K ，那么每一个 k 对应着一个窗口，这些窗口互不重合，每个窗口的大小为 $3 * M$ ；
- (3) 对于第 k 个窗口，统计所有 $3 * M$ 个像素的灰度值的总和，记录为 $f[k]$ ；
- (4) 循环 K 次，直到统计完所有 K 个窗口，得到数组 $f[K]$ ；
- (5) 对于数组 $f[K]$ ，比较相邻两个元素的值 $f[k]$ 和 $f[k-1]$ ；
- (6) 设 T 为一个阈值，若 $(f[k] - f[k-1]) > T$ ，认为发生了上跳变，记录此时的 k 值到数组 U_A ，另外，若 $(f[k] - f[k-1]) < -1$ ，认为发生了下跳变，记录此时的 k 值到数组 V_A ；
- (7) 检查数组 U_A, V_A 中保存的所有上跳变与下跳变，根据导航线的宽度判断扫描区域 A 的形态；
- (8) 同理扫描其他 3 个区域 B、C、D；
- (9) 根据 A、B、C、D 4 个区域的形态判断机器人的航向。

在以上步骤中，每个大小为 $3 * M$ 的窗口相当于一个滤波模板，计算这 $3 * M$ 个像素的灰度值的总和 $f[k]$ ，相当于进行了一次中值滤波，而比较相邻两个窗口的灰度值 $f[k]$ 和 $f[k-1]$ 相当于在中值滤波后进行了一次梯度扫描。在第 1 步中，扫描区域的长度取 3 的倍数的好处是能够使得导航线边缘处，相邻两个窗口的 $f[k]$ 产生较大的差，边缘容易被检出。在第 2 步中，滤波窗口的宽度 M 可以根据需要选择， M 选得太大，将浪费处理的时间， M 选得太小，窗口的面积也小，中值滤波的效果就差。经过实验比较，

我们认为 M 的值选 10 左右为合适。值得注意的是，在第 6 步中，我们提到了一个“阈值”。同前面的讨论一样，这个阈值的确定可以根据经验值来定，也可以动态地根据实时图像统计出来。

3 实验结果及分析

为了控制机器人灵活的行走，计算机不断地对采集到的视频流进行计算，提取导航线的信息，因此，可靠性与实时性是判断导航算法是否可行的重要指标。

3.1 可靠性

就可靠性而言，2.2 所述的边缘提取结合 HOUGH 变换的方法具有较高的检出率，对导航线边缘的检测可以达到 100%。而 2.3 所述的区域扫描方法在某些时候就会失效。

如图 8 所示，(a)、(b)、(c) 是机器人行走过程中先后采集到的 3 幅图像。如果采用 2.3 所述的区域扫描方法，那么在处理 (b) 的时候就会失效了。遇到这种情况，我们常采用的对策就是作“暂时放弃”处理。即若出现无法判断的图像，如把它丢弃，并不向上一级的决策系统报告此时的位置偏差，若连续多幅图像（例如，设定临界值为 3 幅图像）都无法判断，才向上一级的决策系统报告错误。

一般而言，因为机器人在运动当中，因此出现“错误”的图像只会连续 1~2 帧，也就是说，遇到图像 (b) 以后，马上就会出现图像 (c)。而我们又可以重新对图像 (c) 进行有效的识别，提取导航信息。

3.2 实时性

常用摄像机的可以是 NTSC 制式和 PAL 制式，每秒所对应的刷新频率分别是 30 Hz 和 25 Hz，每帧图像的采集周期对应为 33 ms 和 40 ms。这就要求算法的完成时间应该控制在毫秒级。

利用 3 部不同的计算机对主要算法的完成时间进行了测试。为了得到较好的精度，我们把主要算法进行 100 次循环以后再求平均。得到的结果表 1。

表 1 两种算法的时间复杂度

Tah 1 Processing time of the two algorithms

CPU 型号	2.2 算法循环 100 次		2.3 算法循环 100 次	
	总时间	平均时间	总时间	平均时间
	ms	ms	ms	ms
P4 2.8G	360	3.60	60	0.6
P3 1.2G	701	7.01	120	1.2
Celion 733M	805	8.05	136	1.36

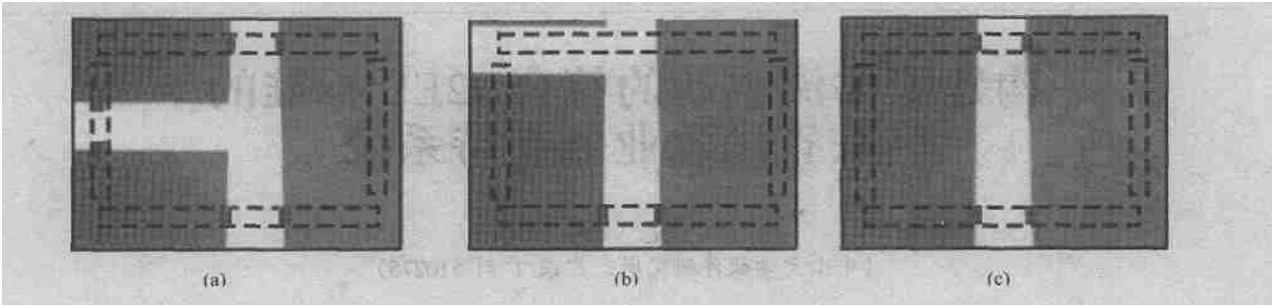


图 8 区域扫描方法的检测结果

Fig 8 Results obtained with region scanning algorithm

事实上，因为每部计算机的软硬件配置不同，算法完成的时间会稍有区别，但表中的数据能定性反映了以下几方面的问题：

(1) 从表 2 中可以看出，两种算法的完成时间都远小于 33 ms，能较好地满足实时性的要求（事实上，计算机除了完成视觉图像识别以外，还需要完成别的工作，因此图像处理的时间控制在 10 ms 以内是较为合适的）；

(2) 2.3 中所述的的算法比 2.2 的算法具有更高的效率；

(3) CPU 的性能越好，算法完成的时间越短。

4 讨 论

以上我们针对机器人比赛的题目提出了两种导航线识别的算法。经过检验，这两种算法都具有较好的实时性，对于直线形态的各类导航线以及对于已知半径的弧形导航线具有较高的可靠性，适用于近年国内外组织的机器人比赛，也可以扩展到更多

的实用领域，如物流业的仓库运输车以及智能交通的无人驾驶汽车等。

以上算法目前对于分支较多的直线形态的导航线以及未知半径的弧形导航线的效果不佳，这是今后努力的方向。

参考文献：

[1] 贾云得. 机器视觉 [M] . 北京：科学出版社，2000.

[2] 崔屹. 图像处理与分析——数学形态学方法及应用 [M] . 北京：科学出版社，2000.

[3] 蔡自兴. 机器人学 [M] . 北京：清华大学出版社，2000.

[4] SONKA M, HLAVAC V, BOYLE R. Image processing, analysis, and machine vision [M] . 2nd Edi. Brooks/Cole, 2002.

[5] NAVON R. Process and quality control with a video camera, for a floor-tilling robot [J] . Automation in Construction, 2000; 113—125.

A Robot Navigation Algorithm Based on Computer Vision

TAN Xiao jun, GUO Zhi hao

(Department of Electronics and Communication Engineering, Sun Yat sen University, Guangzhou 510275, China)

Abstract: The authors here put forward two algorithms that are used in navigation of an auto robots. These algorithms were once successfully used in some robot contests and performed considerable reliability. Compared with the traditional methods of image processing, the algorithms have better pertinence and better performance in real time.

Key words: computer vision; robot navigation; AGV