|  |  |
| --- | --- |
| **分类号 密级** | 校徽 |
| **UDC** |
|  |

本 科 毕 业 论 文

基于CAMSHIFT的自动瞄准系统

设计与开发

**学生姓名** 李智勤 **学号** 13090041031

**指导教师** 黎明

**院、系、中心** 工程学院自动化及测控系

**专业年级** 2013级自动化

**论文答辩日期** 2017年 6月 日

**中 国 海 洋 大 学**

基于CAMSHIFT的自动瞄准系统设计与开发

完成日期：

指导教师签字：

答辩小组成员签字：

基于CAMSHIFT的自动瞄准系统设计与开发

摘 要

|  |
| --- |
| 本论文提出了一种基于机器视觉的自动瞄准系统设计。该系统通过摄像头采集数字图像，使用CAMSHIFT、密度匹配、面积匹配、重叠匹配算法实现目标的识别与跟踪，进而控制由57步进电机搭建的二自由度云台实现自动瞄准，并能通过Android端应用实现远程实时视频传输、自动控制和手动遥控的无扰切换等功能。该系统对遮挡、噪声干扰有较好的容忍性，结合密度匹配、面积匹配、重叠匹配算法能够有效识别目标丢失，并在目标重新进入视野后继续自动瞄准。本论文详细论述了该系统核心图像处理算法的实现，介绍了基于EmguCV的PC端上位机的设计，介绍了基于STM32F407ZGT6设计的控制器的软硬件实现，介绍了Android端应用的实现，并验证了系统的性能。  关键词：自动瞄准；数字图像处理；EmguCV；CAMSHIFT； |

**Design and Development of Automatic Aiming System Based on CAMSHIFT**

**Abstract**

|  |
| --- |
| This thesis presents a design of automatic aiming system based on machine vision. This system acquires digital images through the camera, uses CAMSHIFT, density matching, area matching algorithm to identify and track the target, and then control the two-degree PTZ to achieve automatic aiming, and through an Android application to achieve remote real-time video transmission, automatic control and manual remote control and other functions. This system has robust noise and occ-lusion tolerance, and it is effective to identify the missing target by combining the density matching and the area matching algorithm, and then continue to aim when the target re-entering the field of view. In this thesis, the realization of the core image processing algorithm is discussed in detail. The design of PC software based on EmguCV is introduced. The hardware and software implementation of controller based on STM32F407ZGT6 is introduced, and the application of Android is introduced,then the performance of the system is verified.  Keywords: Automatic aiming; Digital image processing; EmguCV; CAMSHIFT |

目 录

摘 要 I

Abstract II

目 录 III

1 绪论 1

**1.1** 自动瞄准系统的意义及作用 1

**1.2** 基于视觉的目标识别与追踪算法概述 1

**1.3** 本课题的研究目的及意义 2

**1.4** 论文的主要工作内容 2

2 总体方案设计 3

**2.1** 系统技术指标 3

**2.2** 核心识别追踪算法选择 3

**2.3** 系统总体设计方案 4

**2.4** 系统硬件开发平台选择 4

**2.5** 系统软件开发平台选择 5

**2.5.1** PC端软件开发平台选择 5

**2.5.2** 嵌入式软件开发平台选择 5

**2.5.3** Android端软件开发平台选择 5

**2.6** 本章小结 5

3 系统硬件设计 6

**3.1** 步进电机及其驱动器选型 6

**3.2** 电源选型 6

**3.3** 图像传感器选型 7

**3.4** 嵌入式系统电路设计 7

**3.4.1** 嵌入式系统电源电路设计 7

**3.4.2** 系统基础电路设计 8

**3.4.3** MCU引脚分配 8

**3.4.4** 嵌入式系统外围接口电路设计 9

**3.5** 本章小结 10

4 系统软件设计 11

**4.1** PC端软件设计 11

**4.1.1** PC端软件总体设计 11

**4.1.2** PC端软件UI设计 12

**4.1.3** 核心图像处理算法实现 12

**4.1.4** PC-嵌入式系统通信协议 16

**4.1.5** Socket通信及视频传输实现 17

**4.2** Android端软件设计 18

**4.2.1** Android端软件总体设计 18

**4.2.2** Android端软件UI设计 18

**4.2.3** Android端Socket通信实现 19

**4.3** 嵌入式系统软件设计 20

**4.4** 本章小结 20

5 系统性能评估 21

**5.1** 帧率测试 21

**5.2** 瞄准精度评估 21

**5.2.1** 静态目标瞄准精度评估 21

**5.2.2** 动态目标瞄准精度评估 22

**5.3** 系统资源占用情况分析 24

**5.4** 本章小结 24

6 总结与展望 25

参考文献 26

致谢 27

附录 28

# 绪论

## 自动瞄准系统的意义及作用

随着信息技术的高速发展，军队信息化成为各国的军事技术的重要发展方向。自动瞄准系统作为信息技术军事应用的典型，有着人工瞄准无法达到的准确性、安全性，并且能够有效减少友军人员伤亡。此外，自动瞄准系统能够长时间稳定工作，无论是进攻还是防守，都能发挥出其巨大优势。

近年来，数字图像传感器的性能迅速提升，将数字图像传感器用于辅助、自动瞄准成为了一个重要的研究方向。相对于使用其他传感器的自动瞄准系统，使用图像传感器的系统获得的信息更多，使用范围更广。在不同的环境场合，只需更换特定的数字图像传感器即可。

## 基于视觉的目标识别与追踪算法概述

一个基于视觉的自动瞄准系统，其核心技术之一就是基于视觉的目标识别与追踪。国内外已经提出了大量可以用于目标识别与追踪的算法，广泛使用的包括:

1. 基于形状特征的识别。如SURF[1]、Hu不变矩匹配。
2. 基于颜色特征的识别。如颜色直方图匹配、反向投影法、CAMSHIFT[2]。
3. 基于目标运动的识别。如背景减法。
4. 基于机器学习的识别。如级联分类器[3]。
5. 模板匹配。

本设计中使用到的CAMSHIFT(Continuously Adaptive Mean-Shift)算法是在1998年提出的一种改进的Mean-Shift算法，通过迭代求矩的方式获得被跟踪目标的重心、长短轴等信息进行跟踪，当目标大小发生改变或者旋转时能自动适应并做出调整。

CAMSHIFT算法提出至今已有19年，在此期间有许多应用该算法的设计论文发表，同时也有大量结合kalman滤波等其他方法的改进算法被提出[4]。CAMSHIFT及其各种改良算法广泛应用于手势识别、目标跟踪等场合。相比较于其他基于Haar特征、Hu不变矩等特征匹配算法有明显的速度优势，更适合于视频图像的实时处理中。但其算法存在一定缺陷，当目标色调和背景色接近时追踪效果不佳；当目标发生丢失时，该算法不能正确识别并继续跟踪。

随着半导体制造工艺的快速发展，各种处理器的性能增长迅速，过去看来较为复杂的算法也有很大的实用价值。而一些相对快速的识别算法也在不断改进中提升识别效果，并在经济型应用领域得到大量应用。

## 本课题的研究目的及意义

目前国内外的军用自动瞄准系统开发都还在探索阶段，虽有少量研究成果，但因为价格过高或不够成熟，尚未有大规模列装的例子。

本课题旨在设计一个由上位机、二自由度云台、摄像头、移动端app组成的基于图像识别的自动瞄准系统，该系统应能通过摄像头识别出用户指定的一个与环境有不同色调的目标，并能通过控制一个二自由度的云台使得摄像头能够跟随目标运动，并评价瞄准的性能。用户需要时可实现自动控制和手动控制的无扰切换。系统采集处理的图像应能实时传输到移动端app上。

针对CAMSHIFT无法识别目标丢失的问题，本设计通过结合密度匹配、面积匹配、重叠匹配算法，使得目标丢失能够被有效识别，并能在目标重新进入视野时自动继续瞄准而无需人工干预，改善了系统的瞄准性能。

除此之外，由于该设计的核心识别追踪算法有较好的适应性，其潜在的应用场合包括但不限于：

1. 足球、篮球比赛球员跑动距离、热区图自动生成。
2. 无人机的目标自动跟踪。

## 论文的主要工作内容

本课题的主要工作内容包括：

1. 图像处理算法的设计与实现；
2. PC端图像处理上位机的编写；
3. 基于STM32F407的电机控制器的软硬件设计；
4. Android端APP的编写；
5. 二自由度云台的搭建以及系统整机调试；
6. 系统性能评估。

论文具体内容的安排如下：

第一章 绪论：阐述了自动瞄准系统的意义和应用价值，简单介绍了目前应用较为广泛的基于视觉的目标识别与追踪算法，介绍了本课题的研究目的及应用价值。

第二章 总体方案设计：对系统应达到的性能指标做出了说明，对核心识别追踪算法的选择做出了解释与说明，介绍了基于CAMSHIFT的自动瞄准系统的总体设计方案和结构框图。

第三章 系统硬件设计：

# 总体方案设计

## 系统技术指标

由于透视现象，图像传感器得到的图像中的目标随距离增加而变小，自动瞄准系统的精度也随之降低。为了便于观测和评估，本课题的瞄准精度使用瞄准稳定后瞄准中心与目标正对采集系统的横切面重心的像素点绝对差值来评估，其定义为：



式中 ,——采集图像中心的坐标；

,——目标中心在采集图像中的坐标。

由于瞄准无误差时，采集图像中心和目标中心重合，因此越小，精度越大，在最大理想精度时像素点绝对差值为零。本课题设计应达到的基本参数如表 2‑1所示。

表 2‑1 系统设计基本指标

|  |  |
| --- | --- |
| 采集帧率 | ≥25 fps |
| 光照条件 | 100~150lux |
| 运动目标 | ≤60 |
| 静态目标 | ≤10 |
| 瞄准范围 | 0°≤X≤180°，0°≤Y≤180° |
| 有效瞄准距离 | 0.5m~2m |
| 目标半径大小 | ≥10cm |
| 目标最大移动速度 |  |

## 核心识别追踪算法选择

针对绪论中提到的几种目前应用较为广泛的算法，编写测试程序分析发现：

1. SURF算法能够检测到大量特征点，当目标特征点和源图像中吻合较多时效果很好，但对于实时连续图像采集系统来说耗时过长。
2. 背景减法在背景环境较为稳定时效果良好，耗时短，资源占用低，但考虑到本设计的采集摄像头在工作中会发生移动，背景会有较大变化，该方法不适用于本设计中。
3. 基于Haar特征的级联分类器训练要花费大量时间，并且由于本设计中的目标有不确定性，该方法亦不适用于本设计中。
4. CAMSHIFT算法在目标色调与背景相差较大时效果较为理想，其耗时较短可以用于实时连续图像采集系统中，但不能分辨出目标是否丢失。

考虑到使用基于颜色特征的识别时间、空间复杂度较低，能够有效地保证系统的实时性，本设计将CAMSHIFT作为核心算法。针对CAMSHIFT算法存在的目标丢失无法识别、物体移动速度过快时效果不好的问题，本设计通过结合轮廓识别和密度、面积匹配、重叠匹配算法来改善其跟踪性能。

## 系统总体设计方案

由基本设计指标可知，系统对图像采集、处理帧率要求较高，因此设计将需要消耗大量运算资源的图像处理算法放在PC端， PC较为出色的运算性能可以保证系统处理的实时性。另外由于瞄准精度指标要求，系统对二自由度平台的电机精度要求较高，因此搭建平台的电机选用步进电机，通过嵌入式系统和步进电机驱动器控制电机转动。在实现目标的识别与定位后，PC将其位置信息反馈到嵌入式系统实现闭环控制。

结合性能指标要求，综合考虑成本、性能、开发难度，本课题设计的系统框图如图 2‑1所示。



图 2‑1 系统总体框图

## 系统硬件开发平台选择

本系统涉及到的硬件开发主要是嵌入式系统硬件开发，嵌入式系统的性能很大程度上取决于系统选用的微处理器。本设计使用的MCU为意法半导体公司的STM32F407ZGT6。STM32F4采用ARM Cortex-M4内核,内嵌硬件FPU单元,主频168MHz[6]，大量集成的片内外设能够有效减少开发成本和开发周期。

## 系统软件开发平台选择

由于本课题所需软件较多且运行环境跨多平台，选择合适的开发平台将会大大减少开发周期和开发难度。下面将对各平台软件开发选择做具体的介绍。

### PC端软件开发平台选择

考虑到本设计PC端软件需要大量使用图像处理算法和多线程、多窗口编程，开发语言选择C#/C++混合编程，结合开源计算机视觉库OpenCV、EmguCV能够有效减少开发周期。因此本设计使用Visual Studio 2015搭建的WinForm工程，工程框架基于.NET Framework 4.5.2，并添加了OpenCV、EmguCV为外部引用。

### 嵌入式软件开发平台选择

为了降低开发难度，本设计的嵌入式平台开发使用意法半导体公司提供的官方HAL库，版本V1.7.0，另使用FreeRTOS[5]作为实时操作系统以充分利用STM32F407的性能。IDE使用Keil uVision5，编译器版本V5.06 update 4，使用C99标准。

### Android端软件开发平台选择

Android端软件开发全部使用Java语言，IDE选择Android Studio 2.2.3， API级别23。为了测试应用的兼容性，除在一屏幕分辨率为1920\*1600的使用Android 7.0系统的实机上调试以外，还使用了Genymotion模拟器测试了Android 6.0系统下1920\*1600的屏幕分辨率下的应用使用情况。

## 本章小结

本章主要介绍系统总体方案的设计过程，介绍了系统的基本设计指标，继而根据该指标并综合考虑开发成本设计了总体方案，最后对系统的软硬件开发平台选择做出了说明。

# 系统硬件设计

性能良好的硬件是系统正常运行的基础。本章将在前文的基础上详细介绍本系统的硬件设计。

## 步进电机及其驱动器选型

本设计搭建云台的步进电机选用57步进电机，配合可进行32细分的TB6600步进电机驱动器，电机每转需6400个脉冲，能有效提高控制精度并减少步进电机的低频振动和超调量。使用两个步进电机可搭建一个二自由度的云台。嵌入式控制系统通过USB与PC通讯后收集目标位置信息，进而计算输出量输出脉冲控制信号和方向控制信号至步进电机驱动器，控制电机转动。其中使用的云台和步进电机驱动器如图 3‑1所示。

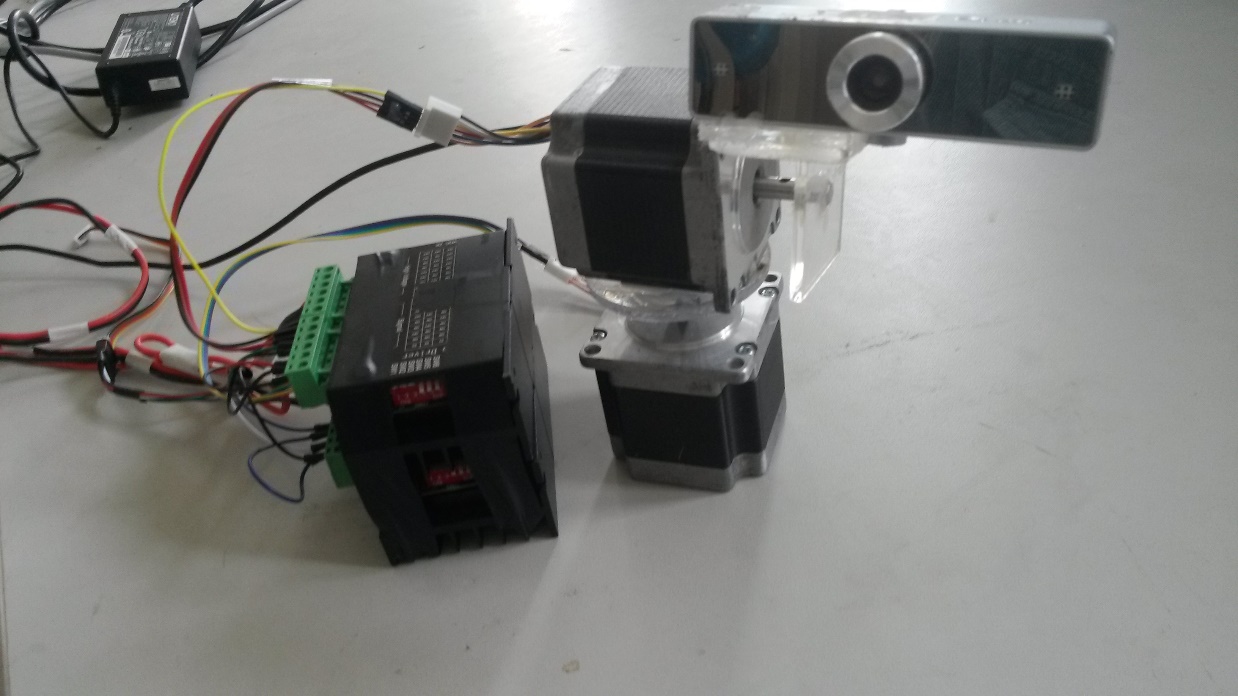


图 3‑1 云台和步进电机驱动器

## 电源选型

本设计硬件部分使用到的电源有两种，分别是：

1. 步进电机、步进电机驱动器电源；
2. 嵌入式系统电源。

其中步进电机电源由步进电机驱动器给出，步进电机驱动器的供电电压范围为10V~42V，因此本系统的步进电机、步进电机驱动器电源选用了一款如图 3‑2所示的开关电源，其额定输出电压为24V，额定输出电流为10A。



图 3‑2 步进电机驱动器开关电源

嵌入式系统电源选用市面上常见的DC稳压电源，其额定输出电压9V，额定输出电流1A。

## 图像传感器选型

基于成本和性能指标的考虑，本设计选用了一款使用USB接口的OV7725摄像头，最大采集图像尺寸为640\*480。经测试，该摄像头在曝光度设置为-6时，采集速率可以稳定达到到33fps，能够有效减少因目标移动速度过快导致的采集图像的拖影和形变。

## 嵌入式系统电路设计

基于STM32F407的嵌入式控制系统是本设计的核心组成之一。下文将详细介绍该嵌入式控制系统的电路设计。

### 嵌入式系统电源电路设计

STM32F407ZGT6的工作电压为3.3V，因此9V稳压源不能直接给MCU供电，本设计选用了集成的DC-DC稳压芯片LM1117-5V和LM1117-3.3V分别得到5V和3.3V的电压，为MCU及嵌入式系统板的其他器件供电，电路原理图如图 3‑3所示。



图 3‑3 嵌入式系统电源电路

### 系统基础电路设计

除电源电路外，系统基础电路还包括BOOT电路、时钟电路、复位电路、JTAG调试接口等。其中BOOT电路和时钟电路如图 3‑4所示，复位电路和JTAG调试接口电路如图 3‑5所示。



图 3‑4 BOOT电路和时钟电路



图 3‑5 复位电路和JTAG接口

### MCU引脚分配

本设计使用到的MCU引脚较多，除最小系统所需的晶振引脚和JTAG调试引脚外，还用到了数个GPIO和定时器输出引脚。MCU引脚分配如表 3‑1所示。

表 3‑1 STM32F407引脚分配

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position | Name | Type | Signal | Label |
| 1 | PE2 | Output | GPIO\_Output | LED1 |
| 2 | PE3 | Output | GPIO\_Output | LED2 |
| 3 | PE4 | Output | GPIO\_Output | LED3 |
| 4 | PE5 | Output | GPIO\_Output | LED4 |
| 23 | PH0-OSC\_IN | I/O | RCC\_OSC\_IN |  |
| 24 | PH1-OSC\_OUT | I/O | RCC\_OSC\_OUT |  |
| 46 | PB0 | Output | GPIO\_Output | DIRX |
| 47 | PB1 | Output | GPIO\_Output | DIRY |
| 60 | PE9 | I/O | TIM1\_CH1 |  |
| 96 | PC6 | I/O | TIM8\_CH1 |  |
| 103 | PA11 | I/O | USB\_OTG\_FS\_DM |  |
| 104 | PA12 | I/O | USB\_OTG\_FS\_DP |  |
| 105 | PA13 | I/O | SYS\_JTMS-SWDIO |  |
| 109 | PA14 | I/O | SYS\_JTCK-SWCLK |  |
| 110 | PA15 | I/O | SYS\_JTDI |  |
| 114 | PD0 | I/O | GPIO\_EXTI0 | KEY1 |
| 115 | PD1 | I/O | GPIO\_EXTI1 | KEY2 |
| 116 | PD2 | I/O | GPIO\_EXTI2 | KEY3 |
| 117 | PD3 | I/O | GPIO\_EXTI3 | KEY4 |
| 133 | PB3 | I/O | SYS\_JTDO-SWO |  |
| 134 | PB4 | I/O | SYS\_JTRST |  |

其中定时器1和定时器8的输出通道1输出脉冲控制信号，DIRX、DIRY输出方向控制信号，另外为了方便调试，外接了4个按键和4个LED。

### 嵌入式系统外围接口电路设计

本系统主要使用到的接口电路包括USB接口电路和电机驱动器控制信号输出接口电路。系统中用到的USB为全速USB，应添加电阻进行阻抗匹配。不进行阻抗匹配时，USB传输速率可能受到严重影响，甚至可能无法工作。在完全匹配的条件下，信号源的输出能量将会被终端匹配电阻完全吸收，不产生反射[7]。

由于电机驱动器识别的信号为TTL电平，高电平为+5V，而MCU控制输出信号电平为3.3V的CMOS电平，为了使控制输出信号被有效识别，选用74HCT245D进行3.3V-5V的电平转换，同时隔开步进电机驱动器和嵌入式系统电路，对嵌入式系统电路起到保护作用。另外，74HCT245D输入端应接下拉电阻，保证输入端电平的确定性。其设计原理图如图 3‑6所示。

图 3‑6 接口电路

## 本章小结

本章主要介绍了系统的硬件设计，对步进电机和其驱动器选型作出了说明，具体介绍了电源和嵌入式系统电路的设计。

# 系统软件设计

软件设计是本课题的核心工作内容之一，本章将在第二章的基础上详细介绍各平台的软件设计。

## PC端软件设计

### PC端软件总体设计

本设计的PC端软件是组成统的最重要的核心，主要承担的任务有：

1. 用户交互；
2. 读取摄像头采集数据；
3. 识别并追踪目标；
4. 与嵌入式系统信息交换；
5. 建立Socket服务器，与Android端通信。

由上可知，其承担任务较为复杂，如果只使用单线程操作，除了无法发挥处理器的多线程处理能力，还有可能因线程阻塞导致UI线程响应缓慢甚至无响应，因此PC端软件设计为多线程程序。除去外部引用创建的线程，程序运行时可能会产生8个线程同时运行，其代码图如图 4‑1所示。

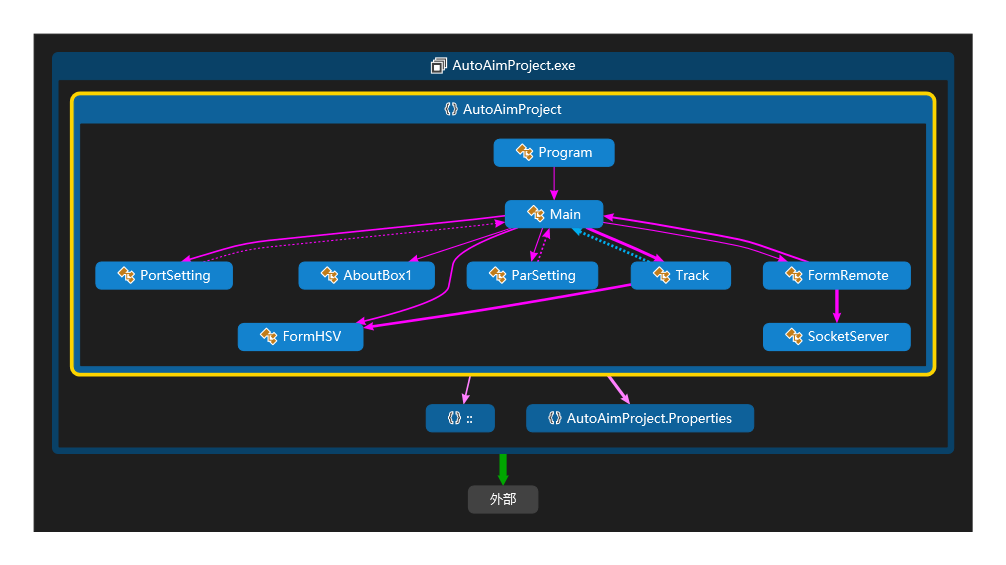


图 4‑1 PC端软件代码图

另由于运行平台为Windows 8.1 x64，为了充分发挥x64平台性能，程序生成为x64。

### PC端软件UI设计

设计通过多个继承自System.Windows.Forms.Form的窗口实现人机交互，其中包括：

1. Main类：包含多个ImageBox的主界面，实时显示采集画面及处理过程，用户可通过鼠标实现选择目标、控制瞄准工作启停等功能，如图 4‑2所示。
2. FormHSV类：用于设置图像处理过程的可调参数，包括设定的S、V范围，以及直方图的bin个数。
3. PortSetting类：用于设置与嵌入式系统交互的端口。
4. ParSetting类：用于PID参数调节。
5. FormRemote类：用于建立Socket服务器并设置相关参数、发送接收数据包。

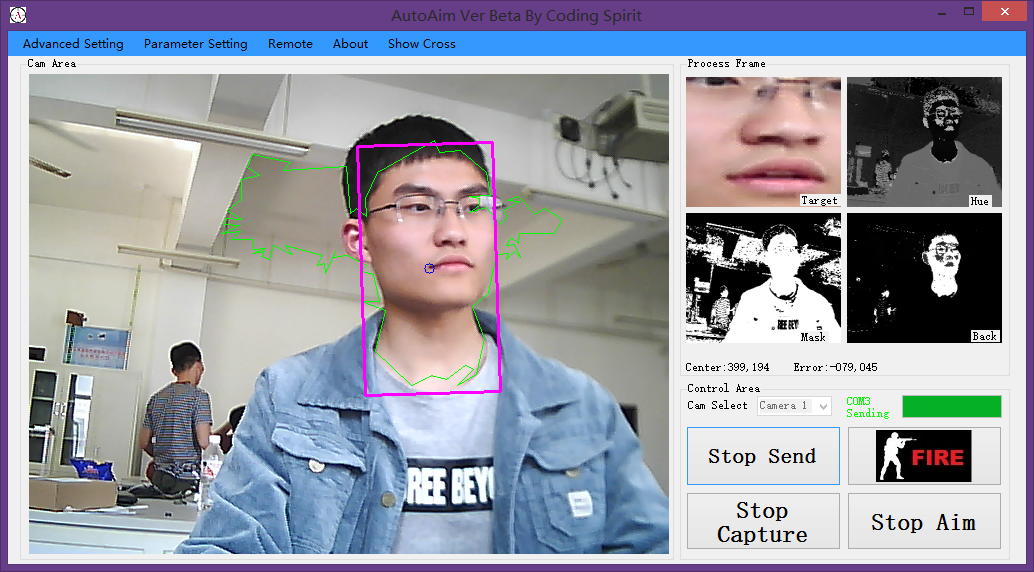


图 4‑2 PC端软件主界面

### 核心图像处理算法实现

除外部引用外，设计的核心图像处理算法均以Tracking类实现，包含在文件Tracking.cs中。设计使用CAMSHIFT算法结合密度、面积、重叠匹配、实现目标的追踪识别和丢失检测寻回。在主线程创建后台进程ProcessFrame后，ProcessFrame将会创建Tracking类的对象objTracking，之后进入图像处理流程。进程ProcessFrame中有一些需要向主UI线程发送的数据，这种跨线程的数据调用设计通过委托实现。图像处理算法流程图如图 4‑3所示。



图 4‑3 图像处理算法流程图

由于设计时考虑到追踪的目标特征可能是肤色，同一人种人体肤色大致相同或接近，肤色检测结果主要受复杂背景下光照、亮度变化影响较大，所以需要选择与亮度无关、对光照的变化不敏感并且具有良好肤色聚类特性的颜色空间模型[8]。在HSV空间中，H分量对光照变化并不敏感，因此本设计中图像处理使用了HSV颜色空间。

在将采集图像frame转为HSV格式后，通过求取用户指定追踪窗口selection内的H分量直方图，可以获得目标H分量的分布情况。图 4‑4显示了在明亮的室内时，划分16个bin时脸部肤色的H分量直方图。

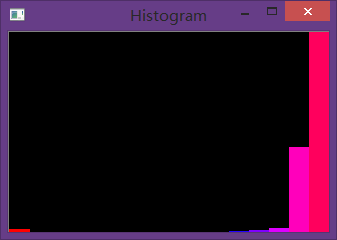


图 4‑4 脸部肤色直方图

在得到目标H分量的分布情况后，通过计算back-projection[9]，可以获得某帧图像中目标颜色特征的概率分布图。图 4‑5显示了根据肤色直方图求得的某帧图像的back-projection和原图。可以通过图像直观地看出，肤色色块被有效地分割出来。

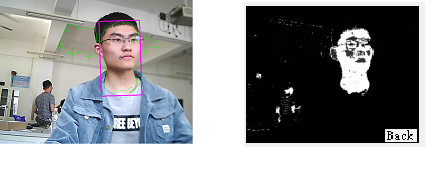


图 4‑5 原图和根据肤色直方图求得的的back-projection

在得到back-projection后，为了减少干扰并加快图像处理速度，设计使用MASK掩盖用户不关心的点。图 4‑6显示了根据用户指定SV分量值范围得到的MASK。使用MASK与back-projection求与，可以去掉大量不关心的点。



图 4‑6 根据用户指定SV分量值得到的MASK

经过前面的处理后，通过求取图像的矩可获得追踪窗口内的重心和旋转方向。若有一*i*行*j*列的二阶灰度图像矩阵，那么图像的零阶矩 的定义如下：



由定义可知，零阶矩代表了灰度图像像素点的总质量。

一阶矩、的定义如下：





由一阶矩和零阶矩可以求出矩阵的重心：



在得到selection的重心后，移动selection使得selection的中心与上次计算得到的重心重合，重复这一迭代过程直到达到迭代终止条件，如达到限制次数或者移动距离小于指定值。

为了得到目标的旋转角度，可利用二阶矩、求得矩阵的旋转角。其中二阶矩、的定义如下：





可得矩阵的旋转角：



其中：



为使追踪框能够自适应目标因透视效应造成的在镜头中的大小变化，算法实现时尝试将追踪框selection长宽都扩大20个像素，再根据求得的旋转矩的长短轴确定新的追踪框。

当确定了新的追踪框且目标尚未丢失时，设计通过统计目标连通体面积和密度均值的方式寻找目标特征。一个二维连通体可以表示为一个包含*n*个二维点的集合：



若使用这个集合中点的个数*n*来表示这个连通体的面积，则定义该连通体的密度表示为：



式中 ——处点的灰度值。

当发现求得的新跟踪框内目标连通体面积和密度均值不在给定范围内或新跟踪框过小时，判定为目标丢失。目标丢失后，系统将尝试从back-projection中寻找在给定范围内的连通体，并以此连通体的最小外接矩形为新的跟踪框，继续跟踪过程。由于目标丢失时云台停止运动，镜头内的非目标连通体面积和密度都较为稳定，因此误识别的几率大大降低。

图像处理线程实现目标的识别与追踪后，通过求跟踪框中心坐标可以得到目标在二维图像中的像素点位置。此后，程序将把该坐标与图像中心坐标的差值通过USB反馈到嵌入式系统，即可实现闭环控制。

### PC-嵌入式系统通信协议

PC端需要发送到嵌入式端的数据和指令主要有：

1. 当前目标位置与摄像头中心的位置差；
2. 当前设定的PID参数；
3. 手机端发出的手动控制指令。

由于每帧图像都会发送一次数据，为了加快嵌入式系统的数据解析速度，通信协议规定发送数据为定长的数据包。当反馈目标位置与摄像头中心的位置差时，发送数据格式如表 4‑1所示。例如，当目标位置与摄像头中心的位置差为(-200,+168)时，发送数据为“@-200+168”；

表 4‑1 位置反馈数据格式

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧起始 | X符号 | X误差 | | | Y符号 | Y误差 | | |
| @ | +/- | 0~6 | 0~9 | 0~9 | +/- | 0~3 | 0~9 | 0~9 |

当发送当前设定PID参数时，数据格式如表 4‑2所示。例如，当设置PID参数为比例参数56，积分参数10，微分参数2时，发送的数据为“P56I10D2”；

表 4‑2 PID参数设置格式

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 帧起始 | P参数 | | 间隔符 | I参数 | | 间隔符 | D参数 | |
| P | 0~9 | 0~9 | I | 0~9 | 0~9 | D | 0~9 | 0~9 |

当发送手动控制指令时，数据格式如表 4‑3所示。例如，手动控制云台水平顺时针转动一个单位，发送的数据为“[RRRRRRR]”。

表 4‑3 手动控制指令格式

|  |  |  |
| --- | --- | --- |
| 帧起始 | 指令 | 帧结尾 |
| [ | 7位连续的L/R/D/U | ] |

### Socket通信及视频传输实现

在本系统中，PC端和Android通信是通过Socket通信实现的，其中PC端作为服务器端，Android端作为客户端。其中用到的SocketServer类包含在SocketServer.cs中。服务器UI线程创建服务器后台线程后，一个SocketServer对象server被创建，并在其构造方法中对用户指定的IP和端口创建TcpListener监听。Android端试图连接时，触发自定义事件ConnectCallBack，建立异步接收线程和异步发送线程进行通信，并保持连接直到一方断开连接。

与图像处理后台进程的跨进程访问方式不同，服务器后台线程和主线程数据交换是通过自定义事件实现的。相关事件均在ServerEventHandler中处理。

当Socket通信实现后，通过写入、读取流可以很容易地实现数据传输。在开发初期时，尝试了以直接传送定长bmp图像实现的视频传输，但由于bmp图像未经压缩，文件尺寸较大，在达到测试网络实际极限速度3.8Mb/S时仍无法达到25fps导致视频显示有明显卡顿。后采用压缩算法将图片压缩为jpg格式，但此方法压缩得到的图片长度不定，因此需要设计不定长传输协议。在该协议中，实际发送的数据前会有一包说明数据包长度和类型的包头，其定义如表 4‑4所示。

表 4‑4 包头定义

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 帧起始 | | | 数据长度 | | | | |
| # | ! | # | 0~9 | 0~9 | 0~9 | 0~9 | 0~9 |

在发送包头后，再发送实际数据，即可实现不定长传输。

## Android端软件设计

### Android端软件总体设计

由系统总体设计可知，Android端应用主要的任务有：

1. 建立Socket客户端并接收服务器端（PC）传输的视频画面、文本内容并显示；
2. 远程控制系统自动、手动瞄准的切换；
3. 提供手动瞄准的用户交互界面。

由于显示内容较多，单一的Activity已经无法满足任务需求，因此Android端软件设计为包含SplashActivity和MainActivity的多界面应用，并在MainActivity中通过ViewPager和继承自PagerAdapter的MyPagerAdapter类实现文本、视频显示的滑动切换。此外，Android端应用也使用多线程模式防止复杂的数据处理导致UI界面失去响应。

另由于Linux的权限访问控制，而程序需要使用到网络，申请一些网络相关的权限是必需的。表 4‑5给出了在AndroidManifest.xml中申请的权限及说明。

表 4‑5 申请的权限

|  |  |
| --- | --- |
| 权限名称 | 说明 |
| android.permission.INTERNET | 允许程序打开网络套接字 |
| android.permission.ACCESS\_WIFI\_STATE | 允许程序访问Wi-Fi状态信息 |
| android.permission.CHANGE\_WIFI\_STATE | 允许程序改变Wi-Fi连接状态 |
| android.permission.CHANGE\_NETWORK\_STATE | 允许程序改变网络连接状态 |

### Android端软件UI设计

SplashActivity设计为启动界面，主要显示应用信息。在启动界面消失后启动主界面MainActivity，主界面第一页设计为连接设置和文本收发，第二页设计为视频显示和遥控器界面。设计效果如图 4‑7所示。

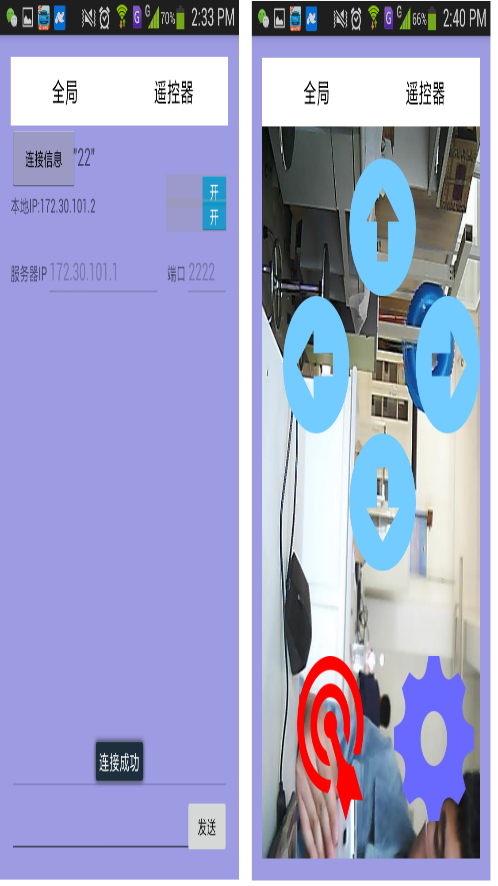


图 4‑7 Android端UI设计

### Android端Socket通信实现

在本系统的通讯过程中，Android端是作为客户端与PC端连接的。程序使用到的SocketClient类包含在SocketClient.java中。进入主界面时，一个SocketClient对象马上被创建，当用户从连接设置界面完成对连接到的服务器的IP和端口号设置后，程序将启动线程connectThread并尝试向该IP和端口发起连接请求，并在连接成功后建立sendThread和ReceiveThread线程并尝试保持连接，直到一方离线。

ReceiveThread线程在连接成功后将一直尝试从输入流中抓取数据包包头。在发现包头后，实际的数据长度得到确认，可以从数据长短判断数据类型，从而进行分类操作。使用BitmapFactory类的decodeByteArray方法可以将流中的字节数据转为bmp格式并在UI上显示。

sendThread线程在连接成功后将一直尝试将发送缓冲区的数据写入输出流。当用户操作UI界面的遥控器或者发送文字数据时，sendThread线程将把这些数据发送到PC端。其中遥控器发送指令协议如表 4‑6所示。

表 4‑6 遥控器指令

|  |  |
| --- | --- |
| 指令 | 功能 |
| /UP/ | 云台仰角增大或俯角减小 |
| /DOWN/ | 云台仰角减小或俯角增大 |
| /LEFT/ | 云台水平逆时针转动 |
| /RIGHT/ | 云台水平顺时针转动 |
| /FIRE/ | 继电器控制 |
| /MANUAL/ | 切换到手动控制模式 |
| /AUTO/ | 切换到自动控制模式 |

## 嵌入式系统软件设计

由于嵌入式系统使用了FreeRTOS V9.0.0作为系统内核，MCU在HAL库初始化和系统时钟配置完成后将配置FreeRTOS并启动系统内核和任务调度器，因此本设计的嵌入式系统的大部分功能实现都是在FreeRTOS中以任务方式完成的，发生外部中断、定时器中断时，回调函数也会通过任务通知告知相应的处理任务。表 4‑7给出了系统中使用到的任务列表。

表 4‑7 嵌入式系统任务列表

|  |  |  |
| --- | --- | --- |
| 任务名称 | 优先级 | 功能 |
| getStateTask | osPriorityHigh | 数据包接收与解析 |
| controlTask | osPriorityHigh | 自动/手动控制进程 |
| outPutTask | osPriorityNormal | 输出电机控制信号 |
| keysTask | osPriorityBelowNormal | 按键响应 |

除上述任务外，为了控制采样周期和利用定时器输出指定个数的脉冲，系统还使用了TIM1、TIM3、TIM8的定时器中断。设定TIM3的溢出中断每15ms触发一次，并在中断回调函数中向getStateTask发送任务通知以将闭环控制周期设置为15ms，即每隔15ms系统将从USB接受缓冲区取出一段数据包解析当前指令，进而切换到controlTask根据数据和指令进行控制，再通过outPutTask输出控制信号。

利用高级定时器TIM1和TIM8的RepetitionCounter触发定时器更新中断可以实现指定个数的脉冲信号输出。

系统还使用了四个外部中断来响应按键调试操作。当外部中断发生后，中断回调函数将向keysTask发送任务通知，由keysTask进行相应的操作。

## 本章小结

本章详细介绍了核心图像处理算法的实现过程，分别介绍了PC端软件和Android端应用的UI设计和通信实现，介绍了嵌入式系统软件的设计实现。

# 系统性能评估

本章将通过分析系统测试数据来评估系统相关的性能指标。测试的环境为明亮的室内，测试的瞄准目标分别为人脸和一个青色的瓶状物体。PC端程序运行操作系统为Windows 8.1 x64，PC处理器为四核八线程的Intel Core i7-3630QM，主频2.4GHz。

## 帧率测试

当设置曝光度为-6时，系统采集图像帧率稳定在33fps。测试结果如图 5‑1所示。

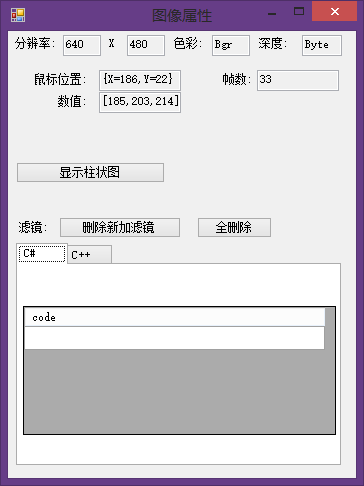


图 5‑1 系统帧率测试

## 瞄准精度评估

### 静态目标瞄准精度评估

测试了系统在测试环境下对近乎于静态的人脸目标的瞄准精度。其结果如图 5‑2所示，测试中系统的稳定在3以下。

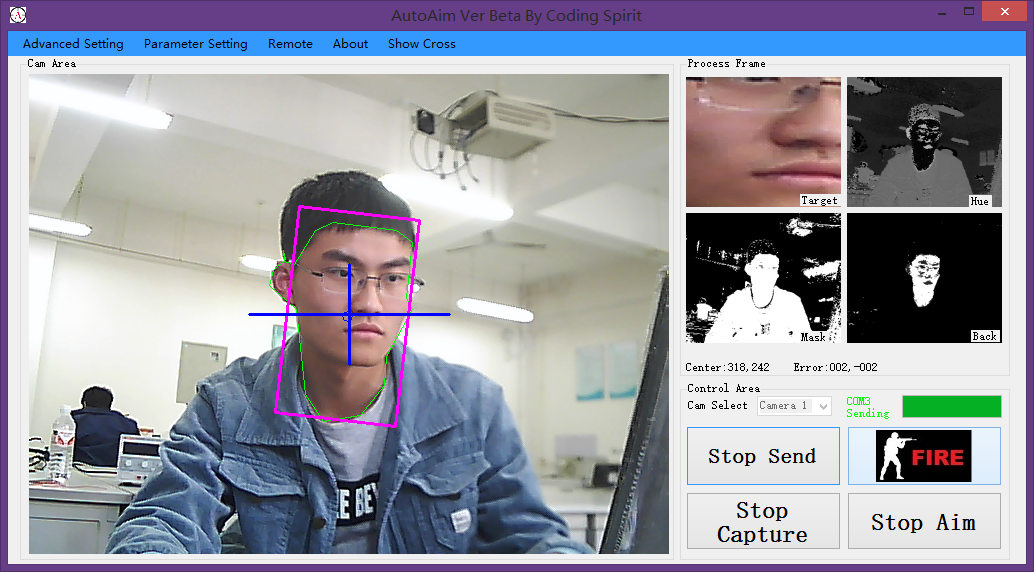


图 5‑2 静态人脸目标瞄准测试

### 动态目标瞄准精度评估

分别测试了系统对于两个运动目标的瞄准精度。在测试中，人脸目标在距离摄像头2m的位置以正常步行速度运动，瓶状目标在距离摄像头1.5m处做单摆运动。图 5‑3显示了测试中系统对运动的脸部目标的追踪情况，测试过程中系统的最大值为45。

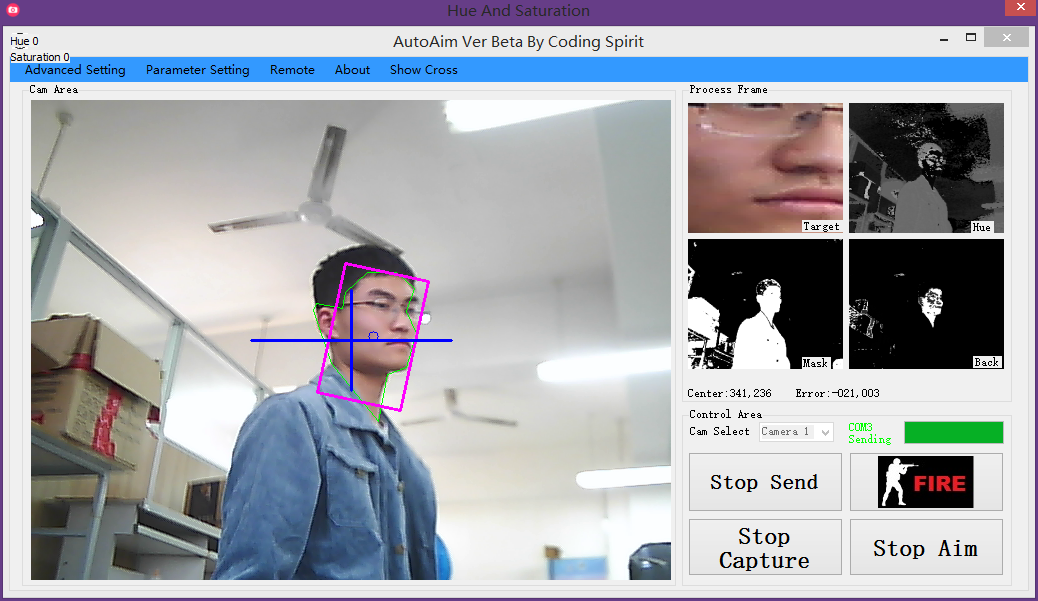


图 5‑3 动态脸部目标瞄准测试

对于做单摆运动的瓶状目标，当摆角小于等于60°时，测试系统表现出了良好的跟踪性能，系统的能够快速稳定在20以下，图 5‑4显示了测试时当摆角为30°时系统的工作情况。当摆角大于60°时，默认的云台运动速度已经无法跟上目标运动，虽然通过调整步进电机驱动器细分数可以加快云台运动速度，但这也导致系统的瞄准精度降低，当摆角为60°，细分数设置为32细分时，测试中最大为28，测试中系统的工作情况如图 5‑5所示。

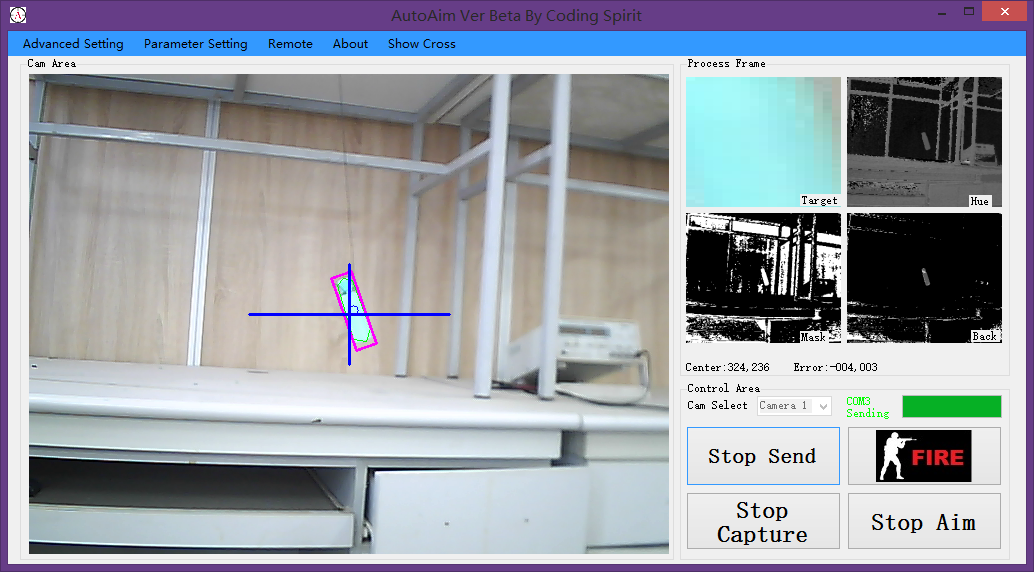


图 5‑4 目标摆角为30°时系统性能测试

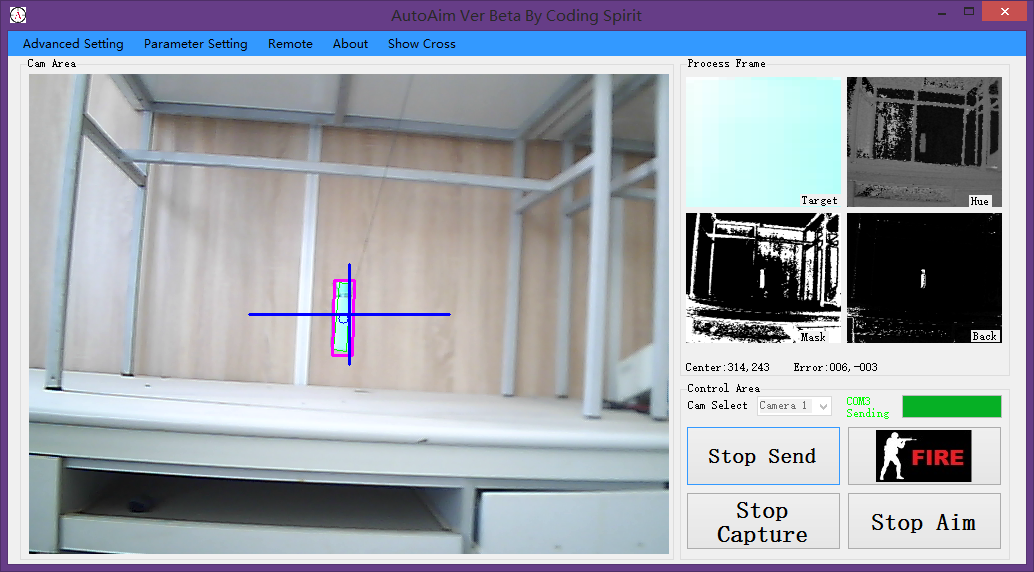


图 5‑5 目标摆角为60°时系统性能测试

## 系统资源占用情况分析

在测试平台中，启用跟踪瞄准并启动Socket服务器进行25fps视频传输时，PC端应用AutoAimProject.exe进程占用内存43.1MB。由于进行了大量的运算，并且只做了多线程优化而未进行多核优化，程序运行过程中只使用了一个CPU核心，CPU占用率为59.4%。网络带宽占用约为4.5Mbps，传输速度约为576KB/s测试结果如图 5‑6所示。



图 5‑6 PC软件系统资源占用情况

## 本章小结

本章介绍了系统测试的结果，并通过结果分析了系统的相关性能参数。由测试结果可知，系统工作性能满足设计要求。

# 总结与展望

第1章 绪论

绪论中需要交代清楚以下内容：选题的目的和意义；国内外的发展现状；本文的结构和内容。

第2章 总体设计

第3章~第n章，对于设计中的模块进行详细说明

第n+1章 总结与展望

具体情参阅《规范》文档。

在完成论文后，应进行格式自查：

* 所有中文字体和中文标点，是否按要求仅使用宋体和黑体而无其他字体？
* 所有数字、西文字体和符号，是否仅采用新罗马体（Times new Roman）而无其他字体？
* 所有图、表是否居中？题注是否已套用“题注”样式？
* 所有表格是否采用开口或三线表？文字是否已套用“表中文字”样式？
* 所有正文内容是否已套用“论文正文”样式？
* 所有英文变量符号是否已经斜体？
* 文中还有超过3行的大片空白吗？若有请将后面的文字提上来。
* 所有参考文献的引用和标注是否符合要求？

当你能信心满满的对以上问题回答“是”时，就说明此论文的格式基本满足提交要求了。最后找同学翻阅一下你的论文，看看通篇是否还有语言不通顺、排版不美观的地方，如有请按要求进行修改。

参考文献

1. Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110(3):404-417.
2. Bradski G R. Computer Vision Face Tracking For Use in a Perceptual User Interface[C]// IEEE Workshop on Applications of Computer Vision. IEEE, 1998:214-219.
3. Mita T, Kaneko T, Hori O. Joint Haar-like Features for Face Detection[C]// Tenth IEEE International Conference on Computer Vision. IEEE Xplore, 2005:1619-1626 Vol. 2.
4. Huang S, Hong J. Moving object tracking system based on CAMSHIFT and Kalman filter[C]// International Conference on Consumer Electronics, Communications and Networks. IEEE, 2011:1423-1426.
5. Barry R. Using the FreeRTOS Real Time Kernel - a Practical Guide[J]. 2010.
6. 廖义奎. ARM Cortex-M4嵌入式实战开发精解:基于STM32F4[M]. 北京航空航天大学出版社, 2013.
7. 晋晶晶. USB2.0中物理层接口的设计[D]. 北方工业大学, 2008.
8. 王鼎, 沈辉, 娄海涛. 一种基于H-CrCb颜色空间的肤色检测算法研究[J]. 计算机科学, 2012, 39(s2):223-226.
9. Xu M, Wang L V. Universal back-projection algorithm for photoacoustic computed tomography.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2005, 71(2):016706.

注意，

* 参考文献10篇以上，至少2篇英文；
* 五号字，单倍行距。中文宋体，英文Time New Romans；
* 格式按照国标规范GBT-7714-2005执行，可参考第4章；
* 建议采用CNKI E-Study来规范参考文献的格式；
* 参考文献中采用英文的标点符号，参考文献末尾以“.”结束。参考文献需要在正文中引用。
* CNKI E-Study生成的参考文献字体和段落样式略有问题，需要按此部分再修改一下。

致谢

简述自己对本论文（设计）工作的体会，并对在课题研究和设计说明书（论文）撰写过程中曾直接给予帮助的人员（例如指导教师、答疑教师及其他人员）表示自己的谢意。

例：

衷心感谢导师×××教授对本人的精心指导。……，他的言传身教将使我终生受益。

感谢×××教授，以及实验室全体老师和同窗们的热情帮助和支持！

感谢×××同学在×××方面给予我的帮助。

感谢家人一直以来的支持。

本课题承蒙××××基金资助，特此致谢。

…

附录