

动态复杂背景下的智能视频监控系统设计与实现

张纪宽, 彭力, 陈志勇

(江南大学物联网工程学院, 江苏 无锡 214122)

摘要: 采用二重对称帧间差分目标检测算法和基于压缩感知的目标跟踪算法, 设计并实现了一种可适应动态复杂背景下的智能视频监控系統; 基于目标检测该系统能提取本地视频文件中局部运动目标并进行视频压缩, 减少回放、查看视频时间, 可实时播放并处理本地或网络摄像头数据, 也可根据光照变化动态调整二值化阈值, 实现实时区域入侵检测与报警; 基于目标跟踪本系统能在动态背景下对选定目标进行跟踪, 可通过客户端手动控制监控云台跟踪, 也可对入侵目标实现云台自主大角度追踪; 实验表明, 此系统能在日常复杂环境下对运动目标准确检测和大角度跟踪, 在智能家居和移动安防领域有很好的实用性。

关键词: 目标检测; 压缩感知; 自主跟踪; 复杂背景; 智能监控; 监控云台

Design and Realization of Intelligent Video—surveillance System in Dynamic Complex Background

Zhang Jikuan, Peng Li, Chen Zhiyong

(Engineering College of Internet of Things, Jiangnan University, Wuxi 214122, China)

Abstract: In this paper, an intelligent video surveillance system is designed and implemented in dynamic complex background based on the target detection algorithm of double symmetrical frame difference and the target tracking method based on compressive sensing. This system, based on target detection, has functions such as local moving objects extracted from local video file and compression to reduce playback time, local or network camera data real-time playing and processing, real-time area intrusion detection and alarm according to the change of illumination to dynamically adjust the binarization threshold. This system, based on target tracking, also has functions such as tracking the selected target in dynamic background, controlling monitoring PTZ through the client manually, tracking the invasion target independently with large angle. The results showed that the system can accurately detect and track moving target in daily complex environment, have great practical needs in smart home and mobile security field.

Keywords: target detection; compressive sensing; active tracking; complex background; intelligent video surveillance; monitoring PTZ

0 引言

传统视频监控仅采集、显示、存储和回放等简单的功能, 仅仅记录发生的事情, 很难起到预防和报警的作用。随着视频监控应用日益广泛, 智慧城市和公共安全需求的日益增长, 监控视频数据已成为大数据时代的重要数据对象, 采用人工监看视频方式已经远远不能满足需求^[1-3]。由此智能视频监控成为如今研究热点, 文献 [1-2] 概括提出了当今智能监控系统发展的趋势, 大规模的应用尚处于积累阶段, 智能视频监控系统的的需求将日益迫切。文献 [4] 用三帧差分法实现了运动目标的跟踪监控, 为系统的实际应用打下了基础; 文献 [5] 提出了复杂背景下的目标检测与跟踪, 为智能视频监控系統打下了理论基础; 文献 [6] 实现了视频网络传输, 但未对视频进行

智能处理。因此, 文中是在文献 [6] 嵌入式前端无线视频传输服务器基础上, 在 PC 端构建智能视频分析客户端; 优化文献 [7] 中二重对称边缘帧差算法完成运动目标检测, 实现区域入侵报警和本地视频特征提取与视频压缩; 采用简单高效的基于压缩感知的跟踪算法^[8-11], 增强其在复杂动态背景下的性能, 实现摄像头快速连续大角度自动跟踪进入特定区域的目标。完成在日常生活中动态复杂背景下的智能视频监控。

1 系统设计及工作原理

文中设计的智能监控系统由嵌入式前端服务器^[6]和智能视频分析客户端两部分组成。前端服务器采用 TMS320DM365 多媒体处理器, 构建流媒体服务器, 完成视频采集, H.264 编码, RTSP 网络传输, 实时接收客户端控制命令并对监控云台进行全方位控制。客户端运行在 windows 平台上, 采用 MFC 编程, 实现实时视频流接收, 本地视频压缩, 运动目标检测与报警, 监控云台手动、自主跟踪。系统整体结构图如图 1 所示。

系统工作原理: 系统上电后, 前端服务器初始化硬件设备, 打开摄像机, 归位云台并构建 RTSP 服务器等待客户端连接。客户端可通过输入 RTSP 网络地址连接到服务器, 实时接收视频数据, 通过列表框中选择的智能算法来对视频进行运动

收稿日期: 2015-12-28; 修回日期: 2016-02-24。

基金项目: 江苏省产学研联合创新资金——前瞻性联合研究项目 (BY2014023-25)。

作者简介: 张纪宽 (1990-), 男, 硕士研究生, 主要从事网络视频传输方向的研究。

彭力 (1967-), 男, 博士, 教授, 主要从事视觉传感器网络、人工智能、计算机仿真方向的研究。

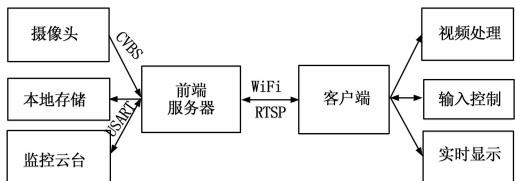


图 1 系统整体结构图

目标检测或跟踪处理, 可通过点击云台控制按钮, 手动向前端服务器发送云台控制命令, 也可选择自主跟踪, 摄像机便自主跟踪入侵区域的运动目标。前端服务器通过 USART 协议控制监控云台转动, 通过无线 WiFi 连入网络与客户端通信。

2 智能算法

智能算法的系统实现必须满足算法简单高效计算量低的特点。针对动态复杂背景时光照等条件易发生突变从而导致目标检测精度低, 背景误判为运动目标的概率大这一难题, 文中系统利用帧间差分法的简单、快速、易实现^[2]等优点, 采用二重对称边缘帧差法实现运动目标检测; 目标跟踪部分采用简单高效的压缩感知跟踪算法, 完成对目标的精确跟踪。

2.1 运动目标检测算法

采用二重对称边缘帧间差分算法实现运动目标检测, 首先根据帧间灰度直方图的巴氏距离判断是否产生光线突变, 若发生突变, 重新建立帧差组合, 保证突变瞬间不影响检测结果, 自主调整二值化阈值, 自适应光照变化后的背景。算法流程图如图 2 所示。

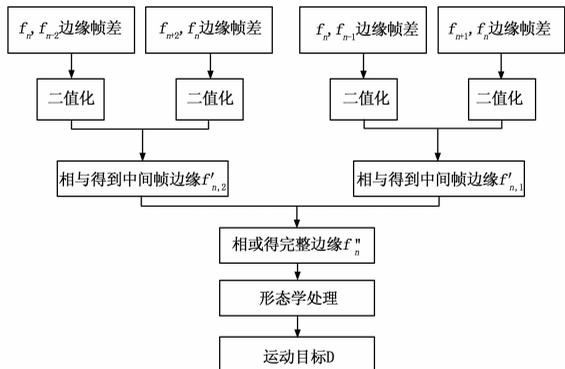


图 2 二重对称帧间差分流程图

二重对称帧间差分步骤如下:

1) 取连续五帧图像 $f_{n-2}, f_{n-1}, f_n, f_{n+1}, f_{n+2}$, 灰度化处理后进行 Canny 算子边缘检测得到边缘图像 $f'_{n-2}, f'_{n-1}, f'_n, f'_{n+1}, f'_{n+2}$ 。

2) 进行五帧边缘差分并二值化处理:

$$\begin{aligned} D(n, n-2) &= |f'_n - f'_{n-2}|, D(n, n-1) = |f'_n - f'_{n-1}| \\ D(n+1, n) &= |f'_{n+1} - f'_n|, D(n+2, n) = |f'_{n+2} - f'_n| \end{aligned} \quad (1)$$

二值化自适应阈值借鉴文献 [7] 中的 OUT 法, 增加一个光照变化量 ΔT , 变为动态阈值:

$$\begin{aligned} T' &= TH + \Delta T \\ \Delta T &= \frac{\theta}{S} \sum_{i=1}^{N-1} \sum_{j=1}^{M-1} |f_n(x, y) - f_{n-1}(x, y)| \end{aligned} \quad (2)$$

其中: θ 为系数, 参考值为 0.1, S 为一帧图像像素个数。

无光照突变时 ΔT 趋近 0, 阈值不变; 当产生光照突变时, ΔT 增大, 阈值对应增大, 从而抑制复杂动态背景时因光照突变带来的二值化影响。

3) 对差分结果进行运算处理, 如公式 (3), 得到完整边缘图像 f''_n 。

$$\begin{aligned} f'_{n,2} &= D_{(n,n-2)} \wedge D_{(n+2,n)} \\ f'_{n,1} &= D_{(n,n-1)} \wedge D_{(n+1,n)} \\ f''_n &= f'_{n,2} \vee f'_{n,1} \end{aligned} \quad (3)$$

4) 进行形态学处理, 得到运动目标 D 。

检测光照突变方法: 根据光照突变时直方图灰度级像素个数相应的发生突变, 因此, 可通过巴氏系数 d 来衡量相邻帧灰度直方图的相似性, 从而判断是否发生光照突变, 如公式 (4):

$$d(H_1, H_2) = \sqrt{1 - \frac{\sum_i \sqrt{H_1(i)} \times H_2(i)}{\sum_i H_1(i) \times \sum_i H_2(i)}} \quad i = 0, 1, \dots, 255 \quad (4)$$

其中: H 为图像的直方图, 图像相似度越高, d 越大, 接近 1, 反之越小, 接近 0, 通过实验, 设置为 0.7 时可很好的判定是否发生光照突变。

2.2 目标跟踪算法

文中系统使用基于压缩感知的目标跟踪算法, 分为特征提取、特征向量分类、分类器更新三部分。其中特征提取选用压缩感知提取, 分类使用的朴素贝叶斯分类器, 更新通过对每一帧图像的在线学习, 从而实现目标跟踪的全过程。算法流程图如图 3 所示。

第 t 帧时, 对此帧图像提取采样获得若干图图片, 其中目标的样片作为正样本, 背景的为负样本, 经过多尺度变换得到正负样本的多尺度图像特征, 利用稀疏测量矩阵对其进行降维得到各自的压缩向量, 最后通过各自的压缩向量训练朴素贝叶斯分类器。第 $t+1$ 帧时, 在第 t 帧跟踪目标周围采样若干个窗口, 进行多尺度变换、降维、提取特征和分类, 降维和分类时分别采用跟第 t 帧相同的测量矩阵和分类器, 最终的跟踪窗口便是分类中分数最大的窗口, 从而完成了第 t 帧到第 $t+1$ 帧的跟踪。

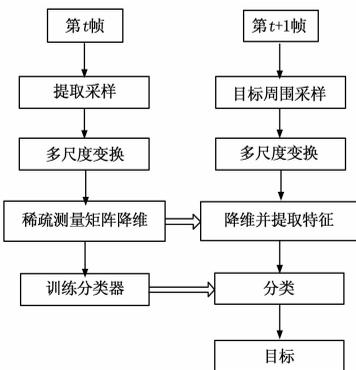


图 3 目标跟踪算法流程图

算法实现步骤如下:

1) 输入: 分别输入视频序列 v , 和正、负样本的采样半径 α 和 β 。

2) 提取样片: 以第 $t-1$ 帧的目标窗口中心 o 为圆心, 分别在以 α 为半径的圆形区域和以 β 为外半径 α 为内半径的圆环区域采集正、负样本。

3) 多尺度变换: 用公式 (5) 所示滤波器对图像某特定位置进行灰度卷积, 其中 i, j 分别是矩形滤波器的宽和高。

$$h_{i,j}(x,y) = \begin{cases} 1, 1 \leq x \leq i, 1 \leq y \leq j \\ 0, otherwise \end{cases} \quad (5)$$

4) 随机测量矩阵选取: 随机测量矩阵的选取必须足够稀疏且符合 RIP 条件才能将数据恢复, 参考文献 [8] 选取的随机矩阵 \mathbf{R} , 其中 \mathbf{R} 中的每一个元素计算公式如公式 (6), 式中 $s = \frac{(\omega \times h)^2}{4}$, r_{ij} 符合标准正太分布。

$$r_{ij} = \sqrt{s} \times \begin{cases} 1, p = \frac{1}{(2s)} \\ 0, p = 1 - \frac{1}{s} \\ -1, p = \frac{1}{(2s)} \end{cases} \quad (6)$$

5) 随机投影降维: 如公式 7, 一个 m 维的图像空间 x 可以被一个 $n \times m$ 的稀疏矩阵变换成 n 维, 变换过程如图 4 所示, \mathbf{R} 中灰白黑三种颜色分别代表矩阵中的正数、零、负数, 带箭头的蓝色曲线表示用 \mathbf{R} 矩阵中的非零元素感知 x 图像空间中的元素。

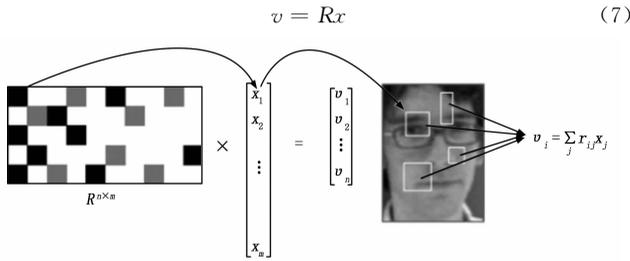


图 4 随机投影降维图解

6) 分类器构建: 对每个样本 $z \in R^{\omega \times h}$, 低维表示是 $v = (v_1, \dots, v_n)^T \in R^n$, 构建朴素贝叶斯分类器:

$$H(v) = \log \left(\frac{\prod_{i=1}^n p(v_i | y=1) p(y=1)}{\prod_{i=1}^n p(v_i | y=0) p(y=0)} \right) = \sum_{i=1}^n \log \left(\frac{p(v_i | y=1)}{p(v_i | y=0)} \right) \quad (8)$$

其中样本标签为 $y \in \{0, 1\}$, 0 代表负样本, 1 代表正样本, 假设特征中每个元素是独立的, 因此可通过公式 8 来建模。假定 $p(v_i | y=1)$ 和 $p(v_i | y=0)$ 也服从高斯分布, 则有:

$$\begin{cases} p(v_i | y=1) \sim N(\mu_i^1, \sigma_i^1) \\ p(v_i | y=0) \sim N(\mu_i^0, \sigma_i^0) \end{cases} \quad (9)$$

7) 分类器更新: 对公式 9 中的参数 $\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0$ 进行增量更新如公式 (10) 所示:

$$\begin{cases} \mu_i^1 \leftarrow \lambda \mu_i^1 + (1-\lambda) \mu^1 \\ \sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \end{cases} \quad (10)$$

$$\mu^1 = \frac{1}{n} \sum_{k=0|y=1}^{n-1} v_i(k), \sigma^1 = \sqrt{\frac{1}{n} \sum_{k=0|y=1}^{n-1} (v_i(k) - \mu^1)^2}$$

学习因子 $\lambda > 0$, 按照同样的方式更新 (μ_i^0, σ_i^0) 。

3 监控系统软件设计

前端服务器参考文献 [6], 文中主要工作是客户端软件设计, 实时接收服务器端视频信息并进行运动目标检测、跟踪, 本地视频压缩处理, 云台控制及云台自动跟踪。在 PC 端 MFC 平台上运用 C++ 多线程编程, 采用 live555 流媒体库获取视频流, FFMPEG 解码库实现实时 H.264 视频流解码, Opencv 实现目标检测和视频压缩, 单独的线程实现目标跟踪, 最后主线程变为控制线程。客户端功能模块图和各线程协同工作流程图分别如图 5 和图 6 所示。

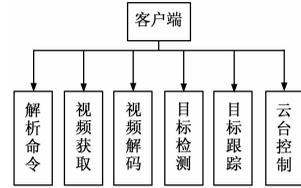


图 5 客户端功能模块图

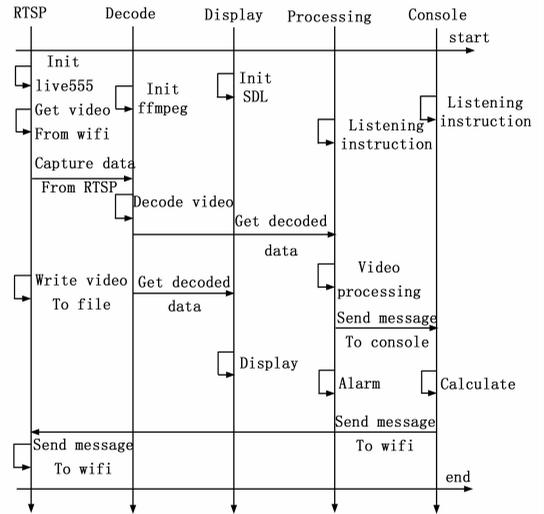


图 6 线程工作流程图

动态背景控制策略: 文中系统可在动态背景下实现目标检测与跟踪, 通过线程互斥机制使视频处理线程和云台控制线程互斥运行, 有效解决帧差法无法适应动态背景的问题和云台转动产生图像漂移对压缩感知的影响。线程互斥流程图如图 7 所示。

在进行视频处理前, 需要先访问信号量, 通过 P 操作申请信号量, 若此时信号量为 0 则线程阻塞, 说明此时云台正在转动, 等待云台控制线程的 V 操作释放信号量; 若信号量为 1, P 操作先将信号量置 0 后再进行视频处理操作。

4 监控系统测试

文中实现的智能监控系统客户端运行在 Intel (R) Pentium (R) CPU 主频 3.00 GHz 的 PC 机上, 采用 C++ 语言在 VS2010 上进行开发, 系统用户界面及区域入侵和云台自主跟踪效果如图 8 所示。上面三幅是区域入侵效果图, 下面三幅是云台自主跟踪效果图。

运动目标检测算法的测试采用自采视频, 分别利用传统帧

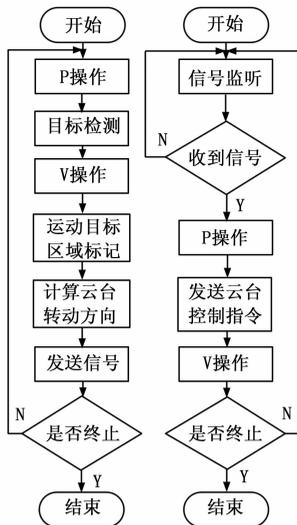


图 7 线程互斥运行流程图



图 8 智能监控效果图

间差分和高斯背景差分法与本文实现算法对有光照突变的视频进行测试, 如图 9。a1 代表光照突变前一帧, 可见在无光照突变时本文使用算法 b1 可以完整检测出目标轮廓, 传统三帧差分 c1 则检测不够连续, 传统高斯背景建模法 d1 则检测结果比较充实。在发生光照突变后, c2 和 d2 则无法检测目标, 由于本文使用算法在光照突变时重构帧差组合且动态调整二值化阈值, 从而仍能清晰检测出目标轮廓。

目标跟踪采用三组标准视频 Occluded face2、David indoor、Tiger1 将本文实现算法与 TLD^[12] 和 MIL^[13] 算法进行对比, 如图 10。

采用跟踪成功率 (SR) 和算法处理时间对目标跟踪算法进行性能测试。定义 $score = \frac{R_T \cap R_C}{R_T \cup R_C}$ 为跟踪每一帧图像得分, R_T 为实际跟踪区域, R_C 为真实区域, 若 score 大于 0.5 代表跟踪结果正确, 3 种算法跟踪成功率如表 1 所示, 平均处理速度如表 2 所示。

表 1 跟踪成功率 (%)

Sequence	Proposed	MIL	TLD
Occluded face2	100	99	48
David indoor	91	70	95
Tiger2	60	47	46

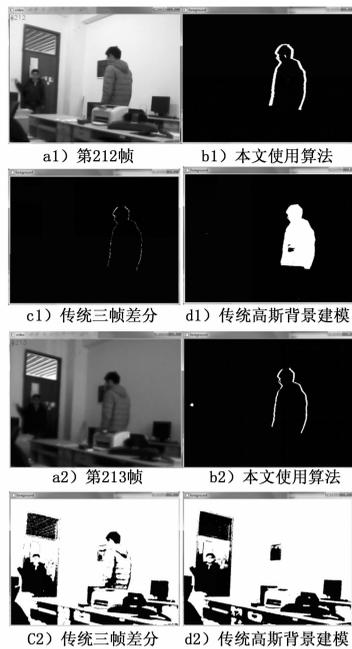


图 9 光照突变时目标检测结果



图 10 目标跟踪结果比较

表 2 平均处理速度(帧/秒)对比

FPS	Proposed	MIL	TLD
Average FPS	34.9	10.47	10.01

由以上比较结果, David indoor 视频序列, TLD 算法表现最好, 略好于本文使用算法; occluded face2 序列, TLD 算法出现丢失目标现象严重, 本文使用算法跟踪效果最好, MIL 算法次之; Tiger2 序列目标不明显且变换复杂, 文中使用算法依然保持最好的跟踪成功率。从处理速度上, 文中使用算法明显高效于其他算法。因此文中算法高效, 实时, 符合系统实现要求。

5 结束语

设计并实现了一套适应动态复杂背景的智能视频监控系统。该系统能播放处理本地视频, 对视频文件进行区域压缩,

接收并处理网络实时视频流, 实现实时区域目标检测和对选定目标进行跟踪, 可手动控制监控云台实现大角度跟踪, 也可对入侵目标进行云台自主跟踪。与文献 [6] 结合组成完整的监控系统前后端, 适用于众多监控领域。

参考文献:

[1] 黄凯奇, 陈晓棠, 康运锋, 等. 智能视频监控技术综述 [J]. 计算机学报, 2015, 38 (6): 1093-1118.
 [2] 黄铁军, 郑锦, 李波等. 多媒体技术研究: 2013——面向智能视频监控的视觉感知与处理 [J]. 中国图象图形学报, 2014, 19 (11): 1539-1562.
 [3] 管菊花, 郭波. 基于 DaVinci 技术的嵌入式智能监控系统 [J]. 计算机测量与控制, 2014, 22 (9): 2754-2757.
 [4] 谢永昌, 许少凡, 狄红卫. 基于 DM642 的运动目标实时跟踪监控系统的设计 [J]. 光学技术, 2010, 36 (3): 469-473.
 [5] 李劲菊, 朱青, 王耀南. 一种复杂背景下运动目标检测与跟踪方法 [J]. 仪器仪表学报, 2010, 31 (10): 2242-2247.
 [6] 张纪宽, 彭力, 陈志勇. 基于嵌入式前端服务器的音视频监控系统设计 [J]. 江南大学学报 (自然科学版), 2015, 14 (1): 38-42.
 [7] 王丽梅. 基于 DM642 的智能视频监控系统的设计与实现 [D]. 北

京: 北京工业大学, 2013.
 [8] Zhang K, Zhang L, Yang M H. Real-time compressive tracking [A]. Computer Vision - EVCC 2012 [C]. Berlin Heidelberg: Springer, 2012, 7574: 864-877.
 [9] Wu Y, Jia N, Sun J. Real-time multi-scale tracking based on compressive sensing [J]. The Visual Computer, 2014, 31 (4): 471-484.
 [10] Zhu Q, Yan J, Deng D. Compressive tracking via oversaturated sub-region classifiers [J]. Computer Vision, IET, 2013, 7 (6): 448-455.
 [11] Jing J, Xu G, Lei B, et al. Using appearance re-matching to improve real-time compressive tracking [A]. Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service [C]. ACM, 2013: 28-33.
 [12] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection [J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2012, 34 (7): 1409-1422.
 [13] Babenko B, Yang M H, Belongie S. Visual tracking with online multiple instance learning [C] Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE, 2009: 983-990.

(上接第 99 页)

式, 触摸屏经过开机画面自动进入工作界面, 如图 6 所示。当钥匙开关打到学习模式, 界面显示各种传感器, 气缸磁性开关的状态如图 7 所示。按下运行按钮, 设备配合冲床开始旋转冲孔。为测试系统的故障诊断功能, 特意不将管料退出, 此时警报声响起, 固定端出料口上方接近开关指示灯亮, 界面弹出故障提示框, 如图 8 和图 9 所示, 按下确定按钮, 并按下运行/停止按钮, 设备重新开始工作。系统采用了跨平台的 C++ 图形用户程序 Qt 作为框架, 提供了应用程序开发人员艺术性的图形用户界面, 通过不断的设备调试, 建立了故障维护数据库, 建立起快速的故障响应机制, 一旦故障发生, 能在短时间内给一线工人提供建设性的意见。



按钮 1: 工件冲孔次数的设置 按钮 2: 退回距离的标定
按钮 3: 加工数量的清零 按钮 4: 当前设备运行状况的显示

图 6 工作界面



图 7 学习界面

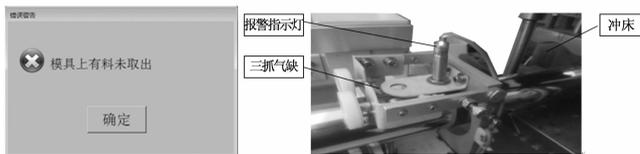


图 8 故障提示框

图 9 故障报警

6 结束语

在冲压自动化生产线领域, 本方案提供了一套智能、高效、稳定的生产系统。其中, ARM 用来实现控制系统的数据存储和传送, CPLD 用来控制上料机机械手的一系列动作, 该控制系统还能实时记录下设备的产量, 大大减少了生产管理部门的工作量, 友好的人机交互界面使得操作更加直观方面。同时, 利用 Linux 系统实现了体积小, 效率高, 可靠性强的应用操作, 具有的智能故障诊断功能减少了不必要的设备停机时间, 提高了企业生产效率。该设计方案已经投入生产并取得了较好的效果, 具有很好的应用前景。

参考文献:

[1] 庄宇昀, 徐辉. 冲压线的发展 [J]. 一重技术, 2005 (4): 20-22.
 [2] 李勇. 新一代冲压技术 [J]. 新技术新工艺, 2003 (4): 31-32.
 [3] 张海波, 陈涛, 王建立, 等. 基于 ARM+Linux 的直流伺服控制系统设计 [J]. 电子技术应用, 2012, 08: 33-36.
 [4] 徐伟业, 江冰, 虞湘宾. CPLD/FPGA 的发展与应用之比较 [J]. 现代电子技术, 2007, 02: 4-7.
 [5] 赵明鑫. Linux 内核在 ARM 上的裁减与移植 [J]. 计算机与数字工程, 2009, 07: 81-84.
 [6] 朱大奇, 于盛林. 基于故障树最小割集的故障诊断方法研究 [J]. 数据采集与处理, 2002, 03: 341-344.
 [7] 郭晓静, 宋胜博. APU 故障智能诊断系统设计 [J]. 计算机测量与控制, 2015, 11: 3585-3588.