

嵌入式双目立体视觉测距系统^①

文成维, 陈文建, 詹 泳

(华南师范大学 计算机学院, 广州 510631)

摘 要: 设计和研究了一种基于 S3C2440 硬件平台和嵌入式 Linux 双目立体视觉测距系统. 该系统通过使用两个 USB 摄像头同步采集图像数据, 并利用 OpenCV 对采集的两幅图像的数据进行处理, 计算目标物在两幅图像中的像素偏差, 进而算出目标物到摄像头平面的距离, 在论文中以车牌作为目标物. 同时该系统用 Qt/Embedded 实现 LED 实时显示图像和数据. 整个系统建立在嵌入式架构上, 能独立完成图像采集, 数据处理, 实时显示等功能. 只要改变定位算法, 则能测出任意目标物的距离.

关键词: 双目立体视觉; 双目测距; 嵌入式 Linux; OpenCV; 定位算法

Embedded Binocular Stereo Vision Ranging System

WEN Cheng-Wei, CHEN Wen-Jiang, ZHAN Yong

(School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: The paper designs and researches a binocular stereo measuring system based on S3C2440 hardware platform and embedded Linux. The system captures images data in synchronization by using 2 USB cameras, and then processes the images data in OpenCV Lib. Finally, calculate the pixel deviation of the target in two images, and then calculate the distance from the target to camera plane, license plate be seen as the target in the paper. At the same time, the system displays the images and dates on LED with Qt/Embedded GUI in real time. The whole system is built on an embedded architecture, able to capture images and process data and real-time display data independently. When change location algorithm, we can calculate the distance of any targets.

Key Words: binocular stereo vision; binocular ranging; embedded Linux; location algorithm

1 引言

随着嵌入式技术的迅猛发展, 嵌入式设备已经广泛应用于汽车电子、消费电子、军工航天等领域. 嵌入式 Linux 由于其开源性、可裁剪、可移植、实时性强、内核小等特点, 成为嵌入式设备的首选操作系统^[1]. 双目立体视觉测距是计算机视觉技术中的一个重要分支, 也是计算机视觉研究的重点和热点之一, 并且已经取得了不少的成果. 双目立体视觉是仿真人眼的视觉系统, 具有非接触、高精度、深度可感知等优点.

本论文在以嵌入式 Linux 作为操作系统和以 S3C2440 为核心硬件系统的基础上, 把嵌入式技术和

计算机视觉技术相结合. 使用双 USB 摄像头作为图像采集模块, 利用 OpenCV 进行图像数据处理, 基于 Qt/Embedded 实现图像和数据在 LCD 上的实时显示. 本系统具有体积小、成本低、稳定性高等优点, 可广泛应用于理测绘、工业检测、机器人导航、三维测量等诸多领域.

2 系统架构

本系统由系统硬件和系统软件两大部分构成. 系统硬件由以 S3C2440 作为核心处理器加上外围设备组成; 系统软件由引导加载程序、嵌入式 Linux、函数库、应用程序组成.

^① 基金项目: 华南师范大学学生课外科研课题基金(11JJC03)

收稿时间: 2012-02-24; 收到修改稿时间: 2012-03-31

2.1 系统硬件架构

系统硬件架构图如图 1 所示, 系统采用 S3C2440 作为核心处理器. S3C2440 是三星公司设计的一款基于 ARM920T 内核的 32 位嵌入式 RSIC 微处理器, 它的最高工作频率可达 533MHz, 并且内部集成了通用 LCD 控制器、8 通道 10 位 ADC 和触摸屏接口, 具备性能高、低功耗的特点^[2]. 此外, 系统配置了 256MB 的 NAND FLASH 和 64MB32 位的 SDRAM, 满足系统存储引导加载程序、Linux 内核, 文件系统、应用程序以及数据和进行图像数据处理的需求. 系统使用中星微公司 ZC0301 芯片的 USB 摄像头采集图像数据; 并且配备分辨率为 320*240 的 LCD 显示器, 实时显示图像和数据.

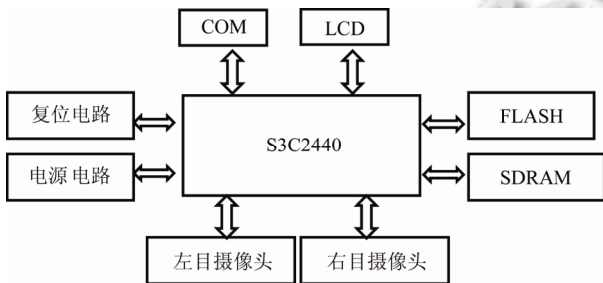


图 1 系统硬件架构图

2.2 系统软件架构

系统软件由 u-boot 引导加载程序、嵌入式 Linux 内核、设备驱动程序、文件系统 cramfs、GNU C/C++ 库、OpenCV 库、Qt/Embedded 库和应用程序组成. 如图 2 所示.

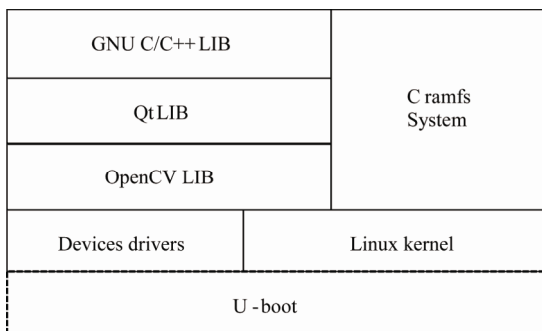


图 2 系统软件架构图

系统上电后, 先运行引导加载程序 U-boot, 该引导加载程序会初始化硬件设备、建立内存空间映射表, 然后将内核启动命令参数传递给内核并引导加载嵌入式 Linux 内核. 启动嵌入式 Linux 内核后, 接着加载

NAND FLASH 驱动程序、LCD 驱动程序、Ethernet 网卡驱动程序、USB 摄像头驱动程序等, 最后挂载所需的文件系统, 执行应用程序^[1].

3 系统测距原理

3.1 针孔成像原理

三维空间中的物体到像平面的投影关系即为成像模型^[3]. 最简单的摄像相机模型是针孔成像模型, 即假定物体表面的反射光都经过一个针孔而投影在像平面上. 如图 3 所示.

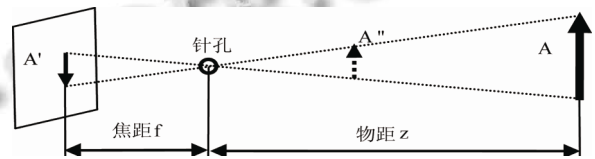


图 3 针孔成像原理图

物体 A 经过针孔在在屏幕上的成像是倒立的 A', 成像 A' 到针孔的距离是焦距 f, 物体 A 到针孔的距离是物距 z.

但是, 针孔成像由于透光量太小, 需较长时间曝光, 且很难得到清晰图像. 实际上, 摄像机中都使用透镜解决上述的问题, 因此实际摄像系统中通常是由透镜或透镜组合构成的. 图 4 是理想单透镜成像原理图.

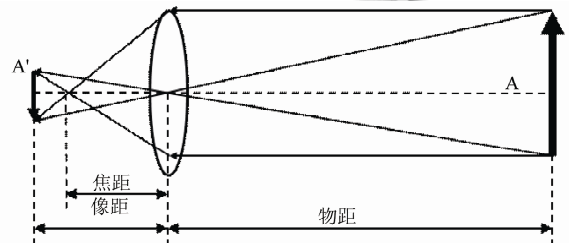


图 4 理想单透镜成像原理

比较上述可以看出, 透镜成像与针孔成像两者的焦距并不是一个概念, 在针孔成像中焦距等于像距, 而在透镜成像中, 焦距并不等于像距. 但两者的关系是一致的, 即像点是物点和光心与图像平面的交点, 所以仍可以把针孔成像模型作为摄像机成像模型.

3.2 平行双目立体视觉模型

如下图 5 所示, 左目摄像头位于 O_l , 右目摄像头位于 O_r , 以 $O_l O_r$ 线段中心点为原点, 建立三维坐标系. 其中两摄像机光轴平行于 z 轴, 两摄像机镜头平面平行于 xOy 平面. 两个平行摄像机镜头中心的距离是 L,

焦点距离是 f , 三维空间的任意一点 $P(x, y, z)$ 在左图像上的投影是点 (x_l, y_l) , 在右图像上的投影是点 (x_r, y_r) . 镜头之间距离的中心点作为三维空间的原点 O , x 轴平行于水准线, z 轴平行于光轴^[4].

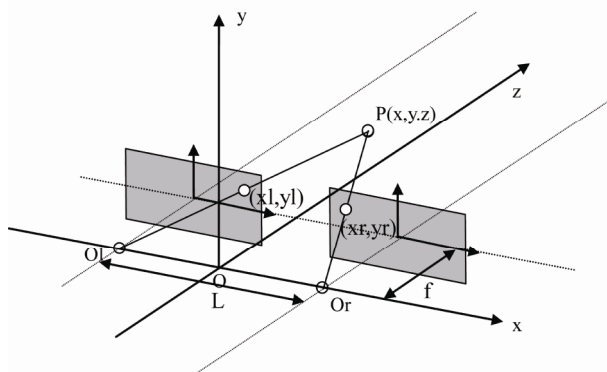


图 5 双目立视觉测距模型图

为了便于观察, 将三维空间中的点 P 投影到平面 xOy 投影点是 P' 简化图如图 6 所示, 底边为 L , 高为 z 的三角形与底边为 $L - (|x_l| + |x_r|)$, 高为 $z - f$ 的三角形是相似三角形. 所以根据相似三角形的比例关系有:

$$L : z = (L - (|x_l| + |x_r|)) : (z - f)$$

即: $z = fL / (x_l - x_r)$, z 值即是三维空间中的被测目标物体距离摄像头镜面平面的距离^[5].

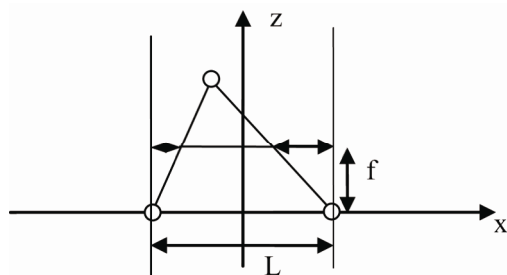


图 6 模型简化图

令 $d_x = x_l - x_r$, d_x 是视差值, 即左右两幅图像上相同的特征在 x 坐标上的差值. 因此只要计算出被测目标物体在左右图像中的像素视差值, 即可确定被测目标物体与摄像头镜面平面的距离.

4 系统实现

4.1 图像采集及编码

系统图像采集模块使用双 USB 摄像头, 其核心部件是中星微公司的 ZC0301 芯片.

CMOS 传感器采集到的数据经过处理后, 使用

JPEG 硬件编码器编码成 JPEG 格式的图像数据输出. 因此, 系统启动并打开 USB 摄像头设备后, 将 JPEG 格式编码的图像数据读入系统, 然后继续进行下一步的图像数据处理.

4.2 摄像机标定和图像校正

在实际实现中左、右目摄像机是不可能像图 5 所示的双目立体视觉测距模型中的左、右目摄像机那样严格向前平行. 并且由于工艺的原因, 摄像机透镜可能存在透镜畸变. 因此, 有必要对摄像机进行参数标定^[3], 通过数学计算方法使左、右目摄像机严格向前平行对准和消除透镜畸变.

二维空间一点表示为 $m = [u, v]^T$, 三位空间一点表示为 $M = [X \ Y \ Z]^T$. 点向量加 1, 则: $\tilde{m} = [u, v, 1]^T$, $\tilde{M} = [X \ Y \ Z \ 1]^T$. 在针孔摄像机模型中世界坐标系中一点 M 在图像坐标系中投影点是 m , 满足关系式:

$$s\tilde{m} = A[R, t]M, \quad (1)$$

其中, s 是比例因子, R 是旋转向量, t 是平移向量, (R, t) 是外参数, A 是内参数, 其表示如下:

$$A = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f/d_x & \gamma & u_0 \\ 0 & f/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

其中, f 是焦距, γ 是倾斜因子, d_x 和 d_y 是单位像素尺寸, $\alpha = f/d_x$, $\beta = f/d_y$ 分别为 x, y 方向的比例因子, (u_0, v_0) 是图像像素坐标系的主点坐标. 假设 $H = A[R|t]$, 则矩阵 H 是单应性矩阵. 根据(1)、(2)有如下关系:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f/d_x & \gamma & u_0 \\ 0 & f/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3)$$

由于现实中使用摄像机不可能是理想的针孔摄像机模型, 因此必须考虑摄像的透镜畸变, 透镜畸变一般有径向畸变和切向畸变. 根据 Brown 方法建立模型:

$$\tilde{u} = u(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1v + p_2(r^2 + 2u^2)$$

$$\tilde{v} = v(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2v^2) + 2p_2u$$

其中, $r^2 = u^2 + v^2$, (u, v) 是图像像素坐标系中原点, (\tilde{u}, \tilde{v}) 是校正后的新点, $D = [k_1 \ k_2 \ k_3 \ p_1 \ p_2]$ 是畸变矩阵.

根据张氏标定方法^[6], 假定模板平 1 面位于世界

坐标系中,且 $Z=0$, 并且用向量 r_i 来表示 R , 则由(3)化简得:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[r_1 \ r_2 \ r_3 \ t] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A[r_1 \ r_2 \ t] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix},$$

相应的单应性矩阵为 $H = A[r_1 \ r_2 \ t]$. 每幅给定的平面图像都可以获得一个单应性矩阵, 记 $H = [h_1 \ h_2 \ h_3]$, 求出的 H 与真正的单应性矩阵相差一个比例因子, 故有:

$$[h_1 \ h_2 \ h_3] = \lambda A[r_1 \ r_2 \ t]$$

由于 r_1 与 r_2 正交, $r_1 r_2 = 0$. $\|r_1\| = \|r_2\| = 1$, 所以每幅图像可以获得两个对内参数的矩阵的基本约束, 即:

$$\begin{aligned} h_1^T A^{-T} A^{-1} h_2 &= 0 \\ h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2 \end{aligned}$$

$A^{-T} A^{-1}$ 是对称矩阵, 可以用一个六维 b 向量表示.

如果有 n 幅图像, 就可以构成一个 $2n \times 6$ 的矩阵, 只要 $n > 3$ 就可以解出 b . 一旦 b 知道了, 摄像机的内外参数就可以求解出.

4.3 图像处理及目标定位

本论文中, 我们将车牌作为目标物. 这里提出了一种基于边缘和灰度值的车牌定位算法, 通过这种方法可以在复杂的背景中很好的实现车牌定位^[7].

4.3.1 特征提取及图像分割

经过初步处理的灰度图像中的车牌由许多的字符组成, 所以车牌区域蕴含丰富的水平边缘信息, 而垂直边缘较少. 只要提取图像中的垂直边缘, 将减少干扰. 我们这里使用垂直 sobel 算子提取图像中的垂直边缘.

由于单个像素能提供的信息太少, 我们将图像分割成正方形小方块. 假设小方块的边长是 n 像素. 对于每一个方块, 分别取小方块中灰度值最高的 $n^2/2$ 像素的灰度值的平均值和灰度值最低的 $n^2/2$ 个像素的灰度值的平均值作为小方块的特征值, 分别记为 E_1 和 E_2 . 公式表示如下:

$$E_2 = \frac{1}{n^2/2} \sum_{i=1}^{n^2/2} v_i, \quad E_1 = \frac{1}{n^2/2} \sum_{i=n^2/2+1}^{n^2} v_i,$$

其中, v_i 表示小方块中灰度值第 i 大的像素灰度值.

通常在车牌区域背景或字体有一个的像素个数所占比例是大于 $1/2$ 的, 故而在一个小方块中通常只有一个特征值是可靠的, 因此我们将根据小方块的特征

值决定是否连接小方块.

4.3.2 连接相似区域

完成图像分割后我们将连接小方块并找出可能的车牌区域. 对于小方块的连接我们有以下的步骤:

1) 计算出当前小方块的特征值 E_1 和 E_2 . 初始设置两个特征值标志 `heightAdjust` 和 `lowAdjust` 均为 `false`, 表示此时尚未确认二者的可靠性;

2) 向右进行匹配连接. 倘若边缘小方块右边的小方块的 E_1 和 E_2 与当前小方块的 E_1 和 E_2 值有一个相差小于 T , 我们可以将新的小方块并入. 并且如果二者中差值最小的对应的 `heightAdjust` 或 `lowAdjust` 为 `false` 则设为 `true`, 否则我们可以按一定的权值改变当前的值;

3) 倘若二者的差值均大于阈值, 且 `heightAdjust` 或 `lowAdjust` 中只有一个为 `false`, 只需简单的将该值对应的值设为当前小方块的值并将该值设为 `true`. 如果 `heightAdjust` 和 `lowAdjust` 均为 `false`, 将 E_1 和当前小方块的 E_2 值的差与 E_2 和当前小方块 E_1 的差中绝对值较大的一组设为新的 E_1 和 E_2 , 并设 `heightAdjust` 和 `lowAdjust` 值为 `true`. 否则连接失败;

4) 连接失败时, 以最近的上一个边缘小方块作为终结, 并从下一个边缘小方块开始进行新的连接. 连接成功时, 如果该边缘点的右边还存在距离小于 D 的边缘小方块, 和以及 `heightAdjust` 和 `lowAdjust` 保持不变. 继续上面的策略尝试接入新的点;

5) 我们称连接所得为线. 对于每条线, 我们用连接时使用的 E_1 、 E_2 特征值作为该线的 E_1 、 E_2 特征. 我们定义线 $L1$, $L2$ 的相似距离为: $|L1.E_1 - L2.E_1| + |L1.E_2 - L2.E_2|$. 如果两条线的相似距离小于阈值 $T3$, 合并成一条; 如果上下相邻的线, 其重合长度大于一定值, 且两条线的相似距离小于 $T3$, 则两条线可以合并成区域.

经过上面的小方块区域连接, 我们得到的是一些由两种灰度值构成且存在垂直边缘的区域, 对于车牌这两种不同的灰度值 E_1 和 E_2 则代表了背景和字体的灰度值.

4.3.3 区域筛选和车牌验证

根据我们对车牌的认识, 我们知道图像中的车牌区域应该具备以下的特征: 1) 车牌的长度和宽度的比例固定; 2) 由于字符原因, 二值化水平投影有一定规律; 3) 车牌字符的大小固定, 字符的宽度大致相等. 根据

水平投影得到字符的宽度, 我们可以比较车牌字符宽度的方差. 根据以上的三个特征进行区域检测, 我们能有效的排除非车牌区域. 定位算法从车牌原图像中抠取出了车牌图像, 并计算出了车牌几何中心的像素坐标是 $center(287,152)$, 车牌定位结果如图 7 所示.



图 7 车牌原图像

4.4 基于 Qt 图像显示

Qt 是一个跨平台的 C++ 图形用户界面应用程序框架. 它提供应用程序开发者创建艺术化的图形用户界面所需的所需功能. Qt 是完全面向对象的, 很容易扩展, 并且允许真正的组件编程.

我们这里使用的 Qt/Embedded 是面向嵌入式系统的 Qt 版本. 它提供了基于 framebuffer 的底层图形引擎. Framebuffer 实在 Linux 内核架构版本 2.2 以后推出的标准的显示设备驱动接口. 本系统并没有直接调用 V4L2 的接口函数, 而是使用 OpenCV 抓取图像, 并创建 QImage 类型对象, 将图像显示出来. 显示图像的系统用户界面如图 8 所示.



图 8 系统用户界面

5 结语

本系统是以 S3C440 为核心平台, 嵌入式 Linux 作为操作系统, 并使用开源的计算机视觉库 OpenCV 来处理图像数据而设计的立体视觉测距系统. 在论文中我们以车牌作为一个目标物, 使用一种基于边缘和灰度值的车牌定位算法正确地定位车牌在图像中的位置, 测得了车牌的距离. 只要改变定位算法, 则能测得任意目标物的距离. 该系统具有体积小、成本低、稳定性高等优点, 可广泛应用于理测绘、工业检测、机器人导航、三维测量等诸多领域.

参考文献

- 1 Hallinan C. 嵌入式 Linux 开发(英文版). 北京: 人民邮电出版社, 2008.1-13.
- 2 陈颐. ARM9 嵌入式技术及 Linux 高级实践教程. 北京: 北京航空航天大学出版社, 2007.1-23.
- 3 Bradski G, Kaehler A. 于仕琪, 刘瑞祯, 译. 学习 OpenCV. 北京: 清华大学出版社, 2009.406-498.
- 4 Zou L, Li Y. A Method of Stereo Vision Matching Based on OpenCV. IEEE ICALIP, 2010. 185-190.
- 5 Liu ZZ. Distance Measurement System Base on Binocular Stereo Vision. IEEE JCAI, 2009. 456-459.
- 6 Zhang ZY. A flexible new technique for camera calibration. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000, 22: 1330-1334.
- 7 陈进. 车牌识别系统关键技术的研究[硕士学位论文]. 合肥: 中国科学技术大学, 2009.5-20.