

# 基于 ARM 的视觉导航 AGV 图像处理方法研究

文生平, 陈志鸿, 张施华

(华南理工大学 聚合物新型成型装备国家工程研究中心 聚合物成型加工工程教育部重点实验室, 广州 510640)

**摘要:** 为了实现更加简单高效的标识线图像的检测与处理, 提出了一种优化的基于 ARM 的视觉导航 AGV 标识线图像处理方法; 首先对采集到的图像进行灰度化处理并使用 Otsu 算法对图像进行阈值分割; 然后采用优化的中值滤波算法进行图像滤波并使用高效的边缘提取策略获取路径边缘特征; 最后采用角度判断的方法剔除错误点并使用最小二乘法拟合合成路径的中心线; 实验结果表明, 该方法有较高的准确率和较好的实时性, 可以满足工业生产中的实际需求, 适用于基于嵌入式系统开发的视觉导航 AGV。

**关键词:** 视觉导航; 边缘提取; 最小二乘法; 中值滤波

## Research on Image Processing Method of ARM-based Visual Navigation AGV

Wen Shengping, Chen Zhihong, Zhang Shihua

(National Engineering Research Center of Novel Equipment for Polymer Processing, Ministerial Key Laboratory of Polymer Processing Engineering, South China University of Technology, Guangzhou 510640, China)

**Abstract:** In order to achieve a more simple and efficient detection and processing of the navigation line image, an optimized navigation line image processing method for ARM-based visual navigation AGV is proposed. First, image collected by the camera is grayed and then segmented by Otsu algorithm. Second, the optimized median value filter algorithm is used to denoise and the efficient edge extraction strategy is used to obtain the path edge. Finally, the error points are eliminated by the method of angle judgment, and the central line of the path is synthesized by the least square method. The experimental results show that the method has high accuracy and good real-time performance, and can meet the needs of industry. It is suitable for visual navigation AGV based on embedded system.

**Keywords:** visual navigation; edge extraction; least square method; median filtering

## 0 引言

AGV (automated guided vehicle) 即自动导引小车, 是一种通过装备光学、电磁等导引装置, 实现在无人干预下沿着既定路径自动行驶的智能化运输设备<sup>[1]</sup>。近年来随着计算机视觉与图像处理技术的快速发展, 使得视觉导航 AGV 成为一个发展热点, 其出色的灵活性与智能化使其广泛应用于自动化生产线、仓储物流、港口物流传送等领域<sup>[2]</sup>。

目前视觉导航 AGV 主要是利用视觉传感器实时采集路面的图像信息, 通过对采集到的图像进行数字图像处理, 识别出车道标识线并获取 AGV 相对于标识线中心线的位置偏移量与角度偏差, 再由控制器发出指令对偏差进行修正, 使 AGV 沿着标识线中心线运动<sup>[3]</sup>。

在图像处理与视觉导引方面, 通常采用以下几种方法:

- 1) 对图像上的标识线进行边缘特征提取, 提取的结果直接作为控制的依据。
- 2) 对图像上的标识线进行边缘特征提取后进行曲线拟合, 拟合的结果作为控制的依据<sup>[4]</sup>。
- 3) 对图像上的标识线进行边缘特征提取, 通过在特征边

缘处找到导引点作为控制的依据。

针对基于嵌入式系统开发的视觉导航 AGV<sup>[5]</sup>, 由于其计算与存储资源有限, 对图像处理算法的执行效率有很高的要求。在确保标识线有高识别准确率的前提下, 需要尽可能降低系统资源的开销, 以提高系统的实时性。

本文提出一种高效的视觉导航 AGV 图像处理方法, 高效提取标识线边缘并进行中心线快速拟合, 该算法有较高的准确率和较好的实时性。

## 1 系统总体设计

AGV 系统包括图像采集单元、图像处理单元、无线局域网模块、控制模块、I/O 模块、底层硬件等, 如图 1 所示。

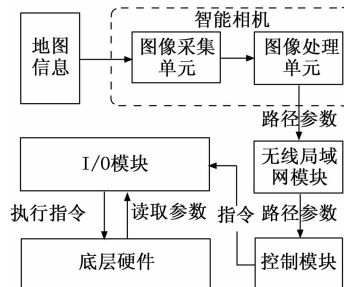


图 1 AGV 系统总体架构

地图信息指包含标识线的场景, 图像处理的目的是实时地将地图信息转化为控制器可读的路径参数, 为视觉导引提供控制的依据。

收稿日期: 2017-05-12; 修回日期: 2017-05-31。

基金项目: 广东省及广州市科技计划项目 (2015B090901020; 201508010058; 201604010079)。

作者简介: 文生平 (1966-), 男, 湖北荆州人, 教授, 硕士研究生导师, 主要从事工业装备的智能控制与机器视觉方向的研究。

图像采集单元负责以一定频率与分辨率采集带有标识线的图像,并将采集到的图像传输到图像处理单元。

图像处理单元负责对采集到的每一帧图像进行图像预处理、特征提取、直线拟合等操作,最终获得小车相对于标识线中心线的角度偏差与位置偏移量,并打包成下位机可读取的路径参数。

无线局域网模块负责将获得的路径参数发送到控制模块。

控制模块根据接收到的路径参数输出控制指令纠正小车当前的位姿。同时,控制模块读取当前的电机状态、电量水平、运行速度等相关参数并实时调整小车的运行状态。

I/O 模块负责传达控制指令到底层硬件模块,同时读取底层硬件当前的电机状态、电量水平、运行速度等参数到控制模块。

底层硬件接受控制指令,并按照控制指令进行动作。

AGV 结构模型如图 2 所示,驱动轮安装在转向装置上。配有环形光源的图像采集模块垂直于地面安装在小车的转向装置下方,通过测量图像采集区域下的  $d$  与  $\theta$ ,得到小车相对于标识线的位置偏差与角度偏差。控制器实时输出控制指令,控制转向装置转动使小车能够始终沿着车道线行驶。

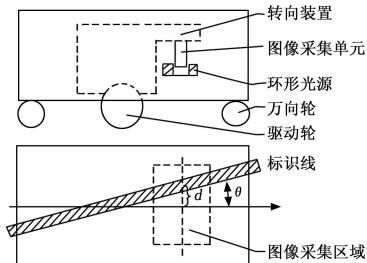


图 2 AGV 结构模型

## 2 标识线导引的具体实现

视觉导航 AGV 的标识线导引需要对采集回来的路径图像进行处理,获得有用的路径参数并传递到控制器,实现导航功能。具体包含以下步骤:

### 2.1 图像预处理

图像预处理<sup>[6]</sup>主要包括了几何校正、灰度化、阈值分割、图像滤波等处理。由于视觉传感器垂直于地面安装,其采集到的图像因透视而产生的形变较少,故可以不进行几何校正,而需要进行灰度化、阈值分割与图像滤波。

#### 2.1.1 图像灰度化

将彩色图像转化为灰度图像的过程称为图像的灰度化处理。图像传感器采集到的图像是 24 位 RGB 彩色图像,这样的彩色图像将占用大量的存储资源与计算资源,因此我们需要对图像进行灰度化处理以降低系统资源的消耗。灰度化的方法有分量法、最大值法、平均法与加权平均法。按照实验与经验的推荐,按下式对  $R$ 、 $G$ 、 $B$  三个分量进行加权平均能得到较理想的灰度图像。

$$f(i, j) = 0.299R(i, j) + 0.587G(i, j) + 0.114B(i, j) \quad (1)$$

尽管灰度化后的图像相对彩色图像丢失了一定的色彩等级,但其仍然能反映整幅图像的整体和局部的色彩和亮度等级的分布和特征。对于仅含有路径标识线的图像而言,灰度化后对整体和局部的图像描述产生的影响可以忽略不计。

#### 2.1.2 阈值分割

阈值分割指把图像分割为前景和背景。分割的方法可以分

为全局阈值分割和局部阈值分割两大类。全局阈值分割通过选取一个阈值  $T$ ,使图像  $g(x, y)$  像素灰度值大于此阈值  $T$  的设为 255,小于此阈值的设为 0。公式如下:

$$g(x, y) = \begin{cases} 255, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \quad (2)$$

全局阈值有多种方法,其中 Otsu 算法<sup>[7]</sup>被认为是图像分割中阈值选取的最佳算法。其算法计算简单,不受图像的亮度和对比度影响,降低了错分的概率。本文利用 Otsu 法分割标识线与背景,效果明显,如图 3 (b) 所示。

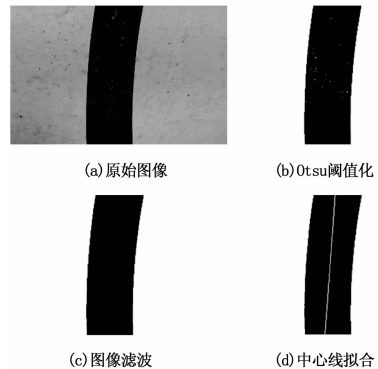


图 3 实验结果

经 Otsu 法阈值分割后的图像背景灰度值为 255 (白色),标识线灰度值为 0 (黑色),分割准确,对比度高,能够最大限度地节约系统存储空间与提高图像的处理速度。

#### 2.1.3 基于均值查找的中值滤波法

中值滤波<sup>[8]</sup>是基于排序统计理论的一种能去除噪声的非线性信号处理技术。其原理是把图像中某一像素点的值用该点的邻域各点的值的中值代替,从而消除孤立的噪声点,对于斑点噪声与椒盐噪声的去除效果尤为突出。

中值滤波的执行效率依赖于排序的运算速度。排序需要遍历像素点及其邻域的像素值,再通过比较每个像素灰度值的大小进行排序。算法本身十分耗时,难以保证实时性。为此文中采用一种基于均值查找的中值滤波算法<sup>[9]</sup>。其算法的思想是:

以像素灰度值集合  $M$  的均值  $avg$  划分  $M$  为相应的两个子集合。通过计算,分别获得子集合元素个数  $n$  与子集合累计舍弃元素个数  $m$ ,并分别与初始集合元素个数  $q$  的一半进行比较,以确定中值所在的目标集合  $M'$ ,并对此集合进行下一次迭代,直到目标集合只包含一个元素为止。该元素则为集合  $M$  的中值。算法示例如图 4 所示。此方法通过集合的多次二分,实现缩小目标集合从而减少不必要的排序操作,大大提高了中值滤波算法的执行效率。

### 2.2 特征提取

要获得车道标识线的中心线首先需要对路径的边缘特征进行提取。霍夫变换是图像处理中一种经典的特征提取技术,用于检测图像中的特定形状<sup>[10]</sup>。对于可近似看作直线的车道标识线,其边缘可以通过霍夫变换检测出来。虽然霍夫变换的应用很广泛,但是其对图像边缘的细节要求较高,并且计算量大,难以达到高实时性的要求。故文中采取一种高效的边界轮廓扫描法,方法如下:

Step1: 对图像自上而下逐像素进行扫描,本方法中图像共有 380 行像素。扫描同时从左右两端开始,方向指向内

部, 如图 5 所示, 获取各个像素点的灰度值。

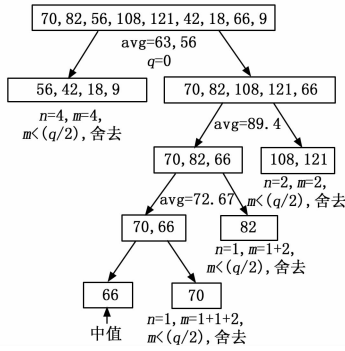


图 4 基于均值查找的中值算法示例图

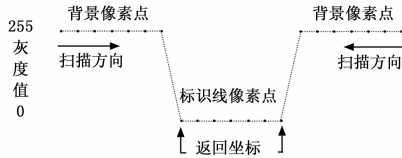


图 5 行扫描示意图

Step2: 当扫描首次遇到像素灰度值为 0 的像素点则返回当前像素点的坐标值, 并插入到向量中。理想情况下将返回标识线两侧边缘的坐标值。如此自上往下逐行扫描最终得到分别表示标识线左右边缘的两个特征向量组  $vec_{lj}$  与  $vec_{rj}$ , ( $j=0, 2, 3, \dots, 379$ )。

Step3: 对向量组  $vec_{lj}$  与  $vec_{rj}$  分别进行元素数量检查, 当元素数量小于 190 (像素行总数的一半) 时, 认为没有找到标识线的边缘, 跳出此步骤。当元素数量大于 190 时, 认为找到标识线边缘并进行下一步操作。

Step4: 向量组  $vec_{lj}$  与  $vec_{rj}$  分别记录了标识线左右边缘的坐标。对两组向量中的坐标值进行数学运算可以得到标识线中线的坐标集合。记该坐标集合为  $Mid$ , 对  $Mid$  进行直线拟合则得到标识线中心线。 $Mid$  可以表示为:

$$Mid = \{(x_i, y_i) \mid x_i = (vec_{lj} + vec_{rj})/2, y_i = j; j = 0, 2, \dots, 379\}$$

### 2.3 直线拟合

当路径为曲线时, 取小段路径进行分析, 曲线路径可以近似为直线处理。由于图像采集单元安装位置靠近地面, 视野较小, 其连续采集到的标识线图像可近似为直线, 于是路径的拟合问题可看作是线性回归的问题。线性方程为:

$$y = a + bx \quad (3)$$

根据最小二乘法, 其偏差平方和为:

$$\partial \sum_{i=1}^n v_i^2 = \sum_{i=1}^n (y_i - a - bx_i)^2 \quad (4)$$

式中, 分别对  $a$  与  $b$  求偏导数, 有:

$$\begin{cases} \frac{\partial \sum_{i=1}^n v_i^2}{\partial a} = -2 \sum (y_i - a - bx_i) \\ \frac{\partial \sum_{i=1}^n v_i^2}{\partial b} = -2 \sum (y_i - a - bx_i) \cdot x_i \end{cases} \quad (5)$$

于是得到关于  $a, b$  的线性方程组:

$$\begin{cases} na + b \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \\ a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 = \sum_{i=1}^n y_i x_i \end{cases} \quad (6)$$

解方程, 得  $a, b$ :

$$\begin{cases} b = \frac{s_{xy}}{s_{xx}} \\ a = \bar{y} - b\bar{x} \end{cases} \quad (7)$$

其中:

$$s_{xy} = \sum x_i y_i - \frac{\sum x_i \sum y_i}{n}, \bar{x} = \frac{\sum x_i}{n}, \bar{y} = \frac{\sum y_i}{n},$$

$$s_{xx} = \sum x_i^2 - \frac{(\sum x_i)^2}{n}$$

最小二乘法是进行直线拟合最简单有效的方法, 但其缺点是容易受到错误点的影响从而影响直线的准确度, 为此需要对算法的使用进行改进。为了能够实现对错误点的判断与剔除, 文中采用最小二乘法分段拟合直线, 通过比较各段线段的角度对错误点进行定位, 并剔除错误点所在的小段路径, 最后对保留下来的路径小段重新进行拟合, 得到最终直线段。该方法如下:

Step1: 将像素点坐标集合  $Mid$  在  $y$  方向上每十个像素点分成一段, 得到如下 38 个坐标集合:

$$Mid_0 = \{(x_i, y_i) \mid x_i = (vec_{lj} + vec_{rj})/2, y_i = j; j = 0, 2, \dots, 9\}$$

$$Mid_1 = \{(x_i, y_i) \mid x_i = (vec_{lj} + vec_{rj})/2, y_i = j; j = 10, 12, \dots, 19\}$$

$$Mid_{37} = \{(x_i, y_i) \mid x_i = (vec_{lj} + vec_{rj})/2, y_i = j; j = 370, 372, \dots, 379\}$$

Step2: 使用最小二乘法分别对  $Mid_0$  到  $Mid_{37}$  进行直线拟合, 得到 38 段直线段  $L_0, L_1, \dots, L_{37}$ 。

Step3: 设定一个角度阈值  $T$  来区分角度正常与角度异常的直线段, 设定一个计数器  $f$  来统计错误线段的总数, 计算各线段的斜率  $k$ , 线段角度  $\theta_i$ , 并计算相邻直线段的角度之差的  $\Delta\theta_i$ 。公式如下:

$$\theta = \arctan(k) \quad (8)$$

$$\Delta\theta_i = \begin{cases} 0 & i = 0 \\ |\theta_i - \theta_{i-1}| & i \neq 0 \end{cases} \quad (9)$$

Step4: 对  $L_0$  是否存在错误点进行判断。当  $L_0$  中包含错误点时, 会导致后面线段的错误点的判断失败, 对于这种情况, 进行如下的处理: 若  $\Delta\theta_1 > T$  且  $\Delta\theta_2 < T$ , 则认为  $L_0$  中包含错误点, 剔除  $L_0$  所对应的像素点集合并重新执行计算。

Step5: 比较阈值  $T$  与  $\Delta\theta_i$  的大小, 若  $\Delta\theta_i$  大于阈值  $T$ , 则认为线段  $L_i$  中出现了错误点, 导致线段的角度出现异常, 剔除线段  $L_i$  所对应的像素点集合  $Mid_i$ , 同时计数器  $f$  加 1。若  $\Delta\theta_i$  小于阈值, 则将集合  $Mid_i$  保存至向量  $vec$  中。

Step6: 检查计数器  $f$ , 当  $f < 19$ , 即当存在错误点的线段数量小于线段总数的一半时, 认为错误点对路径拟合没有影响, 并对  $vec$  中的坐标点集合进行最小二乘拟合, 得到最终直线。当存在错误点的线段数量大于线段总数的一半时, 认为错误点对路径拟合产生影响, 跳出程序, 读入下一帧进行下一循环。

算法流程如图 6 所示, 流程与上述步骤一一对应, 算法在

Linux 操作系统下采用 C++ 进行编写。

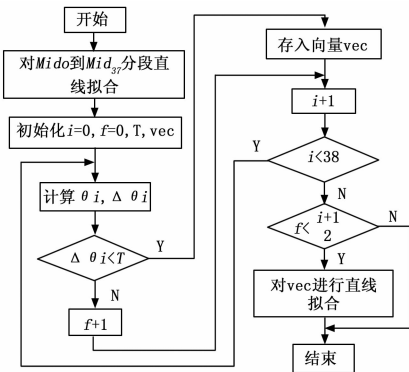


图 6 直线拟合算法流程图

### 3 实验结果及分析

#### 3.1 实验平台

为了验证本文提出的算法效率以及正确性，使用树莓派三代，树莓派摄像头以及 Turtlebot 轮式机器人进行实验。树莓派三代是基于 ARM 的微型电脑，其 CPU 为 ARM Cortex-A53，并且配有 1GB 内存和无线网卡模块，搭配树莓派摄像头，相当于智能相机，能够完成图像采集、图像处理及无线通讯的功能。图像处理后获得的路径数据通过无线局域网模块发送至 Turtlebot 轮式机器人的控制模块进行整车调试。

#### 3.2 结果分析

利用上述的图像处理方法，如图 3 所示是对路径标识线图像处理的过程与结果。可以看到原始图像是一段曲率较小的曲线，路面上存在大量干扰。使用 Otsu 法进行阈值化后可以看到路径与背景已经很好地分割开来，路径呈黑色，背景呈白色，但是图像上仍存在大量的斑点噪声与椒盐噪声。经过优化的中值滤波处理之后在中心线拟合的图像中可以看到大部分干扰因素已经被很好地去除，余留下的少量干扰因素因为被直线拟合算法排除，没有影响到中心线的拟合，使本算法的鲁棒性得到保证。

对本图像处理方法（优化的中值滤波、高效边缘提取与最小二乘直线拟合）与传统的图像处理方法（传统的中值滤波、Hough 算法直线检测与最小二乘直线拟合）进行对比实验。计算分别用以上两种方法连续处理 1000 帧图像的合计用时与平均用时，以及统计最终直线拟合的成功率，结果如表 1、表 2 与表 3 所示。

表 1 中值滤波算法处理速度比较

|         | 传统中值滤波  | 优化的中值滤波 |
|---------|---------|---------|
| 合计用时/ms | 206.324 | 141.089 |
| 平均用时/ms | 0.2063  | 0.1411  |
| 效率提升/%  | /       | 31.60   |

表 2 直线拟合方法处理速度比较

|         | Hough 直线拟合 | 优化的直线拟合 |
|---------|------------|---------|
| 合计用时/ms | 210.808    | 89.742  |
| 平均用时/ms | 0.2108     | 0.0897  |
| 效率提升/%  | /          | 57.45   |

表 3 中心线拟合成功率比较

|        | Hough 直线拟合 | 优化的直线拟合 |
|--------|------------|---------|
| 图像帧数   | 1000       | 1000    |
| 处理成功帧数 | 866        | 986     |
| 成功率/%  | 86.6       | 98.6    |

从实验结果可以看到本图像处理方法在处理效率上明显优于传统图像处理方法。就单帧图像而言，优化的中值滤波较传统中值滤波效率提升 31.60%，而本直线拟合方法较 Hough 直线检测加最小二乘直线拟合效率提升 57.45%。中心线拟合依赖于图像预处理效果以及标识线边缘特征提取的完成度，由于本边缘提取算法对边缘细节要求较 Hough 算法低，导致最终中心线拟合的成功率较传统方法高 12%。在实际工作场景中连续处理的图像数量远大于 1000 帧，其效率的提升是相当可观的，这对于计算与存储资源有限的嵌入式系统显得尤为重要。

### 4 结论

本文针对嵌入式系统计算与存储资源有限以及视觉导航 AGV 实际工作环境所要求的高实时性的问题，提出了一种可行的视觉导航 AGV 图像处理方法。该方法采用了 Otsu 算法对图像进行阈值分割；使用优化的基于均值查找的中值滤波算法进行图像滤波；使用高效的边缘提取策略获取路径边缘信息；采用分段角度判断的方法剔除错误点并使用最小二乘法拟合成路径的中心线，大大提高了系统的运行效率，保证了系统的鲁棒性与实际工作中所要求的准确性与实时性，有利于在基于 ARM 的视觉导航 AGV 上的应用。

#### 参考文献：

[1] 李西锋, 魏生民, 闫小超. 视觉引导 AGV 的数字图像处理方法研究 [J]. 科学技术与工程, 2010, 10 (10): 2515-2519.

[2] Xing W, Lou P, Yu J, et al. Intersection Recognition and Guide-path Selection for a Vision-based AGV in a Bidirectional Flow Network [J]. International Journal of Advanced Robotic Systems, 2014, 11 (1): 1.

[3] Zhang H B, Yuan K, Shu Q M, et al. Visual navigation of an automated guided vehicle based on path recognition [A]. International Conference on Machine Learning and Cybernetics [C]. IEEE Xplore, 2004, 6: 3877-3881.

[4] 朴磊, 李剑中, 冯刚. 视觉引导 AGV 的数字图像处理 [J]. 仪器仪表学报, 2006, 27 (s1): 766-767.

[5] 王宜怀, 吴瑾, 蒋银珍. 嵌入式系统原理与实践 [M]. 北京: 电子工业出版社, 2012.

[6] 王红君, 施楠, 赵辉, 等. 改进中值滤波方法的图像预处理技术 [J]. 计算机系统应用, 2015 (5): 237-240.

[7] Otsu N. A Threshold Selection Method from Gray-Level Histograms [J]. IEEE Transactions on Systems Man & Cybernetics, 2007, 9 (1): 62-66.

[8] Hanji G, Latte M V. Novel Median filter for impulse noise suppression from digital images [J]. International Journal of Computer Applications, 2014, 4 (3/4): 1-12.

[9] 鲍华, 樊瑜波, 饶长辉, 等. 基于均值查找的快速中值滤波算法 [J]. 四川大学学报 (工程科学版), 2011 (2): 76-79, 86.

[10] Aggarwal N, Karl W C. Line detection in images through regularized Hough transform [J]. IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society, 2006, 15 (3): 582-91.