

# 硕士学位论文

## 基于嵌入式 GPU 的机载下视 目标跟踪器研制

### DEVELOPMENT OF AIRBORNE TARGET TRACKER BASED ON EMBEDDED GPU

李秋晨

哈尔滨工业大学  
2017 年 6 月

国内图书分类号：TP391.4

学校代码：10213

国际图书分类号：621.3

密级：公开

## 工学硕士学位论文

# 基于嵌入式 GPU 的机载下视 目标跟踪器研制

硕 士 研 究 生：李秋晨

导 师：李君宝 副教授

申 请 学 位：工学硕士

学 科：仪器仪表工程

所 在 单 位：电气工程及自动化

答 辩 日 期：2017 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TP391.4

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

# **DEVELOPMENT OF AIRBORNE TARGET TRACKER BASED ON EMBEDDED GPU**

<b>Candidate:</b>	Li Qiuchen
<b>Supervisor:</b>	Associate Prof. Li Junbao
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Speciality:</b>	Instrumentation Engineering
<b>Affiliation:</b>	School of Electrical Engineering and Automation
<b>Date of Defence:</b>	June, 2017
<b>Degree-Conferring-Institution:</b>	Harbin Institute of Technology

## 摘 要

随着无人机对地探测技术在军事及民用领域的广泛应用,利用机载跟踪设备对地面目标进行自动定位的应用需求越来越高。通过在无人机等飞行设备上搭载自动跟踪器可以辅助执行定位跟踪、定点火力打击等任务。因此,研制机载自动跟踪设备有着重要的应用意义。由于成像高度的原因,地面目标通常面积较小而难以提取有效的特征信息,为跟踪技术在机载场景的应用增加了难度。目前的机载跟踪设备多用于对开阔视野中的地面目标进行跟踪定位,当遇到遮挡区域时经常发生目标丢失的现象,极大地影响了跟踪性能,不能满足应用需求。为此,本课题在研究抗遮挡的鲁棒目标跟踪算法的基础上,研制基于嵌入式 GPU 的目标自动跟踪设备。

首先,课题对机载跟踪系统的应用进行了需求分析,给出系统性能指标并对机载跟踪系统设计了总体研制方案与应用方案。为改善遮挡导致跟踪失败的问题,在对多类跟踪方法进行特性分析的基础上,考虑适应性改进造成算法复杂度提高的问题,从跟踪速度角度出发,选择基于时空上下文的高速目标跟踪技术作为机载系统的跟踪方案。在此基础上设计抗遮挡方案,从遮挡时刻判断、遮挡区域轨迹预测及遮挡后的目标重捕三方面对抗遮挡技术进行了深入研究并制定了解决方案。

其次,在嵌入式 GPU 平台 Jetson TX1 进行了机载跟踪器的研制。课题基于 CUDA 开发环境,利用 Eclipse 结合 OpenCV 以交叉编译的方式对高速跟踪技术及抗遮挡方案的各个环节进行了技术实现,形成抗遮挡跟踪方案的应用开发。并针对跟踪器的实时性要求,利用 CUDA 库函数调用平台的 GPU 资源,通过并行运算进行算法加速以提高跟踪速度;在此基础上,基于 Qt 设计了人机交互界面,实现了对机载跟踪器的控制与显示功能。

最后,针对系统的稳定性和可靠性要求,对机载跟踪系统的抗遮挡能力、场景适应性、人机交互功能及跟踪的速度、准确率等性能进行了全面测试。测试结果表明,本课题研制的机载下视目标跟踪器可对红外及可见光两种场景下的地面目标进行准确跟踪,可对遮挡问题进行有效处理,并支持系统的人机交互;各项跟踪参数均满足系统性能指标,符合机载跟踪器研制的技术要求。

**关键词:** 目标跟踪; 机载下视; 嵌入式 GPU; 时空上下文

## Abstract

With the wide application of UAV detection technology in military and civilian fields, the demand for automatic tracking of ground targets by airborne equipment is becoming more and more urgent. Ground people can control the airborne tracker to finish the monitoring or fixed-point firing tasks automatically. Therefore, the development of airborne automatic tracking equipment has important application significance. Due to the imaging height, ground targets are usually small and difficult to extract effective information for tracking, which adds difficulty to tracking applications in airborne scenarios. The current airborne tracking equipment usually track in the field which has an open view. When in occlusion area, target is often lost, which affects the tracking performance greatly and cannot meet the application requirements. This paper makes research of anti-occlusion robust target tracking algorithm and develops a target automatic tracking device based on embedded GPU.

Firstly, this paper analyzes the requirements of airborne tracking system, gives the system performance indicators and designs the framework of the airborne tracking system from the perspective of research and application. In order to improve the problem of tracking failure, through the analysis of characteristics of numbers of tracking methods, the high-speed target tracking technology based on spatio-temporal context learning is chosen as the airborne system tracking scheme from the point view of tracking speed. Then this paper designs the anti-occlusion scheme and studied the technology from the aspects of occlusion time judgment, trajectory prediction and the target reoccupation and makes a reasonable solution.

Secondly, this paper uses the embedded GPU platform Jetson TX1 to carry on the development of airborne tracker. Based on CUDA development environment, the application is implemented in Eclipse combined with OpenCV in a cross-compiling way to form the high-speed tracking application and the anti-occlusion application. According to the real-time requirements of the tracker, the GPU resources in the embedded platform are called by CUDA library function to accelerate the tracking speed by the parallel operation. Then, this paper designs the human-computer interaction interface based on Qt to control the airborne tracker and shows the tracking results.

Finally, according to the stability and reliability requirements of the system, the function of anti-blocking, scene adaptability, human-computer interaction and

the performance parameters such as tracking speed and accuracy of the airborne tracking system are tested. The results show that the airborne target tracker can track the ground targets accurately in both infrared and visible scenes. The system can handle the occlusion problems encountered in the tracking process and support the human-computer interaction of the system. The tracking parameters of the system meet the requirements of airborne tracker.

**Keywords:** target tracking, airborne scene, embedded GPU, spatio-temporal context

# 目 录

摘 要 .....	I
Abstract .....	II
第 1 章 绪 论 .....	1
1.1 课题背景及研究意义 .....	1
1.2 研究现状分析 .....	3
1.2.1 机载跟踪系统研究现状 .....	3
1.2.2 目标跟踪关键技术研究现状 .....	7
1.2.3 存在的问题 .....	11
1.3 主要研究内容 .....	12
1.4 论文组织结构 .....	12
第 2 章 机载跟踪系统方案设计与抗遮挡技术研究 .....	14
2.1 需求分析与系统性能指标 .....	14
2.1.1 需求分析 .....	14
2.1.2 系统性能指标 .....	15
2.2 机载跟踪系统方案设计 .....	15
2.2.1 机载跟踪系统总体方案设计 .....	15
2.2.2 机载跟踪系统应用方案设计 .....	16
2.3 机载下视抗遮挡跟踪技术研究 .....	17
2.3.1 多类跟踪方法特性分析 .....	17
2.3.2 基于时空上下文的高速跟踪技术研究 .....	19
2.3.3 抗遮挡跟踪技术研究 .....	24
2.4 本章小结 .....	36
第 3 章 嵌入式机载跟踪系统实现 .....	37
3.1 系统环境搭建 .....	37
3.1.1 硬件平台搭建 .....	38
3.1.2 软件环境部署 .....	40
3.2 抗遮挡跟踪技术实现 .....	42
3.2.1 高速跟踪技术实现 .....	42
3.2.2 抗遮挡跟踪技术实现 .....	44

3.3 GPU 加速及人机交互实现 .....	48
3.3.1 跟踪技术的 GPU 加速 .....	48
3.3.2 人机交互界面的设计与实现 .....	50
3.4 本章小结 .....	53
<b>第 4 章 机载跟踪系统综合测试与分析 .....</b>	<b>54</b>
4.1 测试环境与测试方法 .....	54
4.2 机载跟踪系统功能测试 .....	57
4.3 机载跟踪系统性能测试 .....	60
4.4 系统综合测试分析 .....	61
4.5 本章小结 .....	62
<b>结 论 .....</b>	<b>63</b>
<b>参考文献 .....</b>	<b>65</b>
<b>攻读学位期间发表的论文及其它成果 .....</b>	<b>71</b>
<b>哈尔滨工业大学学位论文原创性声明及使用权限 .....</b>	<b>72</b>
<b>致 谢 .....</b>	<b>73</b>



# 第 1 章 绪 论

## 1.1 课题背景及研究意义

本课题来源于“机载下视目标跟踪技术研究”的项目需求。基于此背景，本课题对适应于机载下视场景的目标跟踪技术及其在嵌入式平台的实现进行了深入研究。机载下视场景的运动目标跟踪，通常是对如 UAV(Unmanned Aerial Vehicle)即无人机等飞行设备采集的地面场景视频中的运动目标进行主动式跟踪，实现在前视频中目标的实时定位等功能<sup>[1]</sup>。作为视觉跟踪技术的一个具体应用场景，该种跟踪通常具有跟踪范围广、机动灵活、安全性高等特点，既可以避免地对地跟踪中视线受限及危险性高的缺点，又可以弥补卫星侦察细节不足、实时性差的缺陷，可以对目标进行准确跟踪并及时向观察人员提供目标的位置信息<sup>[2]</sup>。由于飞行器的部署便捷性和运动范围的广阔性，对目标进行主动式跟踪在军事侦查和导航领域优势明显。

机载下视场景的目标跟踪可理解为通过无人机等飞行设备对地面目标进行实时定位跟踪。跟踪系统所处的环境会对跟踪性能产生较大影响。机载环境与摄像头固定的场景监控的不同之处在于，机载平台会跟随飞行设备的振动和自主运动发生抖动及晃动等情况，增加目标跟踪的难度<sup>[3]</sup>。同时，成像设备的运动会影响对目标搜索区域的估计，例如只有将镜头运动的影响计算进去后才能考虑匀速运动、直线运动等理想情况<sup>[4]</sup>。此外，由于机载视频视野中的背景区域的不断变化，也就无法利用背景建模进行分析处理，同时由于无法确定目标的运动模式，也就无法建立运动模型从而准确估计目标状态，增加了此应用场景下跟踪技术的难度。

下视角度对地面目标进行跟踪也会影响目标跟踪的效果。由于无人机等飞行平台通常距离地面目标较远，拍摄到的成像场景较为复杂<sup>[5]</sup>，会存在与目标特征重合的干扰区域，影响跟踪的准确度，容易造成跟错目标的情况；且地面场景中的目标往往面积较小，难以从中提取有效的特征信息用于分析处理，这对提取目标特征的方法提出了较大要求，增加了跟踪定位的难度，并会对此种情况下跟踪的准确性和鲁棒性产生较大影响<sup>[6]</sup>。

目标跟踪除了实现目标定位与状态估计等功能，还可以为高级识别做铺垫<sup>[7]</sup>。如可以在高空进行全方位多角度的对地观察，锁定目标并进行实时定位，采集有效信息；目标跟踪还可以用于分析目标行为，通过分析运行轨迹，预测

目标的运动趋势,估计目标下一步的行为以提前做好准备;随着无人机设备的迅猛发展,将目标跟踪技术应用于无人机设备进行地面目标的定位追踪得到了广泛关注<sup>[6]</sup>。对于执行操控人员来说,跟踪地面目标是一项比较枯燥且难以长时间反复执行的工作,而这类工作在机载平台上又大量的存在,因此对地面目标进行自动跟踪的需求越来越大,而将目标跟踪技术应用于机载场景可以将监视人员从长时间执行的跟踪工作中解救出来,在目标行为异常时向操作人员发出警告,提醒工作人员进行处理<sup>[8]</sup>。

上述应用场景的特点对跟踪器的硬件提出了一定的要求,要实现机载平台的实时跟踪,不仅需要选体积小、重量轻的小型化设备,为能实时处理复杂的跟踪方法,设备必须具备高速运算能力。考虑到飞行平台尤其是无人机等小型设备的体积小,功耗及承重力有限,难以搭载 PC 机等大型平台,嵌入式系统作为目标跟踪的硬件平台是不二之选,而目前市场中,可实现目标跟踪的嵌入式模块非常少,开发可搭载在机载平台,并对地面特定目标执行跟踪任务的嵌入式模块有着十分广泛的应用市场;另一方面,为弥补嵌入式平台由于资源受限而对数据运算能力的影响,可选用具有高性能运算能力的硬件设备进行机载跟踪系统的应用开发,本课题利用嵌入式 GPU 平台进行机载目标跟踪系统的应用研究,利用 GPU 的高速并行计算能力加快目标跟踪定位过程的处理效率,满足对目标实时性跟踪的要求。

机载下视场景的运动目标跟踪技术具有很高的实用价值,在军事及民用的各个领域得到了广泛应用<sup>[9]</sup>。在军事方面,机载平台的目标跟踪系统已成功应用于无人遥感、战场实时监控及目标导航定位等任务的执行,利用无人机等小型飞行设备对地面目标进行高空监控和定位追踪是跟踪技术的重要应用<sup>[22]</sup>。此外,制导技术的日益完善与无人机的大量服役为空对地目标跟踪开辟了广阔的应用领域,如在高空领域对地面目标进行实时跟踪监控、对地目标的定点打击等<sup>[10]</sup>。这些任务成功执行的关键就是飞行条件下动态目标的跟踪问题。随着航空领域飞行器设备的不断发展,机载平台信息采集、分析和控制系统逐渐趋向于高性能化,在飞行平台搭载跟踪设备执行对地自动探测与跟踪已在民用领域得到了广泛关注与发展,如利用自动跟踪系统空中执行智能交通检测任务,远程监控公路道路情况,通过检测道路上的交通流量及是否存在异常行为车辆可及时了解交通情况,在交通事故的预防和事后处理中有着重要应用<sup>[47]</sup>。机载跟踪设备可通过识别空间地表形状进行地质、海洋及石油矿产等资源的勘探;还可用于对地理灾害、环境污染、森林火灾等重大突发性事件的监控以及生态保护、农业规划、森林植被规划等用途,甚至在摄影领域都有着一席之地。由

此可以看出,嵌入式机载跟踪系统在国民经济的各个领域都有着重要的应用意义与实际需求<sup>[12]</sup>。

机载下视目标跟踪设备的研制,一方面可以针对复杂的应用场景对目标跟踪技术展开深入研究,促进目标跟踪领域的发展;另一方面,机载跟踪设备可用于在高空执行地面目标状态检测、定位等多种任务,具有重要的研究意义与应用价值<sup>[13]</sup>。

## 1.2 研究现状分析

为实现视频场景中,尤其是机载下视等复杂场景下对目标的自动定位与状态估计的应用,国内外都针对运动目标跟踪技术及其在各个领域的应用展开了相关探索与研究。国外尤其是美国、欧洲等国的研究始于二十世纪九十年代,无论从目标跟踪的关键技术还是可进行实际应用的跟踪系统方面都处于领先地位<sup>[13]</sup>。相比而言,国内对于跟踪技术及其实际应用的研究起步较晚,随着对目标跟踪的实际应用需求越来越大,许多高校及研究部门也相继从跟踪技术及应用层面开展了一系列的研究探索并取得了一定成绩。

### 1.2.1 机载跟踪系统研究现状

国外对机载下视场景目标跟踪系统的研制开展较早且具备成型系统。受到较多关注的有 1997 年由美国国防高级研究项目署(Defense Advanced Research Projects Agency, DARPA)立项,卡内基梅隆大学等多所美国知名高校及研究机构参与研制的视频监控系统 VSAM(Visual Surveillance and Monitoring)<sup>[14]</sup>。该系统针对机载下视场景运动目标的检测与跟踪技术做了研究,提出了对平台抖动及复杂场景检测跟踪的解决方案,具有良好的鲁棒性与实时性。为实现对无人机平台采集的视频中的目标进行自动跟踪,2005 年佛罗里达大学的计算机视觉实验室基于 MATLAB 开发了 COCOA 系统<sup>[15]</sup>,该系统不仅可适应红外及可见光视频源,还实现了多目标的识别与跟踪技术。

国内对机载下视场景运动目标跟踪系统的研究主要基于 PC 机和嵌入式平台两种系统环境。早期对跟踪技术的研究主要是在 PC 机中进行的,其优异的计算能力及众多的开发环境支持为深入开展跟踪算法理论研究提供了良好的平台<sup>[16]</sup>,但体积大、功耗高的特点限制了其在特定场合尤其是机载环境的使用。要将跟踪技术用于实际应用场景中,使用嵌入式系统是必然要求<sup>[17]</sup>。嵌入式平台一方面由于体积小,可搭载于机载环境;另一方面,嵌入式芯片通常

具有很强的运算能力，可满足跟踪的实时性需求。可用于目标跟踪系统研制的嵌入式架构有 DSP+FPGA 和 DSP+ARM 两种。随着 GPU 技术的兴起，GPU+ARM 架构为嵌入式目标跟踪系统的研究注入了新动力<sup>[18]</sup>。

DSP 是具有数据运算功能的微处理器，主要用于对数据处理算法的快速实现<sup>[19]</sup>，可用于目标跟踪技术的研究。但随着视频信息量的不断增大及对实时处理的要求越来越高，跟踪算法的复杂度越来越高，单核 DSP 已无法满足现代视频处理系统的实时性要求<sup>[20]</sup>；且 DSP 以处理数据运算为主，在系统控制方面的能力较弱，仅利用 DSP 构成嵌入式系统几乎不可能，需要与其他硬件结合改善系统的处理速度与控制能力。现场可编程门阵列 FPGA 可实现结构简单计算量大的运算处理、且可通过内部逻辑门的电路设计实现系统的控制功能。二者可进行功能互补，形成完整的嵌入式目标跟踪系统。DSP 用于处理具有复杂结构的逻辑运算，FPGA 则用于实现结构简单，运算量大的算法及对系统的控制功能<sup>[21]</sup>。考虑到开发成本问题，当前的嵌入式系统以此类架构为主。国内对目标跟踪系统的自主研发较少，AVT31 是当前市场中可调研到的嵌入式跟踪处理器，如图 1-1 所示，由上海凯视力成公司自主研发生产，其主控芯片由多核 DSP 和 FPGA 构成，是一款高清视频自动跟踪模块。



图 1-1 基于 DSP+FPGA 架构的 AVT31 模块图

该嵌入式模块支持模拟及数字两种形式的高清视频分量输入，图像分辨率最高可达 1080P@60Hz，内置白热、黑热、动态检测等多种视频预处理技术，可对静态及动态场景下的跟踪目标进行自适应图像增强，提高运动过程中的图像质量；提供了多种目标跟踪方式，可适应不同的应用需求，如质心跟踪、相位相关跟踪、多目标跟踪、边缘跟踪、组合跟踪等，并支持对多目标的检测与跟踪；具有强大的图像处理能力，场景锁定情况下具有实时电子稳像、视频融合等功能；支持视频的压缩、记录、网络传输等功能，可对运动平台的速度及

位置进行控制；模块中的 OSD 功能可通过标准字符或用户自定义字符如符号标记、瞄准线、目标窗口等进行结果输出。AVT31 可满足较为复杂的光电跟踪应用，具有高性能、低功耗、小型化、轻量级等特点<sup>[20]</sup>，可用于机载平台目标跟踪系统的研制。通过调研，该产品可跟踪的最小目标尺寸为 6\*3 像素，最大目标尺寸为 200\*100 像素。可处理的最大图像尺寸为 1920\*1080 像素，跟踪速度与准确率由于受场景、目标大小等因素的影响，无具体指标。

DSP+ARM 架构的嵌入式系统以 TI 公司推出的达芬奇系列产品为主，是专用于视频图像处理的高度集成片上系统，通过内核集成将 ARM 强大的事务管理能力与 DSP 的信号处理能力结合，ARM 用于完成系统管理、人机交互及控制外部设备等工作<sup>[23]</sup>，DSP 则用于实现复杂的图像处理算法，二者将优势充分结合，可满足对视频目标跟踪技术的实时处理需求和控制系统功能的要求。市面上有多款以达芬奇片上系统为核心的嵌入式视觉开发模块，如北京艾睿合众公司基于达芬奇高清视频处理芯片 DM6467 研发了一款嵌入式高清视频开发平台 VPM6467，可用于机器视觉的通用开发<sup>[24]</sup>。该平台由以 DM6467 为核心的处理单元及视频输入单元构成，具备对视频的多路编解码及视频采集功能，集成了多种高清视频接口与数据传输外设，可用于高清视频处理领域的开发应用。但通过调研，该类模块更多地应用于对常规视频处理的开发，如高清视频会议系统、机器视觉应用、医疗成像应用等，且其开发模块为 PCI 全长卡，并不适合搭载在飞行平台上。因此，该类模块不合适机载跟踪系统的研制。

具有强大计算和并行处理能力的 GPU 技术作为一种新兴的计算资源，正被应用于嵌入式机器视觉的研究<sup>[25]</sup>。GPU 早期主要用于图形图像的渲染工作，分担了 CPU 的大量图形处理任务，使 CPU 可以更高效地进行系统控制。CPU 与 GPU 的异构组合可以提高系统的整体性能。随着技术的不断发展，GPU 已不再局限于视频显示工作，人们开始研究利用 GPU 完成更多的任务，如通过调用系统内部 GPU 进行应用程序的开发，利用其强大的并行计算能力进行加速处理，提高执行程序的处理速度。近年来涌现出了众多的 GPU 开发技术，如通用并行计算架构 CUDA(Compute Unified Device Architecture)和 OpenCL 的发展大大提高了 GPU 的可编程性<sup>[ ]</sup>，开发人员可使用高级语言如 C/C++ 为 CUDA 架构编程，进而调用 GPU 以超高性能运行应用程序以实现特定功能。

基于 GPU+ARM 架构的嵌入式系统受到关注度最高的当属 NVIDIA 公司推出的 Jetson 人工智能计算模块，如图 1-2 所示。由于开发难度大、研发起点高，且考虑到硬件成本问题，当前市场还没有完备的基于 GPU+ARM 架构的嵌入式机载跟踪系统的开发平台，对利用 GPU 设备进行目标跟踪系统的应用

开发还处于研究阶段。

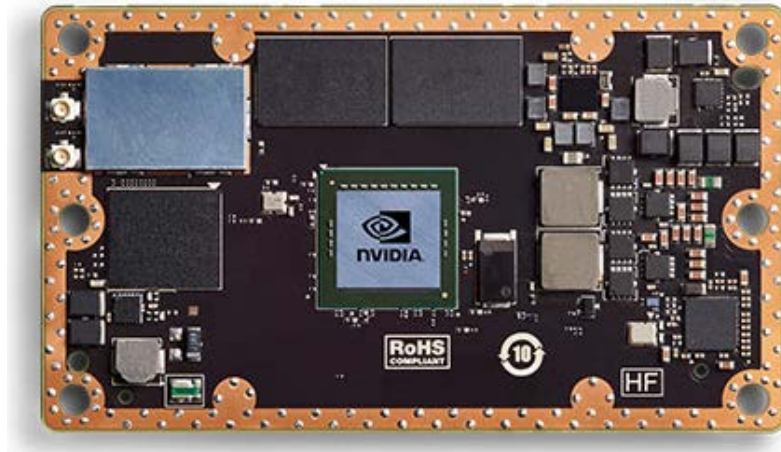


图 1-2 基于 GPU+ARM 架构的 Jetson 模块图

Jetson 模块作为业内领先的 AI 计算平台，面向移动嵌入式系统市场中的 GPU 加速并行处理，可进行深度学习、机器视觉等领域的应用研发。ARM 架构使得模块支持板上操作系统、开发环境及第三方库部署，可脱离对 PC 机的依赖，以微型电脑的形式在板上进行系统应用的独立开发<sup>[26]</sup>。NVIDIA 官方提供了对 Jetson 模块的软件开发环境部署，通过包含 BSP、深度学习库、计算机视觉、GPU 计算、多媒体处理的完整软件开发包来提高软件开发的速度<sup>[20]</sup>；同时，NVIDIA 提供了对 Jetson 开发者套件如 TX1、TX2 等的支持，可用于 AI 计算的全功能开发，适用于需要低能耗和高计算性能的应用程序，支持常见 API 并获得整个开发工具链的支持，在计算机视觉方面的高性能、低功耗计算使其成为计算密集型嵌入式项目的理想平台<sup>[27]</sup>。

对上述三款嵌入式设备，AVT31 为开发机载视频目标跟踪应用的专用模块，性能已得到验证并投入市场使用；Jetson 模块也已被开发人员搭载在无人机设备上目标跟踪的应用研究；而以 VPM6467 为代表的达芬奇系统更多的应用于常规视频处理，因此只对两类对适用于机载环境的嵌入式设备进行了对比分析，表 1-1 为其在性能方面的对比分析情况。

表 1-1 两类嵌入式系统性能分析表

	AVT31	Jetson TX1
重量	500g	200g
体积	260mm x 115mm	170mm x 170mm
处理速度	16G FLOPS	1T FLOPS
功耗峰值	300w	10w

由表可得 TX1 不仅具有体积小、重量轻的特点，且数据处理速度优异，功耗低。TX1 每秒可进行 1T 的浮点运算，而功耗峰值只有 10w，有着明显的性能优势，可大幅度提高应用程序的处理速度。并且小型化飞行平台，如无人机的搭载重量通常在 2kg 左右，嵌入式 GPU 平台完全可以满足要求。因此，课题选用嵌入式 GPU 平台进行目标跟踪器的研制。

### 1.2.2 目标跟踪关键技术研究现状

通过对机载应用场景及目标跟踪系统的分析，机载跟踪需要具备一个完整的处理过程，如图 1-3 所示。其关键技术主要包括视频采集、电子稳像、检测及跟踪四个方面<sup>[28]</sup>。

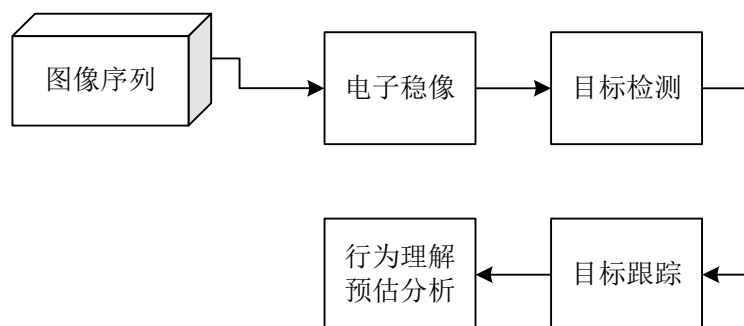


图 1-3 机载下视场景运动目标跟踪过程及关键技术示意图

通过视频采集设备获取视频数据，对应用场景进行稳像处理以弥补平台抖动对跟踪造成的不利影响，通过目标检测确定初始目标后进行跟踪处理，获取运行轨迹等信息，跟踪过程通常与目标检测结合以保证跟踪准确度。

机载下视情况跟踪运动目标面临的首要问题是平台自身运动导致图像信息模糊而造成图像质量下降，从而影响跟踪准确性。电子稳像通过信号处理确定图像间的运动矢量并进行补偿，达到稳定图像的目的<sup>[29]</sup>。实现方法大致分为两类，一类是基于像素灰度的稳像方法，如块匹配法，通过在相邻图像块间进行匹配并计算与距离最近区域的位置关系获得运动矢量<sup>[28]</sup>。该方法原理简单，精确度高，缺点是运算量较大，会降低图像的处理速度。对此已提出了互相关的改进方法，通过卷积计算两幅图像间的关系，利用卷积极值确定运动矢量，抗噪性能优异且具有较高的探测精度。另一类是寻找相同图像特征的稳像方法，通过提取两幅图像的边缘等相关特征并利用其对应关系求出变换矩阵，从而获得运动矢量并进行补偿<sup>[28]</sup>。使用较为广泛的是基于尺度特征的 SIFT(Scale-invariant Feature Transform)方法，该方法可很好地适应图像尺度变



化和旋转等情况，具有较强的稳定性，且具有独特性、多量性、可扩展性等特点。2006 年 Bay 等人在此基础上提出了 SURT(Speed-up Robust Features)方法<sup>[30]</sup>，在保持 SIFT 算法优良特性的同时，提高了计算效率。

运动目标检测技术根据成像平台与应用场景是否存在相对运动分为静态背景的运动目标检测和动态背景的运动目标检测两类<sup>[31]</sup>。静态场景指理想情况背景未发生运动与改变，只有目标在运动的情况。而机载下视场景属于动态场景，该场景下目标运动与背景的运动变化同时存在，二者混合会造成目标的形状、尺寸等特征及光照强度都会发生变化；同时运动目标经常存在被建筑或树木等障碍物遮挡或与多个动态目标共同运动产生检测歧义的情况。因此，机载下视场景中目标检测技术的复杂度要远高于静态场景下的目标检测。目前，静态目标检测技术已较为成熟，帧差法、光流法、背景差法都已得到广泛应用的目标检测方法<sup>[32]</sup>。表 1-2 根据三种方法的特点进行了对比与总结。

表 1-2 常用目标检测方法对比

	光流法	背景差法	帧差法
检测原理	光流特性随时间变化	当前帧与背景模型差分	连续帧间差分
检测效果	较完整的前景目标区域	较完整的前景目标区域	目标会有不连续空洞
环境鲁棒性	易受环境干扰	易受环境干扰	环境适应力较强
复杂程度	复杂	一般	简单
实时性	差	好	好
适用范围	成像设备固定、运动均可	成像设备固定	成像设备固定

利用帧差法进行目标跟踪可满足实时检测的需求，且受环境影响较小，鲁棒性好。但当物体运动较慢时，前后帧几乎重叠，会产生检测不到物体的结果，通常与其他检测方法相结合来提高检测的准确度。如 Wei 等将帧差法与光流法结合提出了一种新的运动目标检测方法，Collins 等利用三帧差分代替两帧差分来检测目标<sup>[33]</sup>。背景差法通过当前帧与背景模型作对比来提取目标区域，从而完成目标检测的一种方法，可检测到完整的目标结果。但在实际应用中，受到复杂环境的影响，理想背景模型的建立难度较大。研究者们在此基础上提出了一系列经典算法，如 Kim 等将聚类思想与背景建模结合来区分目标与背景<sup>[34]</sup>，Stauffer 等提出自适应的混合高斯背景提取方法，可对光照的变化，树木摇摆等复杂场景建模<sup>[35]</sup>。光流法利用目标移动时的速度矢量不同于背景区域的速度矢量来检测运动目标，该方法可应用于动态场景，但计算较为复杂，



难以满足实时性需求。目前还没有较为有效的适用于动态背景下的运动目标检测技术,通常是根椐具体应用背景及特定要求对静态背景的目标检测方法进行改进以适应具体需求。

运动目标跟踪可看作连续帧间目标位置、速度等特征的匹配问题。其处理流程通常如图 1-4 所示,其中  $N$  表示初始化需要用到的视频帧数。在跟踪的初始阶段,通过人工标定或目标检测的方式定位目标,基于目标位置提取目标区域,通过统计建模或特征表达的方法对目标表观建模,提取目标特征,结合跟踪方法定位目标在当前帧中的位置。

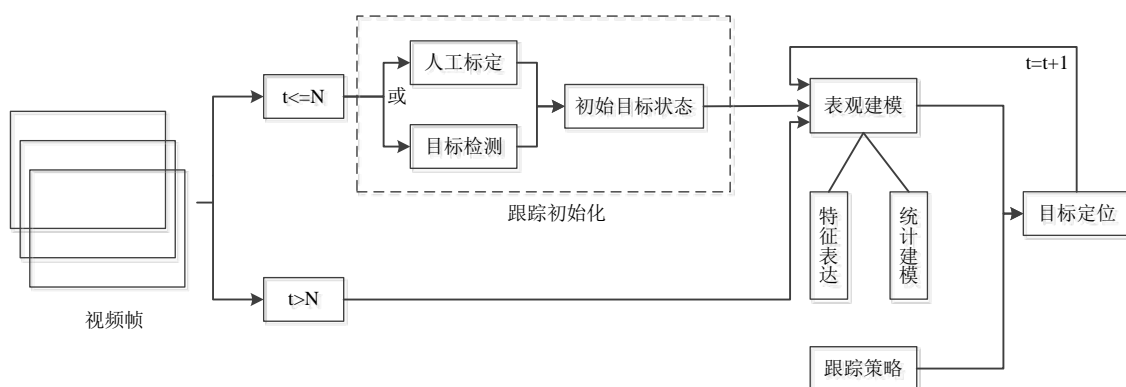


图 1-4 运动目标跟踪流程图

视频目标跟踪的解决思路主要有两种,一种是不依赖于先验知识的自底向上(Bottom-up)的跟踪方法,该方法直接从原始的图像序列提取目标运动特征并进行跟踪处理,属于数据驱动的方法<sup>[36]</sup>。能实现对目标的快速跟踪,但只在摄像机静止时的跟踪效果较。另一种是依赖于先验知识的自顶向下(Top-down)的跟踪方法,根据先验知识构建特征模型,并与图像序列进行匹配求得后验概率,从而获得目标状态,属于利用模型驱动的方法<sup>[36]</sup>。此类方法科学理论严谨、跟踪准确;但对先验知识有依赖性,若没有建模不准会对跟踪结果产生较大影响。两种处理方案各有优劣,通常将二者结合,利用先验知识进行目标的状态估计,然后利用数据驱动进行匹配搜索以实现目标的定位与建模,提高跟踪的有效性。

运动目标的表达及相似性度量方法的定义与目标跟踪的准确性和鲁棒性有直接关系,通常依据特征提取方式的不同分为基于轮廓的跟踪、基于区域的跟踪、基于特征的跟踪和基于模型的跟踪四类<sup>[37]</sup>。基于轮廓的跟踪方法通过在相邻图像中搜索并匹配目标的轮廓信息以实现跟踪功能。Michael Kass 等提出了 Snake 主动轮廓模型<sup>[38]</sup>,基于该模型的跟踪方法通过求得能量函数的最优解以调整动态轮廓,从而趋向于目标轮廓达到跟踪目的。基于特征的跟踪方

法顾名思义是利用目标的特征,通常为多个特征的集合来区分目标与背景区域,从而实现跟踪功能。搜索到与特征集匹配的区域就认为是跟踪上了运动目标。主要由特征提取和特征匹配两方面构成。基于特征的跟踪方法未对目标的图像区域进行直接处理,因此其跟踪效果受形变、光照的影响较小;并且由于可以匹配搜索部分特征,对遮挡情况具有一定的适应性。基于区域的跟踪首先通过人工或图像处理获得目标所在区域的信息并将其看作一个整体,通过在相邻图像中搜寻匹配区域以实现跟踪功能<sup>[39]</sup>。该方法鲁棒性好,但遇到遮挡则整体信息遭到破坏,就会影响跟踪性能。基于模型的跟踪方法依赖于先验知识建立的目标模型进行目标定位,并通过更新模型保证跟踪精度。该方法对目标形变的适应力强,但由于模型更新计算量大,会影响跟踪的实时性。

机载下视场景中的目标跟踪不仅视频源本身受平台运动的影响,目标往往会处于噪声、光照变化、遮挡等多种情况并存的复杂环境中<sup>[40]</sup>。研究如何提取特征建立优异的目标模型及根据实际场景选择一套好的相似性度量方法显得尤为重要。解放军信息工程大学提出利用光流技术检测并跟踪无人机平台下采集的视频目标<sup>[41]</sup>。首先利用光流技术进行全局运动补偿以消除平台运动产生的位移,然后将光流技术与目标轮廓分割相结合实现目标的检测,最后对检测到的目标利用基于区域的跟踪方法进行跟踪,得到良好的跟踪效果,但未考虑阴影、遮挡等影响因素对跟踪效果的影响。国防科技大学针对机载下视场景的运动目标检测与跟踪技术展开了相关研究,利用特征点匹配实现平台的运动估计,并基于高斯混合模型提出了背景自适应更新算法用于目标检测,最后将Kalman滤波与均值漂移算法结合实现跟踪目的<sup>[41]</sup>,利用轨迹预测改善跟踪丢失或遮挡的问题。香港理工大学研究出一种基于时空上下文学习的目标跟踪算法,实时性好、鲁棒性高、可适应目标的长时间跟踪并具备目标尺寸的比例更新功能<sup>[42]</sup>,上述工作针对机载下视或无人机场景对运动目标的检测与跟踪技术进行了深入分析与研究并得到良好的实验结果,但相关工作均以PC机为开发环境进行,目前大部分停留在算法研究层面,并未进行应用实现。因此,对跟踪速度等实时性能指标的说服力不大。

由于机载下视运动目标的跟踪技术具有重要的用途,其理论技术的研究非常重要,国内外广大研究学者都在对目标跟踪算法进行重点开发,也取得了相应的成绩。虽然由于涉及高精尖敏感信息技术,国外对该研究课题中的信息和技术的开放程度不高,但从其他领域的目标跟踪中还是可以得知,国外的目标跟踪技术已远超国内水平,在无人机目标跟踪算法研究领域拥有世界领先水平。在应用领域,欧洲的研究院有相关的算法平台,美国的私人公司甚至以无人机

的应用为主业,不仅在算法方面深入研究,也很快地将其发展目标跟踪系统,用于军方或民用领域<sup>[15]</sup>。国内学者与研究人员也在航拍系统的目标跟踪领域埋头苦干,致力于对算法的深入研究与应用<sup>[5]</sup>。但从调研来看,国内大学对目标跟踪系统的研究大部分还停留于学术研究阶段,应用较少。上海凯视力公司的视频自动跟踪模块是目前国内对目标跟踪技术一个较好的应用实现。

### 1.2.3 存在的问题

通过上述对跟踪系统及跟踪关键技术的研究现状分析可知,机载下视场景的运动目标所处环境较为复杂,影响跟踪性能的因素较多,这大大增加了跟踪难度。通过调研分析,目前该任务面临的问题主要表现为缺乏可满足实际应用的跟踪设备及完善的跟踪方案两个方面。

对机载跟踪系统的研究,早期由于无人机等机载平台的技术限制及缺乏应用需求,对复杂场景下的目标跟踪技术的研究通常是在PC机下进行的,并不能应用于实际中,可搭载在机载平台的小型化目标跟踪设备几乎没有。一些研究人员针对目标跟踪的处理要求,对嵌入式硬件平台进行了设计与实现<sup>[44]</sup>,此类研究通常利用常规跟踪方法对设计的硬件平台进行功能验证,得到的结果不能代表机载下视场景的实际跟踪性能。通过对文献及市场产品的调研发现,目前可满足实际应用的机载跟踪设备少之又少。

在技术方面,由于运动目标跟踪技术的开发难度大,算法较为复杂,对该项技术的研究主要在画质清晰、无疑似目标、无阴影、视野好等理想条件下进行,而机载下视场景由于飞行平台的运动及不可避免的抖动、成像角度及距离等因素,目标的形状姿态等表面信息、光照强度、背景等都会随时间发生变化,会对跟踪性能产生较大影响,算法的鲁棒性受到极大挑战;此外,动态背景中的运动目标经常会被周围环境遮挡,造成目标信息的不稳定甚至丢失,从而影响特征的提取与匹配策略的实施,对跟踪的准确性造成极大影响<sup>[43]</sup>。目前对严重遮挡的研究并不多,很多跟踪方法在目标丢失时通过暂停跟踪并重新选取目标来解决此问题,但在目标运动状态下进行重新选取工作显然无法准确定位目标,会对跟踪效果造成较大影响。暂时还没有发现可以较好地适应机载下视场景的跟踪技术,通常需要根据应用需求,对已有的跟踪方案进行适应性改进,以满足实际应用。

总体来看,目前缺乏可搭载在无人机等飞行设备上对地面目标进行定位的跟踪设备,且对目标跟踪技术的研究也缺乏对复杂应用场景的适应性。因此研究目标跟踪技术在嵌入式平台的实现及对应用场景的适应是十分有必要的。

### 1.3 主要研究内容

本文研究应用于机载下视场景的嵌入式目标跟踪设备。从跟踪技术角度考虑,由于当前尚没有适应于机载下视场景的理想跟踪方案,需要根据应用场景进行适当改进;针对跟踪系统可搭载在机载环境的要求,利用嵌入式平台进行目标跟踪技术的应用开发;最后,对研制的机载跟踪器需要进行综合测试与分析。因此,课题的研究内容分为三方面,首先对机载跟踪系统的研制进行总体设计并深入研究机载下视目标跟踪技术,针对机载下视常见的遮挡问题进行适应性改进,第二项研究内容为嵌入式平台上机载跟踪系统的实现;最后,为验证系统整体性能,对研制的机载跟踪系统从功能及性能方面进行综合测试分析。

针对第一项研究内容,课题针对实际应用对机载跟踪系统进行了需求分析,给出了系统性能指标并设计了机载跟踪系统研制的总体方案及应用方案。设计了遮挡处理方案并深入研究了抗遮挡目标跟踪技术,在对多类跟踪方法进行特性分析的基础上,选择高速目标跟踪技术作为系统核心技术,并从遮挡判断、运动轨迹预测与目标重捕三方面对抗遮挡问题进行了深入研究,制定了合理的解决方案。

针对第二项研究内容,课题根据机载下视目标跟踪器的技术要求,在嵌入式 GPU 平台进行了跟踪器的开发。通过 Jetpack 部署了平台开发环境,基于 CUDA 框架,利用 Eclipse 结合 OpenCV 库以交叉编译的方式实现了跟踪算法的应用开发。针对跟踪器的实时性要求,通过 CUDA 库调用平台的 GPU 资源,实现了算法加速。并基于 Qt 设计了人机交互界面,实现了对机载跟踪器的控制与显示功能。

针对第三项研究内容,课题从系统的稳定性和可靠性角度出发,对机载跟踪系统的抗遮挡跟踪、人机交互等功能及跟踪的速度、准确率等性能参数进行了全面测试并给出了详细分析,最后得出系统的综合分析结论。测试结果表明,本课题研制的机载下视目标跟踪器可对红外及可见光两种场景下的地面目标进行准确跟踪,可处理跟踪过程遇到的遮挡问题,并支持系统的人机交互;各项跟踪参数符合系统性能指标,可满足符合机载下视场景的实际应用。

本文最后对研制机载跟踪系统的过程中的主要研究工作及成果进行总结,并对机载跟踪器的后期完善工作进行了展望分析。

### 1.4 论文组织结构

本文研究在嵌入式 GPU 平台下针对机载下视的应用场景进行目标跟踪器

的研制，对论文各章节的安排如下：

第一章为绪论部分，介绍了机载下视目标跟踪器的研究背景及意义，对机载跟踪系统及目标跟踪关键技术的研究现状进行了分析，并指出当前机载跟踪系统研制存在的问题。最后对课题的主要研究内容及论文组织结构进行了概要描述。

第二章为机载跟踪系统的总体方案设计及抗遮挡跟踪技术的研究。针对实际应用对机载跟踪系统进行需求分析，给出系统性能指标并设计机载跟踪系统的总体研究框架及应用框架。深入研究了机载下视抗遮挡目标跟踪技术，在对多类跟踪方法特性进行综合分析的基础上，选择基于时空上下文的目标跟踪技术作为系统核心技术。针对机载下视的遮挡问题设计了解决方案，并从遮挡判断、轨迹预测与目标重捕三方面对遮挡问题的处理进行了技术分析。

第三章主要内容为机载跟踪系统的实现。针对机载下视目标跟踪器的技术要求，在嵌入式 GPU 平台 Jetson TX1 上进行了机载跟踪器的开发。通过 Jetpack 部署了平台开发环境，基于 CUDA 框架，利用 Eclipse 结合 OpenCV 库以交叉编译的方式实现了跟踪算法的应用开发。针对跟踪器的实时性要求，通过 CUDA 库调用平台的 GPU 资源，实现了算法加速。并基于 Qt 设计了人机交互界面，实现了对机载跟踪器的控制与显示功能。

第四章主要内容为机载跟踪系统的综合测试与分析。对课题研制的机载跟踪系统进行功能性验证及性能分析并给出综合测试分析结果。介绍系统的测试环境与测试数据集，对机载跟踪系统的抗遮挡能力、场景适应性、人机交互功能及跟踪的速度、准确率等性能分别进行测试并给出全面分析，最后得出系统的综合分析结论。

## 第2章 机载跟踪系统方案设计与抗遮挡技术研究

### 2.1 需求分析与系统性能指标

#### 2.1.1 需求分析

随着无人机技术的飞速发展,基于小型飞行平台进行高空对地探测已成为视觉领域非常活跃的研究与发展方向。利用飞行设备对地面目标进行自主跟踪在军事和民用领域都有着广阔的应用需求和重要的应用价值。

(1) 执行航空侦察任务的需求。在执行侦察任务时,常规的人员和地面侦察手段受地形地物及地球曲率对侦察视线的限制,难以对地面情况全面兼顾,且存在针对性不强、侦察人员的安全易受威胁等缺点;而卫星侦察则存在实时性差和细节缺失等不足<sup>[45]</sup>;利用机载跟踪器可以通过无人机等飞行设备进行航空侦察,从高空角度对侦察领域进行全方位多角度的信息探测,向指挥人员提供实时精确的情报信息并对特定目标执行跟踪监控、定点火力打击等任务,在现代局部战争中发挥着越来越重要的作用。

(2) 辅助飞行设备操控人员的需求。对于操控人员来说,手动控制飞行设备执行对地跟踪任务是一项枯燥且困难的工作,而这类工作在机载平台上又大量存在,因此,对目标进行自动跟踪的需求越来越迫切<sup>[46]</sup>。机载跟踪器作为对地探测辅助设备,可以将操控飞行设备的监视人员从长时间的跟踪工作中解放出来执行其他任务或掌握飞行设备运行的整体情况,以及时作出调整。当跟踪的目标出现异常行为时警告操作人员即可。

(3) 无人机执行高级识别任务的需要。机载跟踪器作为无人机的有效载荷,对其作用的发挥起着重要作用。通过机载跟踪设备对地面目标进行定位跟踪可以获得目标位置及其运动轨迹等信息,利用这些信息可进行目标行为分析或结合场景进行目标重要性分析,根据目标的运动轨迹可以分析目标的行为规律,从而预测其下一步的动作并提前做好应对措施;结合所处环境分析目标的位置及运动规律可分析目标在此场景的重要性,以判断是否为可疑目标。

(4) 处理交通异常突发事件的需求。机载目标跟踪系统可以实现空中的智能交通检测,对公路道路情况进行远程监控,通过检测道路上的交通流量及是否存在异常行为车辆,可对交通情况进行及时了解,在交通事故的预防和事后处理中有着重要应用<sup>[47]</sup>。

## 2.1.2 系统性能指标

本课题对研制的嵌入式机载跟踪系统的性能指标如下：

- (1) 对 20\*20 范围内的小目标可进行持续跟踪；
- (2) 可对 1280\*720 像素的图像进行跟踪处理；
- (3) 跟踪速度在 50 帧/秒以上；
- (4) 无遮挡场景的跟踪准确率至少为 80%；
- (5) 遮挡场景的跟踪准确率至少为 60%；

## 2.2 机载跟踪系统方案设计

### 2.2.1 机载跟踪系统总体方案设计

通过对机载跟踪系统进行功能性分析，设计的总体方案如图 2-1 所示，主要包括抗遮挡跟踪技术研究、跟踪方案在嵌入式平台的应用实现及对机载跟踪系统的综合测试与分析三部分。

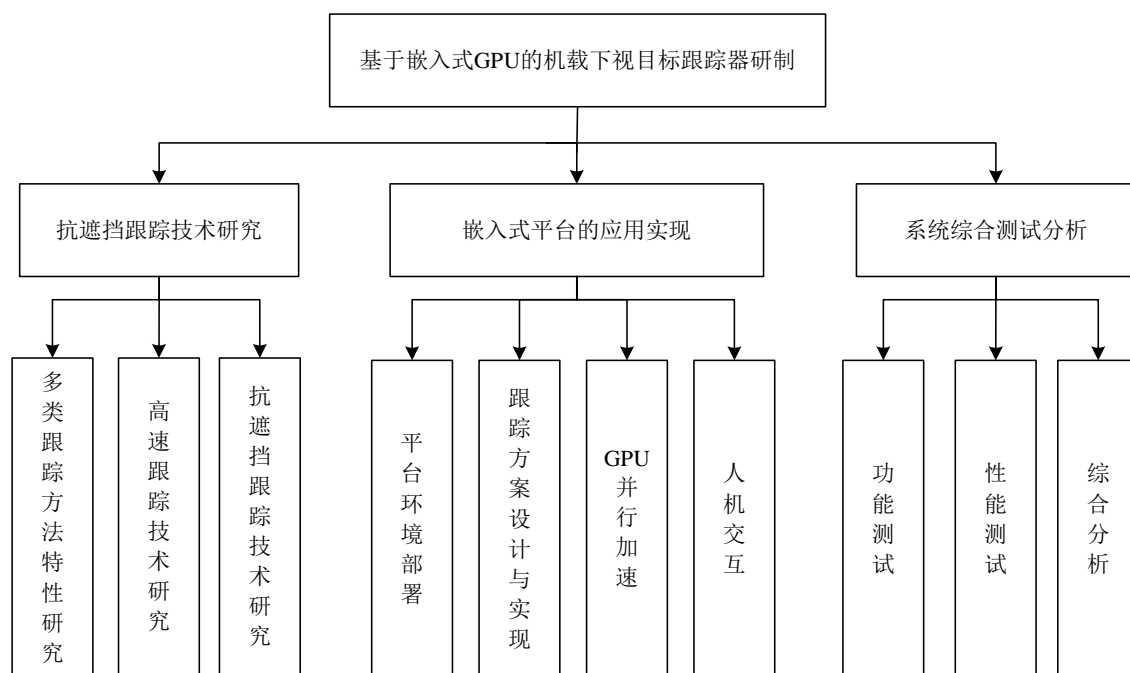


图 2-1 机载跟踪系统总体方案设计

机载下视场景下的目标经常会遇到遮挡而导致跟踪失败，为改善此类情况，课题针对抗遮挡跟踪技术进行了深入研究。目前的跟踪方法对遮挡问题几乎没有处理能力，在遮挡区域经常发生目标丢失，因此需要针对遮挡问题对跟踪方

法进行适应性改进。考虑到改进会造成算法的复杂度增大，从而对跟踪速度产生较大影响，课题对多类跟踪方法进行特性分析，从跟踪速度的角度出发，选择高速跟踪技术进行研究，并为改善遮挡问题对跟踪产生的影响，针对遮挡的各个时刻对抗遮挡跟踪技术进行了深入研究。

机载跟踪系统研制的核心是跟踪技术在嵌入式平台的应用实现。为此，首先需要对平台的开发环境进行部署，之后根据对抗遮挡跟踪技术的研究，设计抗遮挡跟踪方案并利用平台环境进行应用实现。考虑到实时性要求，利用平台的 GPU 资源对算法进行加速以提高跟踪速度。同时为了便于对机载跟踪设备进行控制，设计人机交互界面，通过界面实现对机载跟踪系统的输入及输出控制。

系统的综合测试与分析是验证机载跟踪系统总体性能的重要部分，需要从功能及性能两方面对系统进行综合测试。功能性验证主要包括抗遮挡功能、跟踪方法对红外及可见光场景的适应性、对人机交互的支持等；性能测试主要包括跟踪速度、跟踪准确率、可跟踪的最小目标尺寸等指标，并与系统性能指标进行对比，验证系统性能。

### 2.2.2 机载跟踪系统应用方案设计

机载跟踪设备的应用方案如图 2-2 所示。主要由机载跟踪系统、光电吊舱、控制设备及显示设备四部分构成。通过光电吊舱获取视频流，传入嵌入式跟踪系统进行计算处理得到目标位置信息，将结果通过显示设备显示，通过控制设备控制光电吊舱的运动，配合飞行器的飞行轨迹保证目标始终出现在可视范围内。

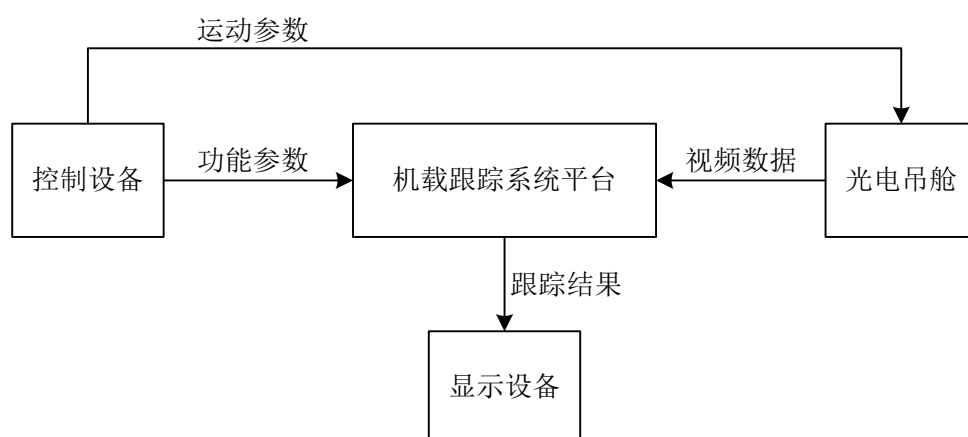


图 2-2 机载跟踪系统应用方案

机载跟踪系统根据所处环境可分为机载环境和地面基站两部分。嵌入式跟



踪平台及光电吊舱设备搭载在飞行器平台,构成机载跟踪系统,主要用于定位跟踪视频中的目标。光电吊舱等视频采集设备采集地面目标运动场景并通过视频通道传输到嵌入式跟踪平台。跟踪平台需要对流入的视频进行解析处理,然后执行相应的跟踪策略完成目标定位及结果分析,并将视频与跟踪结果整理后远程传输到地面设备进行显示。地面基站则执行对机载平台的控制及显示操作,二者通过人机交互建立联系,进行数据传输。控制设备及显示设备构成地面基站,完成对机载平台的运动控制及显示工作。控制设备一方面用来控制飞行平台的运动轨迹,根据跟踪器返回的位置信息确定目标在成像范围内的位置,当目标即将离开视野范围时,控制飞行平台进行位置移动,保证目标始终出现在成像范围内;另一方面,控制设备还可以用来控制成像设备进行适当的角度移动,可以在飞行平台不移动的情况下,通过调整成像角度使目标尽量出现在视频的中心位置,便于跟踪方案的实施。通过显示设备显示跟踪处理结果,供地面人员查看并进行后续分析处理。

## 2.3 机载下视抗遮挡跟踪技术研究

### 2.3.1 多类跟踪方法特性分析

选择能满足课题需求的跟踪方案首先需要根据性能指标分析各类跟踪方法的特性,再根据机载下视的应用情况择优选择。近年来,涌现出了众多具有优异性能的跟踪方法,根据有无检测过程的参与,可将其分为生成与判别两种跟踪方式<sup>[31]</sup>。生成式跟踪首先通过检测技术对目标进行表观建模,然后利用相应的跟踪策略进行目标定位,属于先检测后跟踪。典型的生成式跟踪方法包括基于核学习的跟踪,基于子空间学习的跟踪及基于稀疏表示的跟踪等。基于核学习的方法通常利用核密度估计进行建模并使用 **Meanshift** 方法预测目标位置。基于子空间学习的方法通过构建基函数并建立子空间,实现对目标特征进行表述。基于稀疏表示的方法通过寻找与模板重构误差最小的区域来进行目标定位,如 **L1APG** 方法<sup>[32]</sup>。判别式跟踪则是在每帧图像中都利用目标检测进行目标前景与背景的二分类,通过分类结果确定目标区域,跟踪与检测同时进行,通常包括基于在线 **Boosting**、基于 **SVM(Support Vector Machine)**,基于随机学习及基于判别式分析的方法四种<sup>[36]</sup>。**Struck** 方法、**MIL** 方法等都属于判别式的方法,由于判别式跟踪每帧都涉及目标检测,对跟踪速度会产生较大影响<sup>[39]</sup>。

在跟踪性能的评测方面,国外一些高校及视觉研究中心提供了跟踪方法的

测试平台及测试数据集，可用于对各类跟踪算法进行测评分析。如 VIVID、Benckmark 及 VOT(Visual Object Tracking)都是较为流行的跟踪测试平台。研发人员可在测试平台对自己开发的算法进行各方面的性能测试，并与各类跟踪算法作对比分析。跟踪测试平台通常会提供已标注了位置信息的公开数据集，汇总了目标跟踪的各类复杂场景，供研发人员进行下载与算法测试，为本课题中目标跟踪技术的研究提供了实验数据。其中，VIVID 平台主要关注机载场景下地面车辆的跟踪问题，提供了可见光与红外两种视频数据源，包含遮挡、刚性形变等各类跟踪场景，提供了本次课题研究的主要实验数据；Benchmark 测试平台提供的数据集则主要涉及运动场景下的行人的移动，目标的变化以非刚性形变为主，提供了灰度及可见光图像序列；VOT 是目标跟踪领域的一个重要竞赛，挑战复杂场景下的运动目标跟踪问题，提供的数据集包含遮挡、相机运动、小目标、黑夜等各类复杂背景<sup>[37]</sup>。

由于各跟踪测试平台的评价标准与性能指标都不相同，在对算法进行测试分析时，要注意在同一平台下进行。本课题在对各类跟踪方法进行测评分析时，使用 VOT2015 的公开数据集并在其提供的测试平台对近年来性能优异的跟踪方法在准确度、平均失败率、图像重叠率及跟踪速度四个方面的性能进行了测试分析<sup>[41]</sup>。其中准确度定义为通过算法预测的目标框与真实目标框的位置接近程度；平均失败率定义为跟踪器在跟踪期间丢失目标的次数；图像重叠率定义为每帧图像中预测框与真实框重叠的平均概率。跟踪速度定义为每秒能处理的图像帧数。根据上述标准，得到性能测试结果如表 2-1 所示。

表 2-1 多类跟踪方法特性分析表

跟踪器	准确度	平均失败率 (次数)	平均覆盖率	跟踪速度 (帧/秒)	发布时间
MDNet	0.60	0.69	0.38	0.87	2015
OAB	0.45	4.19	0.13	8.00	2014
Struck	0.47	1.61	0.25	2.44	2014
STC	0.40	3.75	0.12	16.00	2013
CT	0.39	4.09	0.11	12.90	2012
L1APG	0.47	4.65	0.13	1.51	2012
MIL	0.42	3.11	0.17	5.99	2011
IVT	0.44	4.33	0.12	8.38	2008

实时性和准确性是跟踪技术的两个重要指标<sup>[49]</sup>。为适应应用需求，课题在选择跟踪方法时从跟踪速度出发，选择高速跟踪方法。机载下视场景的目标跟踪技术要保证准确性，需要通过改进算法尽量消除飞行运动、地面环境等因

素对跟踪造成的不利影响，而算法的改进会造成跟踪处理的复杂度急剧增加，从而对跟踪速度产生较大影响。因此，在选取跟踪方法时应尽量选择高速跟踪算法，在通过一系列复杂处理对遮挡问题进行适应性改进后，对跟踪的实时性产生的影响较小，仍能保证对目标进行实时定位。同时，跟踪算法在保证速度与准确度的前提下，最好能对光照、遮挡等情况有一定的适应性，以适应机载下视场景的应用需求。

通过对表格的分析可以看出，最新发布的 MDNet(Multi-Domain Networks)方法<sup>[52]</sup>在准确度方面明显高于其他跟踪器，但其跟踪速度是最慢的。原因在于该方法利用卷积神经网络(Convolutional Neural Network, CNN)提取目标特征用于跟踪处理，提高了准确度；但其庞大的网络架构及大量的参数运算影响了跟踪速度，造成算法难以满足实时性要求。而基于时空上下文的 STC(Spatio-Temporal Context)方法<sup>[42]</sup>以高速跟踪的特点在众多算法中脱颖而出，且此类算法在跟踪的准确度方面与其他算法大致相当。算法的仿真验证表明，此种跟踪方法可对机载下视场景的目标进行有效跟踪，并对遮挡问题有一定的鲁棒性，可处理局部遮挡的问题。因此，选择 STC 方法作为课题研究的跟踪方法，并针对遮挡问题对算法进行适应性改进。

### 2.3.2 基于时空上下文的高速跟踪技术研究

基于时空上下文的目标跟踪技术通过条件概率公式建立跟踪目标所在位置与其周围环境特征之间的关系，将跟踪过程转化为一系列寻找置信图极值的过程。

要理解这种跟踪思想，首先需要了解局部上下文的定义。视觉跟踪领域中的局部上下文定义为目标及其周围特定区域的实时环境。连续帧间的局部上下文具有很强的时空关系。在时间方面，由于图像序列间隔足够小，相邻帧间的变化也很小，局部上下文基本保持不变，因此前后帧图像具有很强的相关性；在空间方面，图像中的目标与周围环境必然存在着某些联系，将这种目标与周围环境的关系加入目标的跟踪定位比单独利用目标信息进行跟踪要容易得多，且能提高目标跟踪的准确性。算法流程如图 2-3 所示，首先，根据场景的时间相关性，利用反卷积在目标和周边环境间建立一个空间上下文模型。然后，将此模型用于更新下一帧的时空上下文模型。下一帧中的跟踪是利用卷积计算置信图，整合了稠密时空上下文信息并且最佳目标位置用置信图的最大值来估计。最后，基于估计的置信图，提出了明确的比例适应方案，呈现出一种高且精确度高的跟踪结果。

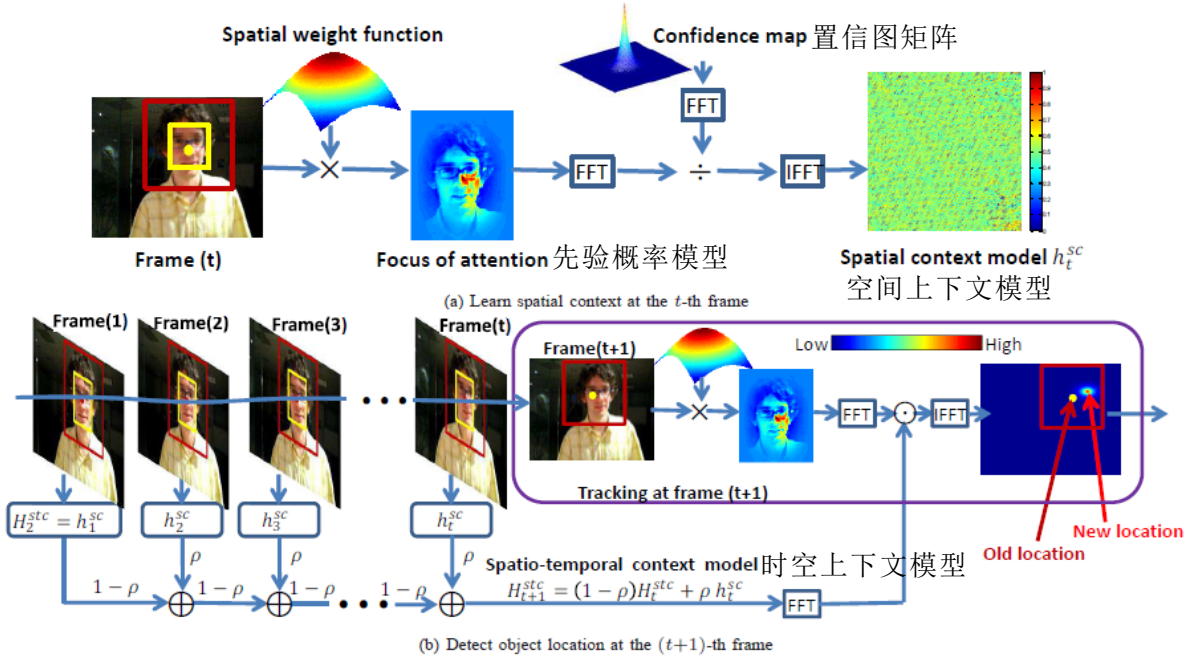


图 2-3 基于时空上下文学习的目标跟踪方法流程图

利用时空上下文进行目标跟踪的核心思想是通过计算目标的概率模型来预测目标位置。对各个概率事件的建模尤为关键。首先定义局部上下文特征集为式(2-1)，包括目标的低级外观特征和位置信息。

$$X^c = \left\{ c(z) = (I(z), z) \mid z \in \Omega_c(x^*) \right\} \quad (2-1)$$

其中， $I(z)$  表示位置  $z$  处像素的灰度值， $\Omega_c(x^*)$  是围绕  $x^*$  的上下文区域。其表征的上下文区域如图 2-4 所示。红框内的区域是稠密局部上下文区域，包含了在黄色矩形区域内以  $x^*$  作为中心的目标区域，面积为目标区域的二倍。

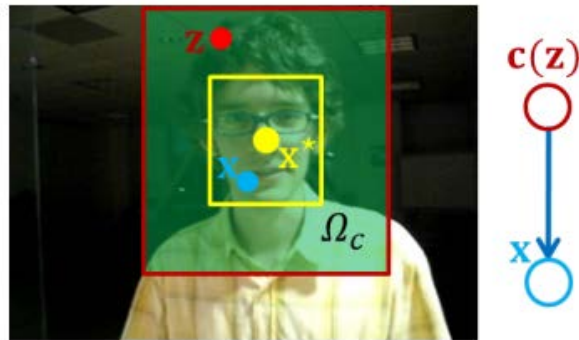


图 2-4 目标与上下文示意图

该跟踪方法是基于贝叶斯概率进行目标定位的，估计目标位置的置信图函

数主要由概率函数组成<sup>[38]</sup>，定义为：

$$\begin{aligned} m(x) &= P(x|o) \\ &= \sum_{c(z) \in X^c} P(x, c(z)|o) \\ &= \sum_{c(z) \in X^c} P(x|c(z), o) P(c(z)|o) \end{aligned} \quad (2-2)$$

其中， $P(x|c(z), o)$ 建立了目标与其上下文的空间关系，有利于改善不同退化程度的图像模糊问题。 $P(c(z)|o)$ 是上下文先验概率，表示局部上下文的外观特征。该算法的重点是在贝叶斯框架下建立空间上下文模型，学习不同像素间的相对空间关系。空间上下文模型的公式定义为：

$$P(x|c(z), o) = h^{sc}(x - z) \quad (2-3)$$

$h^{sc}(x - z)$ 是表示目标位置  $x$  与其局部上下文中某一位置  $z$  的相对距离与方向的函数，它是一个非径向对称函数，即使存在多个疑似位置与真实目标位置距离相同，但与目标的相对位置关系是不同的，空间关系也就不同，这有助于解决相似物体出现时的歧义问题<sup>[38]</sup>。

上下文先验概率模型对目标上下文区域进行建模。利用上下文的灰度信息表征其外观特征，加权函数表征上下文各点对目标特征提取的权值贡献。距离目标中心越近的点，在提取目标特征时赋予的权重自然就大；若距离较远，则与目标关系较小，在进行特征提取时被赋予的权重自然会随之减小。先验概率通过对上下文区域像素赋予不同权重获得，可建模为：

$$P(c(z)|o) = I(z) \omega_{\sigma}(z - x^*) \quad (2-4)$$

其中， $x$  代表目标位置， $o$  表示跟踪对象。 $I(\cdot)$  代表灰度信息，表示上下文的外观特征， $w_{\delta}(\cdot)$  是一个高斯加权函数，定义为：

$$w_{\delta}(z - x^*) = ae^{-\frac{(z - x^*)^2}{\delta^2}} \quad (2-5)$$

其中， $a$  是将先验概率进行归一化的常数。 $\delta$  是比例参数。(2-5)式基于生物视觉系统的“Focus of attention”进行建模，将权重集中于需要详细分析的区域， $z$  距离目标  $x^*$  越近，对预测下一帧中目标位置的贡献度就越大，则该位置就被赋予更大权重。

将先验概率模型与空间上下文模型结合，得到的置信图模型化简为式(2-6)， $\otimes$  代表卷积。

$$\begin{aligned} c(x) &= \sum_{z \in \Omega_c(x^*)} h^{sc}(x-z) I(z) w_\delta(z-x^*) \\ &= h^{sc}(x) \otimes (I(x) w_\delta(x-x^*)) \end{aligned} \quad (2-6)$$

将公式(2-6)转换到频域，利用快速傅里叶变换进行卷积得：

$$F(c(x)) = F(h^{sc}(x)) \times F(I(x) w_\delta(x-x^*)) \quad (2-7)$$

其中， $F(\cdot)$ 代表傅里叶变换，由此可以求得空间上下文模型为：

$$h^{sc}(x) = F^{-1} \left( \frac{F(c(x))}{F(I(x) w_\delta(x-x^*))} \right) \quad (2-8)$$

其中， $F^{-1}(\cdot)$ 代表傅里叶反变换。第一帧的目标位置已通过手动或自动检测被初始化，利用第 $t$ 帧的空间上下文模型去更新时空上下文模型，如式(2-9)所示，可以减少由于目标外观特征变化而产生的噪声。时空模型与先验模型在频域相乘即得到目标在图像中各个位置出现概率的置信图矩阵，如式(2-10)。

$$H_{t+1}^{stc} = (1-\rho) H_t^{stc} + \rho h_t^{sc} \quad (2-9)$$

$$c_{t+1}(x) = H_{t+1}^{stc}(x) \otimes (I_{t+1}(x) w_{\sigma_t}(x-x_t^*)) \quad (2-10)$$

新目标的位置则由置信图的极大值确定，如式(2-11)所示，由此利用目标的上下文信息确定了目标在第 $t+1$ 帧中的位置。

$$x_{t+1}^* = \arg \max_{x \in \Omega_c(x_t^*)} c_{t+1}(x) \quad (2-11)$$

最后，采用如式(2-12)所示的比例更新方案来对目标在当前帧中的尺寸进行修正，减小目标大小的变化对跟踪产生的影响。

$$\begin{cases} s_t' = \sqrt{\frac{c_t(x_t^*)}{c_{t-1}(x_{t-1}^*)}} \\ \bar{s}_t = \frac{1}{n} \sum_{i=1}^n s_{t-i}' \\ s_{t+1} = (1-\lambda) s_t + \lambda \bar{s}_t \\ \sigma_{t+1} = s_t \sigma_t \end{cases} \quad (2-12)$$

基于时空上下文的跟踪方法不仅利用了连续帧间的目标关系，也利用了目标与周围环境的空间关系来完成目标的定位过程，对提高跟踪准确性具有重要意义；利用傅里叶变换进行模型的确立与更新，在保证准确性的同时，极大地提高了处理速度。

利用基于时空上下文的跟踪方法对地面目标进行跟踪定位的效果如图 2-5、2-6 所示。由图可以看出，在无遮挡时，跟踪方案可以对目标进行有效定位，并对目标的形变等有一定的适应性，鲁棒性较好。当遇到遮挡时，由于此种方案充分利用了目标的空间上下文建立了目标与周围环境的空间关系，对环境由一定的依赖性，在目标不发生移动而遮挡物进入目标区域时，可以有效的通过不变的上下文锁定目标区域。而当目标运动进入遮挡区域时，会因为上下文信息锁定遮挡物，造成目标继续运动而跟踪框将不再跟随目标移动的情况。为解决遮挡问题造成的跟踪失败，需要研究抗遮挡技术并与跟踪方案融合。

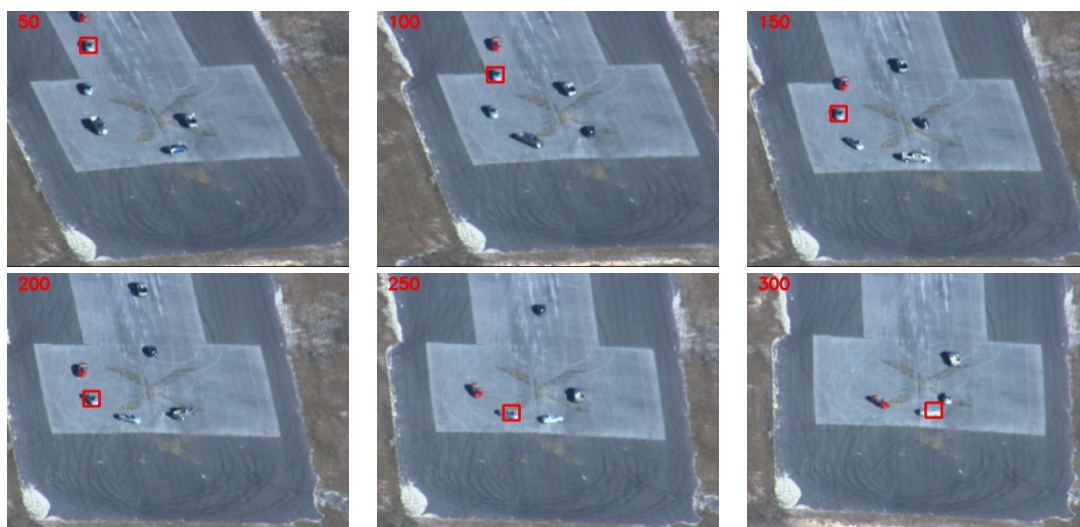


图 2-5 无遮挡情况的目标跟踪效果



图 2-6 有遮挡情况的目标跟踪效果



### 2.3.3 抗遮挡跟踪技术研究

机载下视情况下的目标经常由于遮挡、形变、光照强度变化、机载平台抖动等情况而产生跟踪出错的情况,而目前还没有有效的方案可以解决此类问题,需要分析具体影响因素并根据应用场景设计改进方案,通过算法提高跟踪准确度。利用机载下视图像序列对各类跟踪方法进行实验验证,发现遮挡问题是影响跟踪性能的重要因素。机载下视情况下的目标跟踪经常会受到建筑、树木等物体的遮挡而导致跟踪出错,对此,课题分析了遮挡导致目标跟踪失败的原因并设计了相应的解决方案,以改善遮挡对目标跟踪造成的影响。

通过研究发现,遮挡阶段目标信息的不足是导致跟踪失败的根本因素。利用充足特征信息对目标进行准确表述是保证跟踪性能的关键,而跟踪过程中,树木等障碍物对目标造成的遮挡会导致目标的有效区域变小,从而使目标的信息量减少,影响跟踪的准确性。目标信息在经过遮挡区域的过程中呈现抛物线规律变化,在进入遮挡区域的过程中,目标信息逐渐减少;目标完全进入遮挡区域后的信息量降为零,在离开遮挡区域的过程中目标信息逐渐恢复。

目标行进过程中遇到遮挡会对跟踪结果产生较大影响,多数情况下会造成跟踪目标丢失,从而导致跟踪失败的情况。为避免遮挡区域对跟踪产生影响,应准确判断目标何时进入遮挡区域并制定相应的处理措施。因此,判断遮挡的发生时刻尤为重要。根据目标经过遮挡区域的过程可知,由非遮挡区进入遮挡区时目标信息会产生较大变化。可以将发生这种变化的时刻作为遮挡的发生时刻。在未发生遮挡时,进行常规跟踪处理;判断遮挡发生后,利用当前目标信息结合遮挡前的目标信息估计目标的运动轨迹,直至目标离开遮挡区域,对离开遮挡区域的目标要及时捕获并继续进行跟踪定位。

目标被遮挡的过程可以分为三个时刻,如图 2-7 所示。



图 2-7 目标被遮挡的三个时刻

图 2-7(a)所示是目标将要被遮挡的时刻,该过程目标信息完整;图 2-7(b)



所示是目标处于遮挡中，此时目标信息逐渐丢失；图 2-7(c)所示是目标被完全遮挡，此时目标信息完全消失。不同时刻目标的信息量是不一样的，判断遮挡发生的时刻是处理遮挡问题的关键。只有正确的预测到障碍物，才能对遮挡区域的跟踪进行及时处理，避免遮挡对跟踪造成的不利影响。

课题对完全遮挡问题进行了深入研究，分析了目标经过遮挡区域时信息量的变化，设计的抗遮挡处理流程如图 2-8 所示，主要由遮挡判断、运动轨迹预测及目标的重新捕获三部分构成。

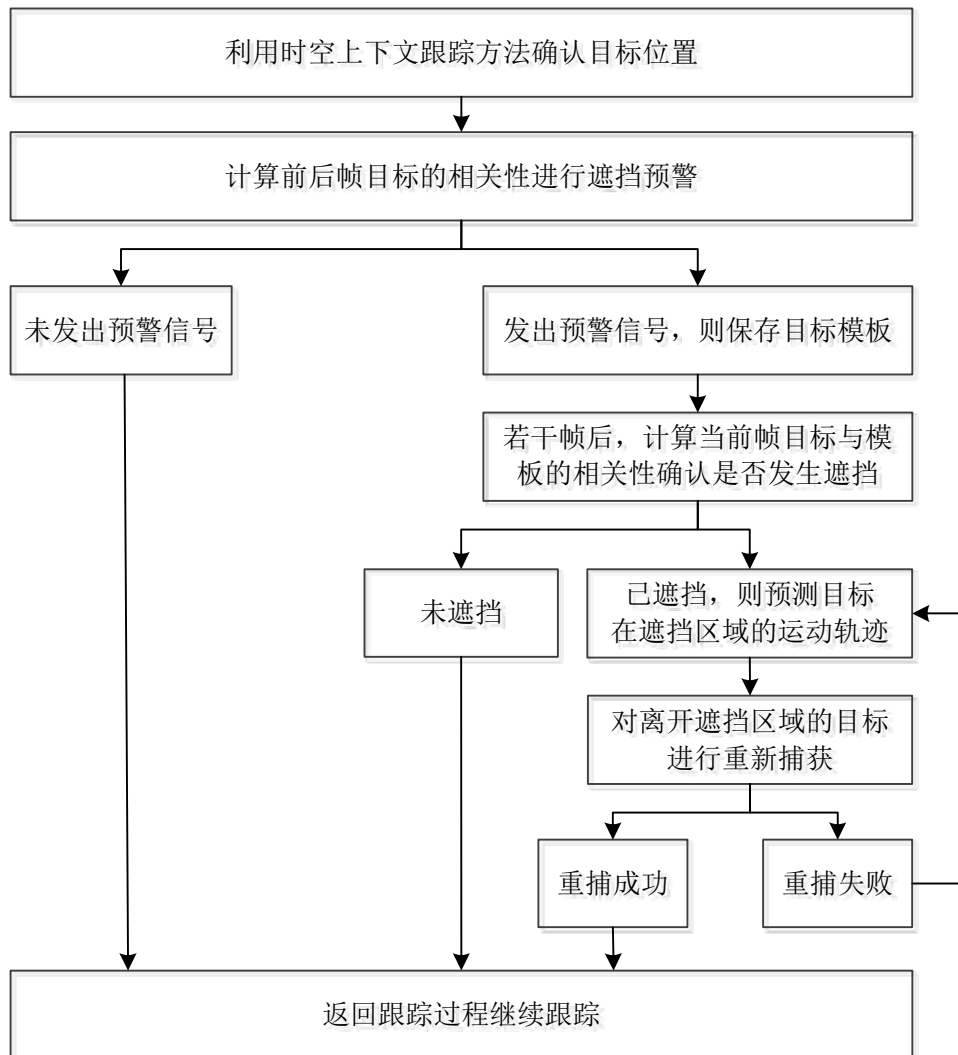


图 2-8 抗遮挡跟踪流程

在抗遮挡处理中，利用前后帧图像中目标的相关性来判断是否遇到遮挡。目标信息的差异可由目标的相关性表示，在无遮挡区域进行跟踪时，由于前后帧图像的差异非常小，目标的相关性系数很高；而在遮挡区域进行跟踪时，由

于目标被逐渐遮挡，相邻两帧的目标区域会发生显著变化，其相关性也会随之发生较大变化。因此将目标的相关性是否会发生显著变化作为判断目标遮挡的依据。

由于阴影、光照强度的变化等会对目标信息的变化产生影响，为尽量排除此类情况，提高判断的准确性，目标的遮挡判断过程又分为遮挡预警和遮挡确认两个过程。通过两次检测目标状态确认是否遇到遮挡。在跟踪过程中，计算相邻图像中目标的相关性，若目标相关性足够大，则说明目标未发生明显变化，保持跟踪过程；若目标间的相关性不足，则说明目标的变化较大，判断可能是遇到了遮挡，发出遮挡预警信号并保存目标区域作模板；继续运行若干帧后，利用当前帧的目标与保存模板的相关性判断是否发生遮挡，若是由于光照、阴影等造成的误判则继续跟踪；若确实丢失，则进行运动轨迹的预测，并在目标离开遮挡区域后，利用模板对其进行重新捕获，若捕获成功，则返回跟踪程序继续跟踪过程；若捕获到的目标与模板相关性不足，则说明目标仍处于遮挡区域，重复目标的轨迹预测与捕获过程，直到捕获成功。在遮挡预警时，利用目标上下文区域的相关变化率作为遮挡预警指标。由于目标的上下文区域是包含目标在内的周围区域，可对目标状态发生变化进行提前预测，因此利用目标上下文区域进行遮挡预警，使得检测到异常时目标信息依然完整。

通过对抗遮挡跟踪技术的研究，上述跟踪方案可以准确判断遮挡的发生时刻，将遮挡区域与非遮挡区分开处理，避免了遮挡区域对正常跟踪产生的影响，对处于遮挡区域的目标，利用遮挡时剩余的目标信息及遮挡发生前的目标信息估计其在遮挡区域的运动轨迹，在目标离开遮挡区域后利用模板匹配及时捕获，对改善了遮挡对跟踪过程产生的影响。

本文研究了利用相关系数及感知哈希指纹两种方法来判断前后帧图像中目标的相关性，从而判断遮挡时刻的发生。

利用相关系数判断遮挡的发生需要计算目标的相关系数矩阵，如式(2-13)：

$$\text{corr}(\mathbf{I}m_t, \mathbf{I}m_{t-1}) = \frac{\sum_{i=1}^n (\mathbf{I}m_{t_i} - \bar{\mathbf{I}m}_{t_i})(\mathbf{I}m_{t-1_i} - \bar{\mathbf{I}m}_{t-1_i})}{\sqrt{\sum_{i=1}^n (\mathbf{I}m_{t_i} - \bar{\mathbf{I}m}_{t_i})^2 (\mathbf{I}m_{t-1_i} - \bar{\mathbf{I}m}_{t-1_i})^2}} \quad (2-13)$$

其中， $\mathbf{I}m_t$  表示第  $t$  帧图像， $\bar{\mathbf{I}m}_t$  表示第  $t$  帧图像的均值。相关矩阵由相邻两幅图像的协方差与标准差的乘积的比值获得。取相关矩阵的最大值并进行归一化处理，得到目标在第  $t$  帧的最大相关系数：

$$\max corr_t = \frac{\max(corr(\text{Im}_t, \text{Im}_{t-1}))}{\text{norm}(\text{Im}_t, 2) \times \text{norm}(\text{Im}_{t-1}, 2)} \quad (2-14)$$

利用两组有遮挡的机载图像序列进行实验验证,得到目标的最大相关系数随图像帧数的变化如图 2-9 所示,其中,空缺部分表示目标被完全遮挡的时段。

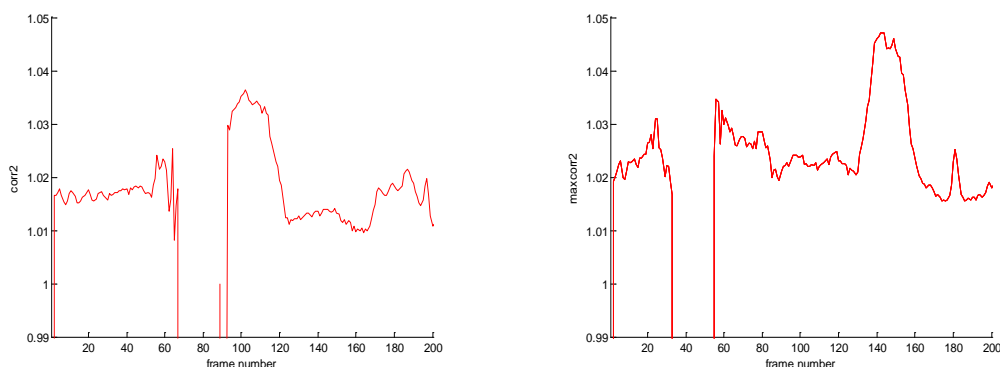


图 2-9 目标的最大相关系数随图像帧数的变化

从图中可以看出,在遮挡发生前,前后帧目标的最大相关系数与正常跟踪情况的差别较小,造成判断遮挡发生时刻的难度增大,从而无法对遮挡的发生进行准确判断。因此,课题利用目标最大相关系数的比值,定义为相关变化率,来判断遮挡时刻的发生。相关变化率为前后两帧图像中目标的最大相关系数的较大值与较小值的比值。这样处理后的相关变化率可摆脱范围的限制,更加方便且精确地进行阈值的测量与设定。

$$CCR_t = \frac{\max(\max corr_t, \max corr_{t-1})}{\min(\max corr_t, \max corr_{t-1})} \quad (2-15)$$

利用与图 2-9 所用相同的测试序列进行实验验证,得到相关变化率随图像帧数的关系如图 2-10 所示。

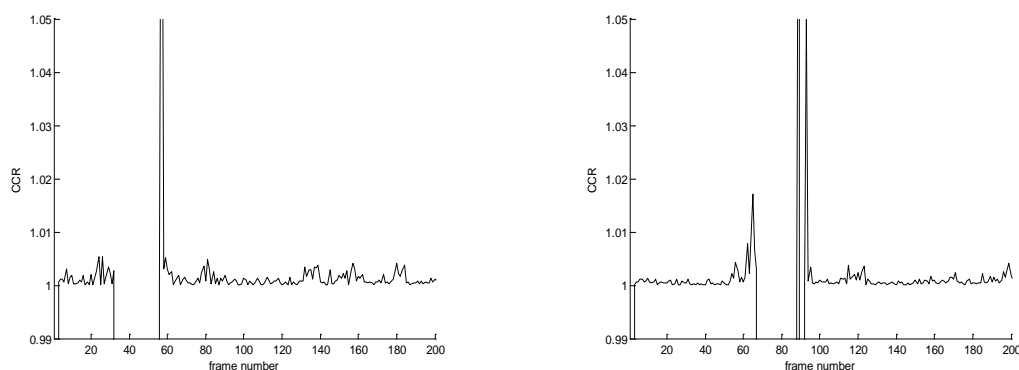


图 2-10 目标的相关变化率随图像帧数的变化

从图中可以看出，在目标将要被遮挡时，上下文区域的相关变化率会突然升高，可以此准确检测到目标发生较大变化的时刻以作为预警指标，并通过设置阈值判断是否存在遮挡预警信号。检测到遮挡预警信号时，保存目标区域的像素信息作为模板。

由于下视目标的特性变化小，在确认目标是否被遮挡时，可以根据直方图的匹配度判断其相关性。图像的灰度直方图反映了图像灰度级的分布情况及统计特性。具有图像平移、旋转、缩放不变性等特点。在利用图像的直方图进行运算时，需要通过直方图匹配将原始直方图变换为统一格式的直方图。图 2-11 表示遮挡预警时的目标区域及其直方图；由图可以看出，两个遮挡时刻的目标直方图的差异是很大的，可以作为判断目标相关性的依据。

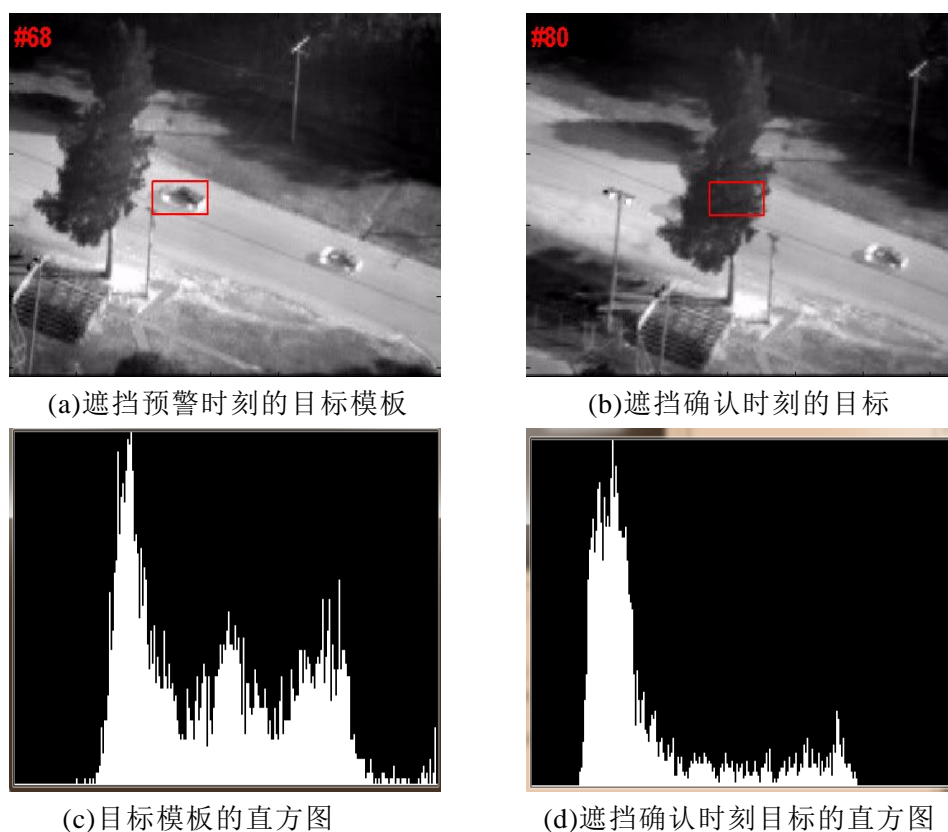


图 2-11 遮挡时刻的目标及其直方图示意图

作为对前方是否遇到遮挡区域的二次确认，需要准确判断目标状态，因此采用目标区域作遮挡确认。在预警信号发出后继续运行 10 帧，计算模板与当前目标区域的直方图匹配度：

$$\max corr_{t,q} = \frac{\max \left( corr \left( hist \left( ImTar_t \right), hist \left( ImTar_{t+10} \right) \right) \right)}{norm \left( hist \left( ImTar_t \right), 2 \right) \times norm \left( hist \left( ImTar_{t+10} \right), 2 \right)} \quad (2-16)$$

其中,  $\text{ImTar}_i$  表示目标区域,  $\text{hist}$  函数表示对目标区域进行提取直方图信息操作, 通过计算模板与当前目标直方图的相关系数的最大值并进行归一化处理, 作为丢失确认的评判指标。将得到的匹配度与经验匹配度 70% 做比较, 大于 70% 证明目标与模板的匹配度较高, 判定目标没有丢失; 若小于 70%, 则说明目标与模板存在较大差异, 判定目标丢失。

感知哈希算法(Perceptual Hash Algorithm, PHA)通过比较两幅图像的指纹来判断图像的相关性, 通过提取图像特征每幅图像都可以生成一个特定的指纹字符串, 不同的图像会产生不同的指纹结果<sup>[54]</sup>。图像中的信息有高低频之分, 高频信息包含图像灰度变化多且明显的部分, 用来表征图像细节; 低频信息显示图像灰度变化小的区域, 描述图像大范围内的信息。图像中均有低频信息, 但若图像模糊则可能高频信息不足, 因此感知哈希算法利用图像的低频信息来生成指纹字符串<sup>[55]</sup>。

图 2-12 是机载下视场景的示意图及其离散余弦变换系数矩阵图。系数矩阵图从左上角到右下角表示图像由低频到高频的信息, 由图 2-12(b)可以看出, 离散余弦变换系数在图像的低频区数值较大, 高频区数值较小, 图像的能量集中于低频区域, 其离散余弦变换系数可以表示目标特征。

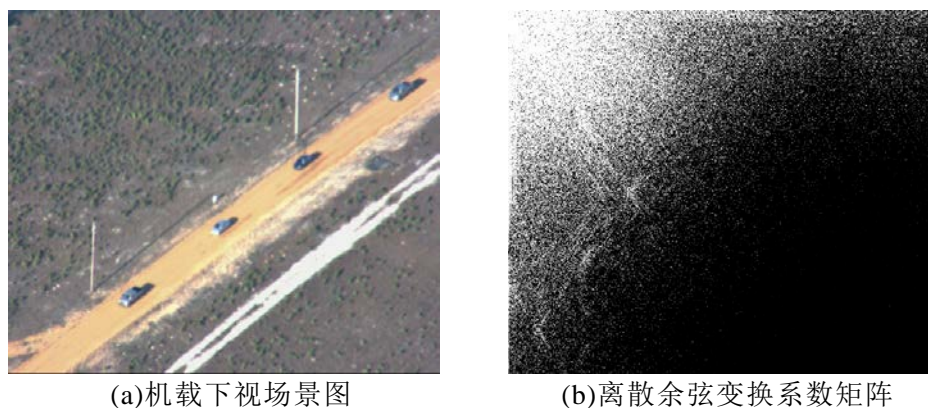


图 2-12 机载下视场景图及离散余弦变换系数矩阵

感知哈希算法通过图像缩放将尺寸缩小到  $8 \times 8$  共 64 个像素来去除高频细节, 并通过减少灰度级的方式进一步减少计算量以提高处理速度, 通过将各像素点与哈希均值进行比较, 生成指纹字符串, 通过比较两幅图像的指纹字符串来判断相似性<sup>[54]</sup>。感知哈希算法分为均值哈希算法和增强哈希算法两种。均值感知哈希算法利用图像的灰度均值来提取低频特征, 若对图像进行直方图均衡, 则会对灰度均值产生影响, 从而影响哈希指纹的生成<sup>[55]</sup>。因此, 本课题选用增强哈希方法生成哈希指纹, 该方法利用离散余弦变换, 如式(2-17), 将

图像信息由时域变换到频域，通过图像压缩来提取低频信息。

$$F(u, v) = \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad (2-17)$$

利用该方法进行图像间相关性的对比时，对图像的纵横比并没有硬性要求，可随尺寸的改变发生变化，只需确保将图像变成  $8*8$  的正方形矩阵，这样就可以不考虑尺寸、比例等因素造成的图像差异，而对任意大小的图像进行相关性的对比<sup>[45]</sup>。为减小计算量，通过将小图片转换为 64 灰度级的灰度图像进行色彩的简化处理。通过计算小图片中的像素灰度或相关系数的均值作为哈希指数的判断标准，将每个像素点的值与像素均值作对比进行二分类。将得到的分类结果按顺序组合在一起，就构成了 64 位的哈希指纹。在比较相似性时，首先计算两幅图像的哈希指纹，然后计算相同位置数值不同的位数，即汉明距离来判断两幅图像的相似性。若汉明距离为 0，说明 64 位指纹一一配准，表示两幅图像非常相似；若汉明距离小于设定阈值，则表示二者虽然不同，但有一定相似性；若汉明距离大于设定阈值，则二者的关系不大。该方法进行目标状态检测的优点是原理简单、计算量小、对图像的尺寸变化、亮度变化并不敏感，对机载下视场景下的视频图像处理具有很好的适应性。

利用机载下视图像序列中含有遮挡情况的图像序列进行验证，得到的目标上下文区域生成的感知哈希值随图像帧数的变化如图 2-13 所示，取感知哈希阈值为 10，判断前后帧目标的相关性。由图可以看出，有两个时刻的感知哈希值超过阈值，表明此时发出了遮挡预警信号。图 2-14 对图 2-13 中发出两次预警信号时的目标状态及 10 帧后遮挡确认时刻的目标状态分别进行了说明。

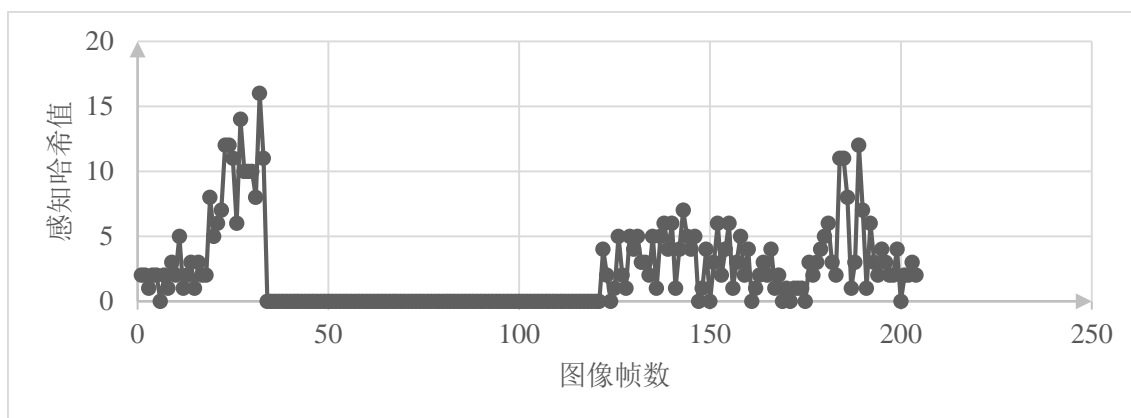


图 2-13 感知哈希值随图像帧数的变化





(a) 第 23 帧的预警及 10 帧后的目标示意图



(b) 第 182 帧预警及其 10 帧后的目标示意图

图 2-14 与图(2-13)对应的两次预警示意图

根据对图 2-13 中感知哈希值的判断（哈希距离以 10 为判据）分别在第 23 帧和第 182 帧出现了遮挡预警，对应遮挡时刻及其 10 帧后的目标情况分别如图 2-14(a)和图 2-14(b)所示，将遮挡预警与遮挡确认两个时刻的图像对比可以看出，第 23 帧的判断是正确的，而第 182 帧的判断是由于周围存在疑似目标的干扰造成的误判。通过遮挡确认环节分析目标相关性系数发现，第 23 帧出现遮挡预警时的目标与 10 帧后的目标的相关性为 0.451，二者的相关性很低说明了目标图像已发生了较大变化；而第 182 帧出现遮挡预警时的目标与 10 帧后的目标相似度则为 0.956，可以判断二者是同一物体。两种对遮挡情况的判断结果与实际时刻相对应，由此可以验证对目标遮挡判断的有效性。遮挡预警对前后帧目标关系的判断十分敏感，可以有效监测遮挡发生时刻；而通过对遮挡情况的二次确认环节可以有效地将疑似目标、背景光照变化等情况排除；二者结合可以准确判断遮挡时刻的发生，由此验证了遮挡判断机制的有效性。

由此可知，利用感知哈希值与目标相关系数结合的方法可以对目标在前进方向是否遇到遮挡进行有效判断。由于感知哈希值对目标的变化较为敏感，所以利用相关系数对目标状态进行二次判断，提高确认目标是否遇到遮挡的准确

度，排除阴影、光照变化、歧义目标等因素对目标状态判断造成的影响。

经实验验证，两种方法都能通过计算前后帧图像中目标的相关性来准确判断遮挡的发生。通过多组含遮挡情况的图像序列检测两种方法的鲁棒性时发现，利用相关系数判断遮挡时刻的发生对环境的依赖性较大，在面对可见光、红外等不同类型的场景应用时需要通过适当调整阈值参数来适应环境的变化；而利用感知哈希算法对判断遮挡发生的环境适应性较好，无需进行多次参数的调整。因此，本文选择利用感知哈希算法判断前后帧目标的相关性，从而判断遮挡时刻的发生。

在进行目标运动轨迹的预测时，由于目标完全处于遮挡区域，无法提取到有效信息，因此，必须利用遮挡前的位置信息进行目标轨迹的预测。常用卡尔曼滤波进行运动轨迹的预测，该方法利用线性最小方差估计来预测动态序列的下一个变量值<sup>[58]</sup>。经实验验证，该方法会对实时性产生较大影响，因此并未采用。考虑到机载下视场景中的目标运动速度及轨迹不会发生太大变化，将匀速直线运动作为轨迹预测的估计函数。利用目标遇到遮挡前的 5 帧图像的目标位置  $pPoint_1$ - $pPoint_5$  估计目标在遮挡区域的运动方向，如式(2-18)；并利用目标中心的平均距离计算运动速度  $(\Delta x, \Delta y)$ ，如式(2-19)、(2-20)。利用此类方法可以对直行目标的运动轨迹进行准确预测，对于后续进行目标的重新捕获具有一定的参考价值。

$$\frac{y - pPoint_1.y}{x - pPoint_1.x} = \frac{pPoint_5.y - pPoint_1.y}{pPoint_5.x - pPoint_1.x} \quad (2-18)$$

$$\Delta x = \frac{\sum_{i=1}^4 pPoint_{i+1}.x - pPoint_i.x}{4} \quad (2-19)$$

$$\Delta y = \frac{\sum_{i=1}^4 pPoint_{i+1}.y - pPoint_i.y}{4} \quad (2-20)$$

遮挡阶段对目标的运动轨迹进行预测的结果如图 2-15 所示。



图 2-15 遮挡区域的运动轨迹预测效果图



对于目标离开遮挡区域后的重新捕获技术,本文研究了边缘检测与模板匹配两种方法。

边缘检测的目的在于标识处图像中亮度变化明显的点,进而提取出有用的信息。其根本原理是通过算法将图像中各个像素点领域内数值变化大的元素进行突出显示,而变化不大的相邻元素进行屏蔽或降低其灰度显示,突出目标与背景的变化率,从而正确区分二者。基于边缘检测的目标重捕技术通过对图像进行以边缘检测为核心的一系列处理,得到疑似目标并利用疑似目标与模板的相关系数来处理图像来获得候选目标,处理的各个阶段效果图如图 2-16 所示:

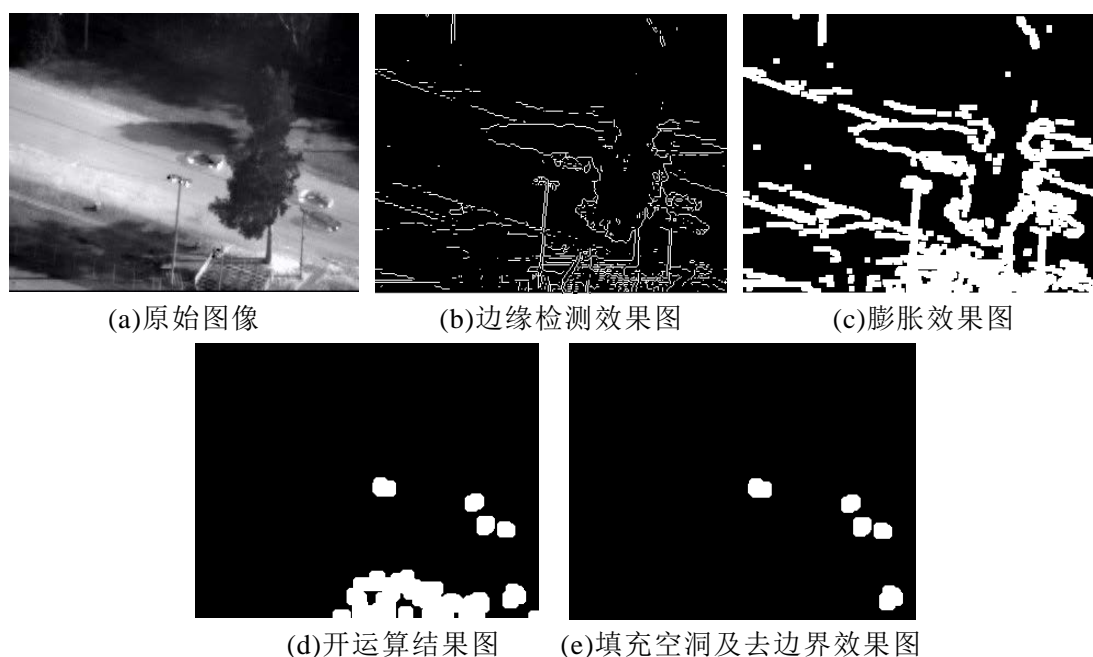


图 2-16 基于边缘检测的目标重捕过程示意图

通过几种边缘检测算子对机载下视图像的检测效果进行验证,选择利用 Sobel 算子实现目标捕获功能,测试图像如图 2-16(a)所示。捕获过程分以下几步进行:

(1) 利用 Sobel 算子进行边缘检测。检测图像中的目标可能出现的位置,使用图像边缘检测算法,使用 Sobel 算子对图像进行处理,同时使用水平和垂直算子,阈值为 0.05 (经验值)。处理结果如图 2-16(b)所示。

(2) 形态学运算,首先膨胀运算,然后做开运算。由于之前 Sobel 算子使用的阈值为经验值,图像的边缘预算结果有可能低于阈值,导致边缘断裂。膨胀运算是将与物体接触的所有背景点合并到该物体中,使边界向外部扩张的过程,希望通过膨胀运算使得目标边界形成一个封闭的区间如图 2-16(c)所示。

然而膨胀操作同时也可能导致目标和背景中的杂波相连接。我们将使用开运算来处理膨胀结果。开运算的操作是在图像中先进行腐蚀运算,再进行膨胀运算,它起到消除小物体、在纤细点处分离物体、平滑较大物体的边界,同时并不明显改变其面积的作用。处理结果如图 2-16(d)所示。

(3) 填充空洞和去除边界,首先把图像的封闭区域全部填充起来,然后去除掉和边界相连的区域。处理结果如图 2-16(e)所示,可以得到若干个区域,这些区域就是疑似目标区域。求疑似目标和保存模板的直方图匹配度,若大于 70%,则认为捕获到目标;若低于 70%,则判断未能捕获到目标,继续进行轨迹预测及目标捕获过程。

模板匹配是另一种常用的目标捕获方法。该方法在一副图像中寻找与模板图像最匹配部分,常用于在复杂场景中对目标的搜索与定位。模板匹配通过将模板图像块在参考图像中滑动进行比较来确定匹配区域<sup>[56]</sup>。模板块逐像素移动,在每个位置都进行一次度量计算来确定目标与当前区域的相关性,获得的度量值矩阵中数值最大的位置代表最高匹配,通常作为搜寻到的目标顶点<sup>[57]</sup>。根据目标顶点及模板尺寸在参考图像中准确找到目标。模板匹配的判断准则通常有平方差匹配、标准平方差匹配、相关匹配、标准相关匹配、相似匹配、标准相似匹配六种,匹配公式如式(2-21)-(2-26)所示。

平方差匹配顾名思义是利用目标与图像的平方差进行匹配,最佳匹配为 0,结果值越大,匹配越差;标准平方差匹配是对其的归一化。

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2 \quad (2-21)$$

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2 \times \sum_{x', y'} I(x + x', y + y')^2}} \quad (2-22)$$

相关匹配在模板和图像间采用乘法操作,数值越大代表匹配度越高,最坏的匹配结果为 0;同样,标准相关匹配是对相关匹配的归一化处理。

$$R(x, y) = \sum_{x', y'} (T(x', y') \times I(x + x', y + y')) \quad (2-23)$$

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') \times I(x + x', y + y'))}{\sqrt{\sum_{x', y'} T(x', y')^2 \times \sum_{x', y'} I(x + x', y + y')^2}} \quad (2-24)$$

相似匹配是将模板对其均值的相对值与图像对其均值的相对值进行匹配,-1 代表最坏的匹配结果,1 代表最佳匹配,0 代表二者没有任何相似性;标准相似匹配同样是对相似匹配的归一化处理。

$$R(x, y) = \sum_{x', y'} (T'(x', y') \times I(x + x', y + y'))$$

$$T'(x', y') = T(x', y') - \frac{1}{w \times h} \times \sum_{x'', y''} T(x'', y'') \quad (2-25)$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{w \times h} \times \sum_{x'', y''} I(x + x'', y + y'')$$

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \times I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \times \sum_{x', y'} I'(x + x', y + y')^2}} \quad (2-26)$$

对如图 2-17 所示的测试图像及其目标模板利用上述六种匹配准则分别进行实验，得到的结果如图 2-18 所示。

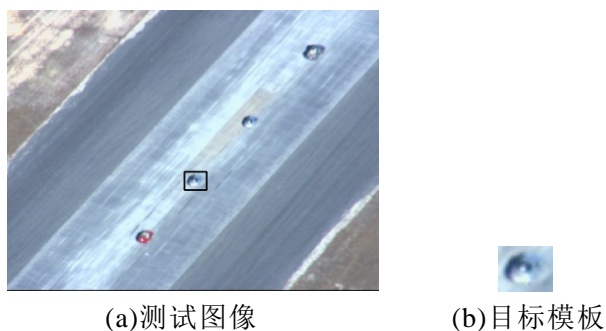


图 2-17 测试图像及目标模板

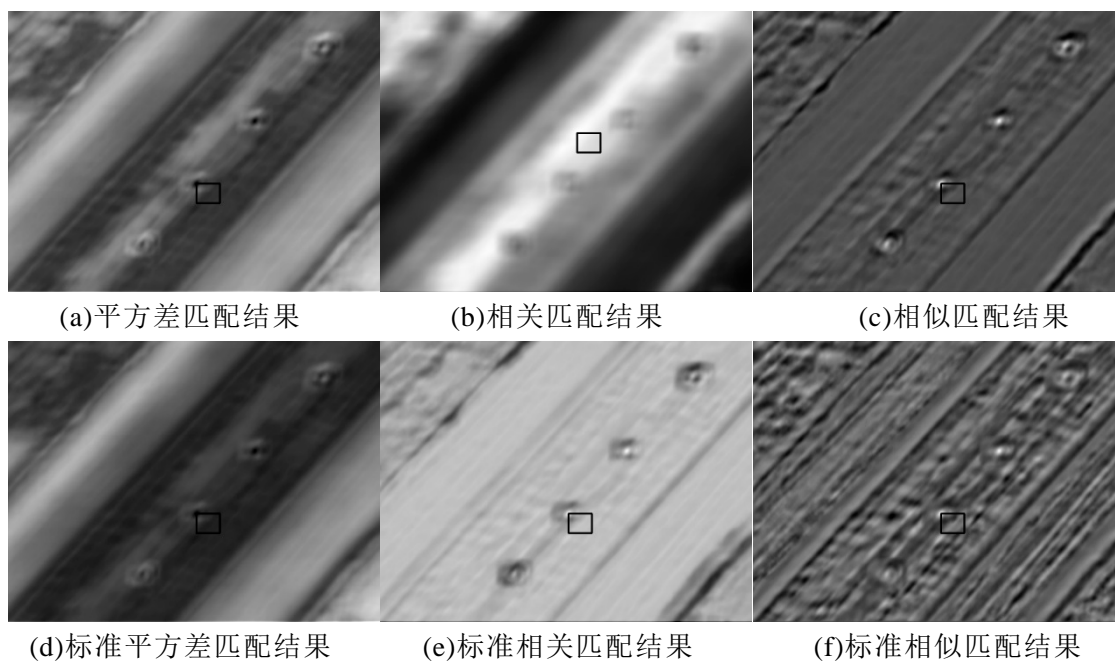


图 2-18 六种匹配准则的模板匹配结果

模板匹配中不同匹配方式的计算量不同，匹配精度也不甚相同。利用多组图片进行测试得出，在不影响实时性的前提下，标准相似匹配得到的系数更为精准。因此，利用此种方式计算实现目标捕获功能，效果如图 2-19 所示。

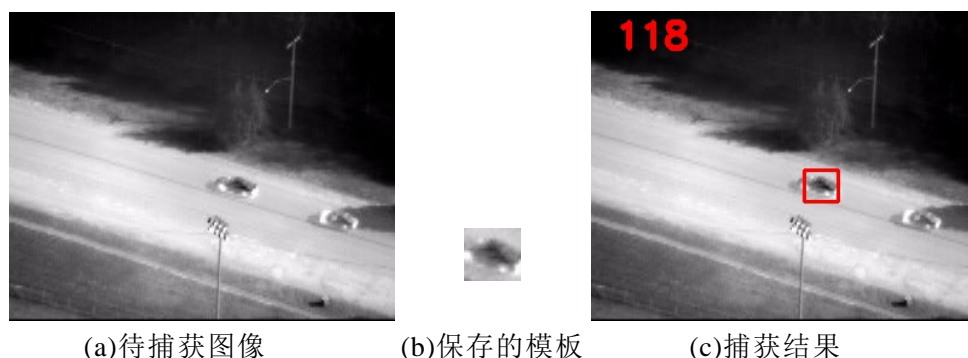


图 2-19 目标重新捕获效果图

通过对两种目标捕获技术进行分析可得，利用边缘检测与模板匹配都能成功捕获到目标，但从算法对遮挡问题的处理速度考虑，利用边缘检测进行目标捕获需要对目标进行一系列处理操作，过程非常复杂。且由于地面目标与周围环境的灰度值差异较小，造成了边缘检测效果并不理想，课题最终选用模板匹配的方式对离开遮挡区域的目标进行重新捕获。

## 2.4 本章小结

本章针对机载跟踪系统的研制进行了总体方案设计，并为解决机载下视场景的遮挡问题设计了抗遮挡处理流程并对抗遮挡跟踪技术进行了深入研究。针对实际应用对机载跟踪系统进行了需求分析，给出了系统性能指标并设计了机载跟踪系统的总体研究框架及应用框架。深入研究了机载下视抗遮挡目标跟踪技术，在对多类跟踪方法特性进行综合分析的基础上，选择基于时空上下文的目标跟踪技术作为系统核心技术。针对机载下视的遮挡问题设计了解决方案，并从遮挡判断、轨迹预测与目标重捕三方面对遮挡问题的处理进行了技术分析，得出抗遮挡跟踪的解决方案。

## 第3章 嵌入式机载跟踪系统实现

### 3.1 系统环境搭建

本课题研制的目标跟踪器在硬件方面要求能搭载在无人机等飞行平台运行，保证实际应用，因此硬件设备的选取需要采用小型化的高频设备<sup>[59]</sup>，在满足机载平台的搭载条件下，尽可能地提高执行任务的处理速度，保证跟踪处理的实时性，达到良好的应用性能；技术方面则要求能对机载下视场景中的运动目标进行准确快速的定位跟踪，保证跟踪效果。

机载下视场景中的目标跟踪由于平台抖动、目标尺度变化、遮挡等问题造成跟踪性能下降，为跟踪技术在机载场景的实际应用增加了难度<sup>[60]</sup>。为克服这一困难，可通过研究目标跟踪技术进行算法优化以提高跟踪的准确度及鲁棒性，对机载下视场景中的运动目标进行准确快速的定位跟踪，改善目标跟踪技术在实际场景中的应用问题。要能适应无人机环境，嵌入式平台是必然选择。此外机载目标跟踪系统的研制还需满足如下需求：

- (1) 具备视频采集卡，支持视频的读入与传输
- (2) 系统的处理能力需满足实时处理视频的需求
- (3) 可部署计算机视觉库等第三方环境，便于目标跟踪技术的实现
- (4) 支持 GPU 加速提高跟踪速度
- (5) 具备网卡及无线模块，用于地面对跟踪平台进行远程控制

综上所述，机载跟踪系统的硬件环境需要选择具有低功耗、小体积、高性能的嵌入式平台，最好能支持 GPU 运行；由于裸机程序很难满足上述要求，还需要支持嵌入式操作系统的运行。综合以上需求从处理速度、功耗等角度考虑，最终选取嵌入式 GPU 设备作为机载目标跟踪系统的硬件开发平台。

机载跟踪系统环境的搭建分为硬件环境和软件环境，如图 3-1 所示。硬件环境主要由视频采集设备和嵌入式 GPU 平台构成。软件环境从底层到上层依次由系统环境、开发环境及人机交互构成。利用机载跟踪系统对地面目标进行定位跟踪时，首先通过视频采集设备采集下视场景的视频数据，并将其送入嵌入式 GPU 平台进行处理。综合上述开发环境对嵌入式系统的调研可知，平台需要具备包含操作系统的系统环境，以完成复杂的跟踪及控制功能，同时，可以较为方便地搭建程序的开发环境，并设计人机交互界面进行远程控制与输出。

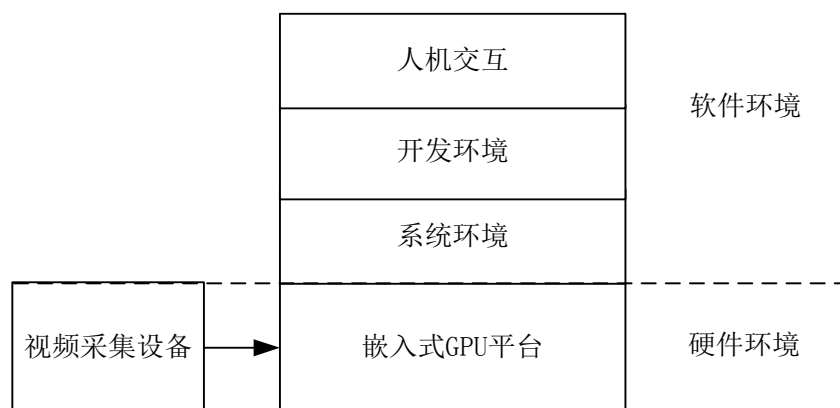


图 3-1 机载跟踪系统环境构成图

### 3.1.1 硬件平台搭建

本课题研制的机载跟踪系统的硬件环境构成如图 3-2 所示，主要包括光电吊舱、TX1 嵌入式平台、系统控制器及显示设备。

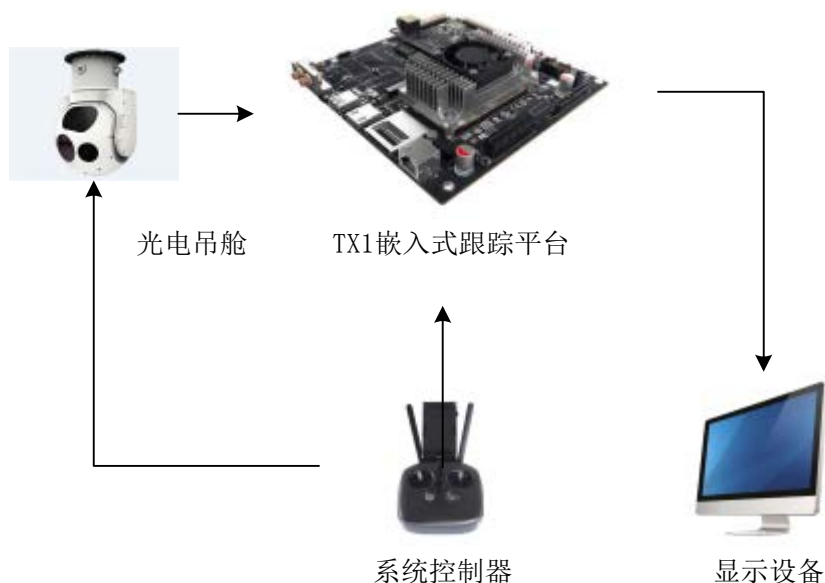


图 3-2 机载跟踪系统硬件环境

光电吊舱也称为机载光电吊舱，通常装有前视红外摄像机、电视摄像机和激光指示器等光电传感器<sup>[60]</sup>，按功用目标主要有机载导航吊舱和瞄准吊舱两类，可搭载在无人机等飞行设备上完成对视频数据的实时采集、存储与传输等工作；在机载跟踪系统的硬件平台选择方面，根据课题对机载目标跟踪器的高性能、小型化、轻量级的需求，通过对各类嵌入式设备进行应用分析，选用 Jetson TX1 平台作为本课题研制的机载目标跟踪系统的开发平台，强大的浮点

计算能力和软件堆栈以及低能耗和小体积的特点<sup>[61]</sup>,使得 TX1 可以负载特定场合下中小规模的机器学习的应用于开发。系统控制器作为机载跟踪系统的整体控制模块,可对光电吊舱、嵌入式开发平台用于操作人员对跟踪平台进行控制及对光电吊舱的角度控制,显示设备用于将跟踪结果显示输出。Jetson TX1 面向移动嵌入式系统市场的智能计算处理,采用了计算机视觉、GPU 计算和图像方面的新技术,是适用于智能处理的全功能开发平台<sup>[62]</sup>,其万亿级的浮点计算能力和强大的软件堆栈以及超低的能耗,足以负载特定场合下的中小规模机器学习的应用和开发,适合需要低能耗和高计算性能的嵌入式应用系统开发,是一款面向无人智能化领域的理想嵌入式开发平台。

表 3-1 Jetson TX1 平台配置表

TX1 主板选项	详细信息
JETSON TX1 模块	NVIDIA Maxwell GPU 及 256 个 NVIDIA CUDA 核心 四核 ARM® Cortex®-A57 MPCore 处理器 4 GB LPDDR4 内存 16 GB eMMC 5.1 闪存 802.11ac Wi-Fi/蓝牙 10/100/1000BASE-T 以太网
按钮	开机/关机、重置、强制恢复、用户自定义
电源选项	以太网 19V 交流适配器
I/O 接口	USB 3.0 类型 A USB 2.0 Micro AB (支持恢复和主机模式) HDMI M.2 Key E PCI-E x4 千兆位以太网 完整尺寸 SD SATA 数据和电源 GPIO、I2C、I2S、SPI 带流控制的 TTL UART 显示器扩展头 摄像头扩展头

Jetson TX1 平台的配置如表 3-1 所示,该平台运行 Linux for Tegra 操作系统,具有 Cortex-A57 架构的 64 位 4 核 ARM CPU,Maxwell 架构的 256 CUDA 核 GPU,4GB LPDDR4 内存,16GB 闪存,具有蓝牙、Wifi 等模块,此外,还

配备了 USB、HDMI、PCI-E、SD、SATA 数据和供电等接口及 GPIO，I2C、SPI、带流控制的 TTL UART 扩展接口及显示器与摄像头的扩展头。平台功耗仅为 10W，真正做到了低功耗标准，拥有强大的计算能力，每秒可执行 1T 的浮点运算。由表格可以看出，TX1 的数据存储只有 16GB，这限制了其开发与应用能力，因此课题中为 TX1 硬件平台配备了固态硬盘，增大存储量，为在 TX1 上对各类开发工具及应用软件的环境部署提供了条件，此外由于平台的端口有限，还需配备 Hub 来扩展接口以接入控制设备、视频采集设备等，显示设备通常通过 HDMI 接口连接，构成机载跟踪系统的硬件开发平台。

### 3.1.2 软件环境部署

机载跟踪系统软件环境的搭建主要包括嵌入式系统环境、人机交互环境及开发环境的部署三方面，如表 3-2 所示。系统环境的搭建主要通过 PC 机与嵌入式平台联合安装 Jetpack 工具包完成，配备 Ubuntu16.04 的操作系统、Nsight 开发工具及 GPU 加速库 CUDA。目标跟踪的应用开发利用 Nsight 工具以同步编译的方式在宿主机机利用 TX1 中部署的 OpenCV 及 CUDA 等环境进行应用程序的远程编译与开发，人机交互是在板上利用 Qt 开发的，通过触发信号与槽函数建立程序输出与控制的关系。

表 3-2 嵌入式目标跟踪系统软件环境构成表

环境类型	环境名称	功能
系统环境	Jetpack	包含系统映像、开发环境及示例、文档等在 内的 TX1 一体化功能包
第三方库环境	OpenCV	提供计算机视觉库
人机交互	Qt	人际交互实现

嵌入式系统环境的搭建通过安装 Jetpack 包完成。NVIDIA 为 TX1 提供了工作环境一体化安装包，可统一部署系统环境<sup>[61]</sup>。Jetpack 是 x86 二进制文件，不能在 ARM 环境运行，需要利用宿主机烧录到嵌入式平台<sup>[63]</sup>。安装 Jetpack 时，通过路由器将 TX1 与宿主机连接并接入网络，进行嵌入式平台的环境部署，如图 3-4 所示。课题使用的 Jetpack 安装包为 2.3.1 版本，宿主机需要有 Ubuntu14.04 的 Linux 操作系统。

Jetpack 包含的工具如表 3-3 所示，该工具包为研制嵌入式 GPU 平台提供了系统环境、库函数及编译器等开发环境的支持。Jetpack 部署完成后，可以在 TX1 上独立开发应用程序，也可以通过 SSH 建立宿主机与 TX1 的远程联系，



对嵌入式平台进行远程控制与开发。

表 3-3 Jetpack 2.3.1 主要工具表

类型	工具名称	功能描述
系统映像	L4T 24.2.1	64 位 Ubuntu16.04 操作系统
开发工具	CUDA 工具包	构建 GPU 加速的开发环境,包含 GPU 编译器、GPU 加速库、相关数学库及调试与优化工具
	多媒体 API	主要包括相机应用程序 API Libargus 和视频编解码、格式转换驱动程序 API V4L2
	VisionWorks	视觉和图像处理开发包,包括视觉编程接口 VPI
	OpenCV4Tegra	CPU 和 GPU 加速版的 OpenCV
	Tegra 图形编译器	控制台工具,允许开发人员调试配置 OpenGL
	系统分析器	多核 CPU 采样分析器,提供数据的交互视图
	PerfKits	提供对 OpenGL 驱动和 GPU 计数器访问的软件库
示例	多媒体 API 示例、CUDA 示例、VisionWorks 示例、OpenGL 示例	
文档	Jetson 文档、Jetpack 文档	

应用程序的开发是利用 CUDA 架构及其开发工具 Nsight Eclipse Edition 进行的。通过 Jetpack 完成 CUDA 在宿主机及嵌入式平台的环境部署, CUDA 编程可以通过调用 GPU 上的计算资源加速 C 或 C++ 应用程序的开发, CUDA 为开发人员提供了可以用来调用 GPU 资源的加速库函数, 可以通过 C/C++ 语言直接调用库函数或开发自定义应用程序, 实现对应功能的加速。Nsight 是基于 Eclipse 的 CUDA 集成开发工具, 用于 CUDA-C 应用程序的开发, 提高程序的处理速度<sup>[64]</sup>。作为针对异构计算的开发平台, 利用 Nsight 进行 CUDA 应用程序的开发可充分发挥 CPU 和 GPU 的性能, 其强大的调试和分析工具, 不仅可以优化应用程序的性能, 还可以帮助开发者进行代码分析, 了解对 GPU 资源的使用情况, 是编辑、构建、调试和配置 CUDA-C 应用程序的一体化集成开发环境。由于 Nsight Eclipse Edition 是安装在宿主机上的, 需要远程调用 TX1 上的 GPU 资源及相关库环境, 通过交叉编译的方式在宿主机实现对嵌入式环境下应用程序的远程编译, 编译成功后会在嵌入式平台生成二进制格式的可执行文件, 直接运行该文件即可实现特定的函数功能。

跟踪技术的实现还需要配置 OpenCV 和 CUDA 两个库环境, 通过调用 OpenCV 的库函数辅助完成跟踪方案的技术实现, 通过调用 CUDA 中的库函数调用 GPU 资源, 进行跟踪程序的加速处理。

调用 OpenCV 库函数可以快速实现视频处理的相关功能, 加快应用程序的

开发进度；作为计算机视觉的专业函数库，OpenCV 内置多种图像处理和计算机视觉领域的通用算法并对一些算法进行了优化，同时 OpenCV 支持多类视频及图像格式的读取并将其转化为统一的数据格式，可方便的进行后续图像处理的相关操作<sup>[66]</sup>。在跟踪方案的实现过程中，视频的读取及某些功能的实现都是通过调用 OpenCV 的库函数来实现的。OpenCV 的环境部署首先需要安装环境依赖库，之后进行 OpenCV 的编译与安装，安装完成后还将其需要加入动态链接库，并配置到系统环境变量中。通过在编译器上加入 OpenCV 路径与需要用到库函数实现对 OpenCV 的调用。

调用 CUDA 库函数可实现对嵌入式平台上 GPU 资源的调用，用于应用程序的加速处理。CUDA 库是一系列用于 GPU 加速处理的接口集合，通过库中包的函数接口即可实现特定的加速功能，而无需直接对 GPU 资源进行编程，极大地减小了 GPU 的开发难度及开发时间，并可以达到很好的加速性能<sup>[67]</sup>。CUDA 库包含了多种加速函数，对快速傅里叶变换、线性函数等常见的视觉处理功能都可以进行加速处理。Nsight Elclipse Edition 作为 CUDA 的开发工具，可直接调用 CUDA 库进行加速程序的开发。

人机交互的设计与实现是通过 Qt 完成的，作为一款图形应用软件开发工具包(Software Development Kit, SDK)，Qt 可以使开发者高效地完成界面设计与程序开发工作，并进行无缝式的跨平台程序移植<sup>[68]</sup>。Qt 既支持 GUI 程序的开发，方便人机交互的设计，又支持非 GUI 程序的开发，如控制台应用程序等，并且支持众多 C++ 类及扩展组件，内置多种第三方库，加之源代码的开放使得 Qt 受到广大开发者的欢迎<sup>[69]</sup>，是嵌入式系统程序开发的理想解决方案。完整的 QT 开发环境由 Qt 框架、Qt Creator 集成开发环境(Integrated Development Environment, IDE)、示例以及文档等构成。在本课题的人机交互实现过程中，通过 Qt Creator 进行交互界面的设计并完成跟踪程序的移植。

## 3.2 抗遮挡跟踪技术实现

### 3.2.1 高速跟踪技术实现

对抗遮挡跟踪技术的研究是在对高速目标跟踪技术研究的基础上进行的。课题利用机载下视图像序列对多类跟踪方法的实际跟踪性能进行实验验证，为保证跟踪算法在加入抗遮挡处理后仍能对地面目标进行实时跟踪，选择跟踪速度优异的时空上下文跟踪方法进行抗遮挡处理研究，通过第二章对跟踪思想及跟踪过程的技术研究，设计的基于时空上下文的跟踪方法流程图如图 3-3 所示，

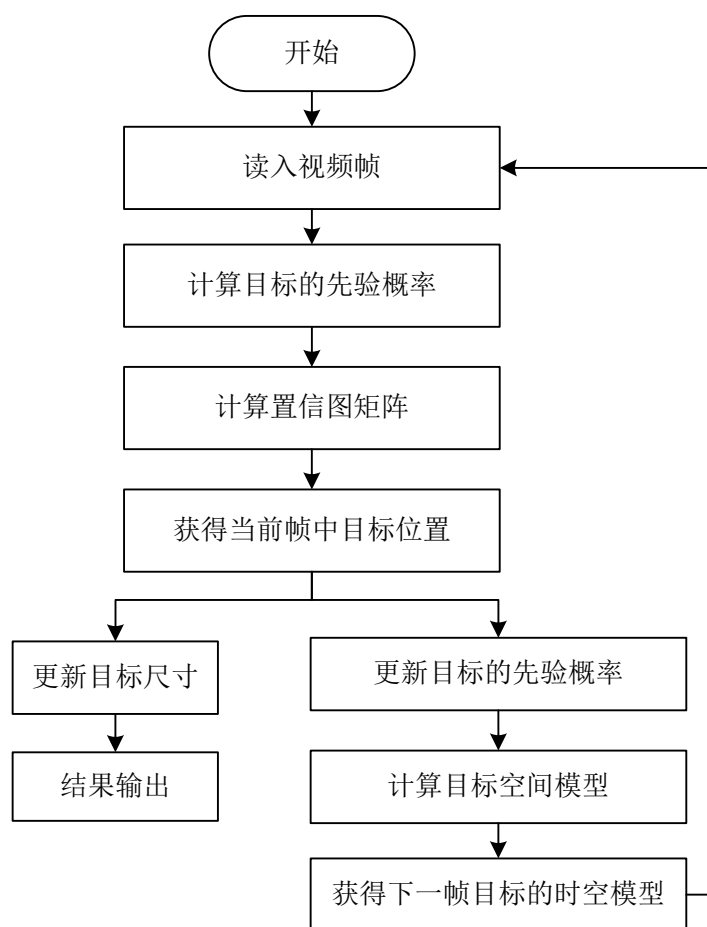


图 3-3 基于时空上下文的目标跟踪流程图

利用时空上下文进行跟踪的方法利用目标在前一帧中的位置及与当前帧的关系来确定目标在当前帧中的位置。读入视频数据后，根据当前帧中已知的目标信息计算目标的先验模型，其中，第一帧的目标由人工确定。通过先验模型与时空模型求得目标的置信图，并由此得到目标在当前帧中的位置，利用比例更新修正目标尺寸后，即可得到当前帧中的目标位置<sup>[69]</sup>。根据新的目标位置进行先验概率模型的更新，并根据置信图模型与先验概率模型得到表征目标与周围空间关系的空间模型，对空间模型进行加权求和，得到计算下一帧中目标位置的时空模型，在读入下一帧图像后，利用先验概率模型与时空模型继续进行目标的定位。跟踪过程中，利用目标在前一帧中的位置信息获得先验概率模型，通过与当前帧的时空模型进行计算，求得目标在当前帧的位置，由此实现跟踪的动态过程。通过更新先验模型对目标与周围空间的关系进行修正<sup>[70]</sup>，从而建立准确的时空模型，用于下一帧的目标定位。该方法利用傅里叶变换进行卷积与反卷积的相关运算，极大地提高了跟踪速度。

跟踪方案实现的关键是对各个概率函数的建模及模型在时域与频域间的转换。先验概率模型根据“Focus of Attention”原理，利用高斯加权函数对目标上下文区域的灰度特征进行加权处理，以实现目标在当前图像中的特征信息进行描述。空间概率模型通过置信图模型与先验概率模型经过傅里叶变换后在频域相除得到频域表述，再经过傅里叶逆变换后得到的概率矩阵。时空概率模型是对上一时空模型与当前帧的空间模型进行加权求和实现的。利用时空模型与目标在本帧图像中的先验概率模型进行卷积操作，得到表示目标在图像中各个位置出现的概率置信图，出现概率最大的位置即为目标位置，由置信图的最大值来确定。比例更新根据表示目标位置中心的置信图极值来预测连续帧间的目标比例，从而完成目标尺寸的更新。为减少预测误差造成的噪声影响，通过滤波处理得到预测的目标比例<sup>[73]</sup>。同时，为避免目标尺寸的频繁变化对跟踪造成的影响，在跟踪方案的实现中，每隔 5 帧进行一次比例更新，以减弱跟踪过程对目标尺寸的敏感性<sup>[42]</sup>。

跟踪方案的实现过程涉及多种变量类型，较为关键的有承载视频的 `Videocapture` 类，承载图像和概率矩阵的 `Mat` 类及用于跟踪结果显示的 `Rect` 类。对视频或摄像头的读取是通过建立 `Videocapture` 类实现的。`Videocapture` 类是 `OpenCV` 提供的用于处理视频文件的类结构<sup>[71]</sup>，通常有两种用法，一种是通过视频文件名打开视频数据；另一种是通过建立视频通道打开视频采集设备，进行视频的实时读取。摄像头等视频采集设备的视频通道默认为 0。对图像数据的承载、概率函数间的参数传递及函数内部的矩阵处理等操作都是通过建立 `Mat` 类结构实现的。`Mat` 类结构的使用非常灵活，在创建过程中可根据承载的对象自动开辟存储空间，在使用结束后也无需进行手动释放<sup>[72]</sup>。另外，`Mat` 类结构可以通过计数机制使几个 `Mat` 类共享一个矩阵，便于函数间图像矩阵的传递。跟踪结果的显示是通过 `Rect` 类实现的。跟踪结果是通过以目标位置为中心，目标尺寸为大小的 `Rect` 类矩形框在图像中进行标注显示的。`Rect` 类包含四个成员变量，分别表示矩形左上角的坐标和宽高，与表示跟踪结果的四个参数一一对应。此外，跟踪技术的实现过程中还利用 `Rect` 矩形类来进行目标模板的提取工作。

### 3.2.2 抗遮挡跟踪技术实现

通过对抗遮挡跟踪的方案设计与技术研究，对目标经过遮挡区域时的跟踪处理分为遮挡时刻的判断、遮挡区域目标运动轨迹的预测及离开遮挡区域后的目标的重新捕获三个环节进行。下面分别对这三个环节的实现过程进行说明。

利用感知哈希指纹值及目标相关系数判断遮挡发生的实现流程如图 3-4 所示。

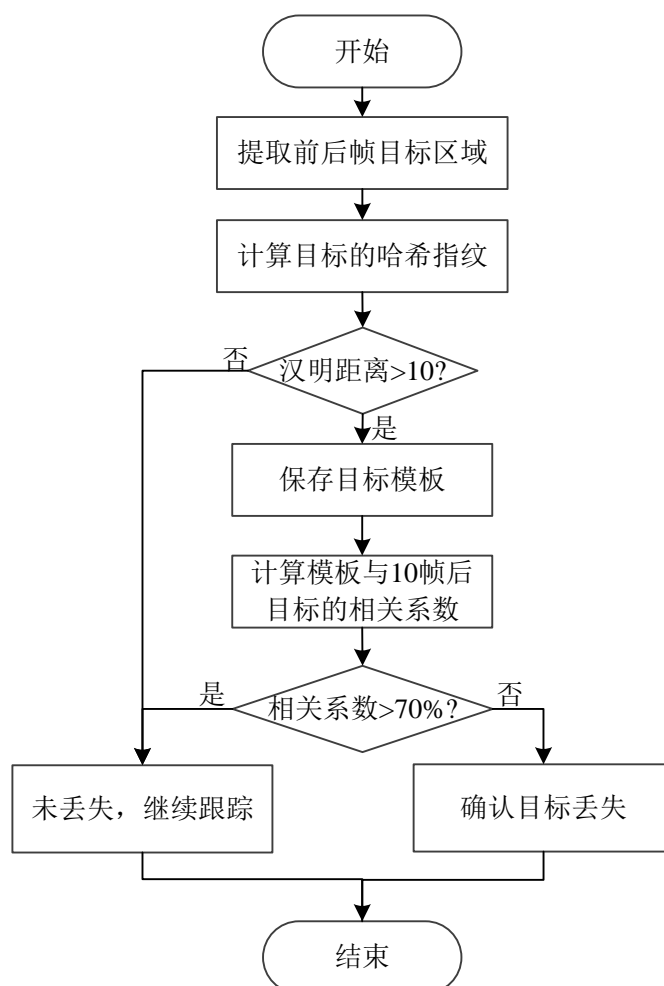


图 3-4 判断遮挡发生流程图

由于抗遮挡方案中需要对目标状态进行两次判断,因此采用两种相关性判断方式来检测目标状态。遮挡预警阶段,利用感知哈希算法生成哈希指纹来表征目标特征,该方法利用图像压缩提取目标的低频信息,生成表征目标身份的指纹字符串<sup>[74]</sup>,利用哈希指纹的汉明距离判断目标间的相关性,从而判断目标是否可能遇到遮挡,汉明距离的阈值取 10,若汉明距离小于 10,则证明前后帧目标的相关性比较大,表明目标未被遮挡,继续跟踪过程;若汉明距离大于 10,则保持目标此时的模板,并在继续运行 10 帧后计算模板与当前帧目标的相关系数,系数阈值取为 70%,若相关系数高于 70%,则说明目标未被遮挡,可能是预警时由于阴影、光照变化等情况对目标产生的影响而发生了误判,

继续跟踪过程；若低于 70%，则确认跟踪的目标遇到了遮挡情况，进行后续遮挡区域的处理流程。

利用感知哈希指纹判断目标相关性的实现过程包括哈希指纹的生成与相关性的判断两部分，具体实现步骤如下：

(1) 缩小尺寸。通过减小图像尺寸来去除细节信息，避免图像比例带来的差异，保留结构、灰度等基本信息用于哈希指纹的生成。

(2) 简化色彩。图像通常具有 256 级灰度，通过将其转化为 64 位灰度级来减少计算量。

(3) 图像压缩。通过离散余弦变换进行图像压缩得到 DCT 系数矩阵。

(4) 提取低频信息。有图 2-7 可知，图像信息主要集中于低频区，因此只保留表示图像最低频率的左上方区域系数矩阵，通常为  $8 \times 8$  区域。

(5) 计算系数均值。计算缩小的系数矩阵的均值，作为哈希值的评价标准。

(6) 生成哈希指纹。根据利用系数均值对低频系数矩阵做二分类处理，大于等于均值的记为 1，小于均值的记为 0。由此，将表征目标特征的  $8 \times 8$  区域用一条包含 0、1 信息的 64 位指纹字符串表示。

(7) 计算汉明距离。根据哈希指纹不相同的位数生成汉明距离，判断目标间的相关性。

(8) 目标相关性判断。通过多次实验并结合经验数据，汉明距离取 10 作为目标相关性的判断标准。若前后帧目标生成的哈希指纹的汉明距离小于 10，则说明目标相关性很高，由此判断目标未遇到遮挡；若汉明距离大于等于 10，则说明目标的相关性较差，由此判断目标在前进方向可能遇到遮挡。

由于上下文区域是目标区域的两倍，可对遮挡做出提前反应。因此，首次判断遮挡时，利用了目标的上下文区域做相关性计算，这样可提前判断遮挡的发生以便及时保留目标模板，而在二次判断时，为提高准确性，选用目标区域进行相关性的判断。

对目标在遮挡区域的运动轨迹预测是利用遮挡发生前 5 帧的目标中心建立运动方程来实现的，其实现流程如图 3-5 所示。由于遮挡时刻的目标运动状态难以判断，造成了对遮挡区域的运动轨迹预测困难。若目标在遮挡时出现拐弯、停止或超车等复杂情况，从目前的处理技术来看，还没有有效的处理方法。因此，课题中根据目标的常规运行轨迹，将目标运动假设为直线运动，利用遮挡发生前 5 帧的目标中心构造直线运动函数，运动速度及方向通过各中心点构成的运动矢量得到。

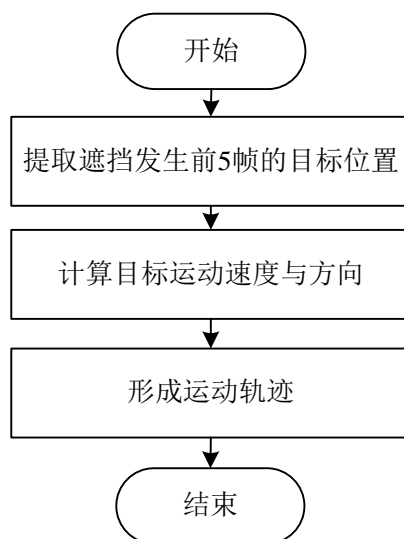


图 3-5 遮挡区域目标运动轨迹预测流程图

在目标离开遮挡区域后,对目标的重新捕获是利用模板匹配在当前图像区域内搜索与保存模板相似度最高的图像区域实现的,其实现流程如图 3-6 所示。

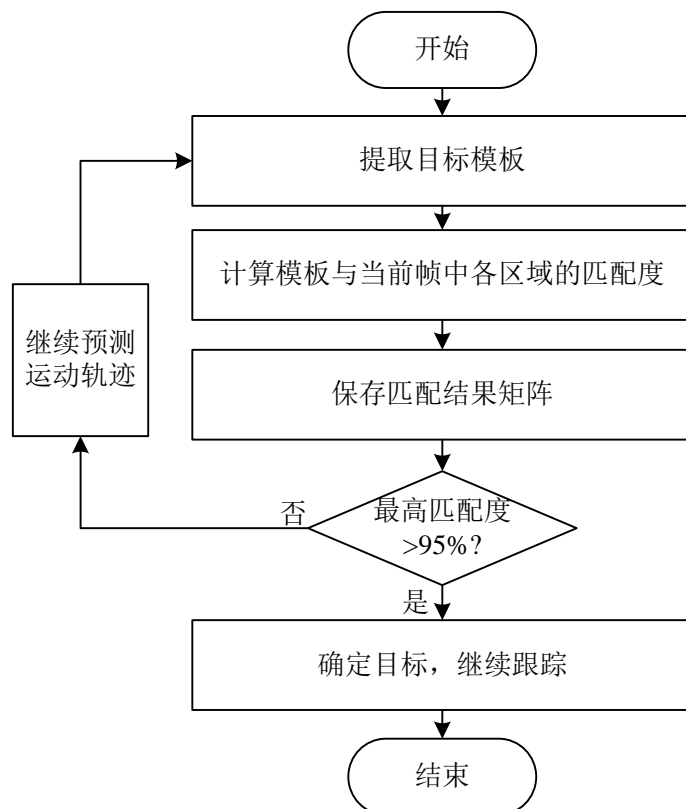


图 3-6 目标重新捕获流程图

提取目标模板,并利用模板在当前图像帧滑动计算各个位置与目标区域的

匹配度，将结果以矩阵形式保存并进行归一化处理，将匹配度最高的点与经验值作比较<sup>[75]</sup>。若匹配度最大值达到标准，则将其作为目标顶点，目标尺寸由目标模板的尺寸确定，确定目标后继续进行跟踪任务。若匹配度不足则表示未能重新捕获，目标仍在遮挡区域内，则进行若干帧的轨迹预测后继续进行目标捕获，直到目标与模板的匹配度达到标准，完成对目标的重新捕获。为提高搜索效率，以轨迹预测的目标位置为中心，在范围为目标模板三倍的区域内进行匹配搜索，缩减搜索区域的同时，也将部分背景因素排除在外，提高了捕获的准确性。最高匹配度的位置代表目标顶点，根据模板尺寸确定目标大小，由此完成捕获过程。

通过上述对遮挡处理机制的各个环节在实际跟踪过程中的效果进行有效性分析可以得出，感知哈希算法与相关性系数结合判断目标的相关性，可以有效判断目标在前进方向遇到遮挡的时刻；利用遮挡前的目标信息可以有效地对目标运动轨迹进行预测；在目标重新出现时，根据模板匹配，利用遮挡发生前保存的目标模板在当前帧中捕获到目标，以进行后续的跟踪定位。由此可以验证本文提出的抗遮挡跟踪方案可以对遮挡情况下的目标跟踪进行有效处理，这对提高机载跟踪系统的准确性起了重要作用。

### 3.3 GPU 加速及人机交互实现

#### 3.3.1 跟踪技术的 GPU 加速

为提高跟踪算法的处理速度，利用嵌入式平台的 GPU 资源对程序进行加速处理。跟踪技术的 GPU 加速是基于 CUDA 框架实现的。作为 NVIDIA 公司开发的用于 GPU 编程的开源框架，CUDA 通过调用显卡的 GPU 资源进行数据运算，提高了设备的实时处理能力，并将 CPU 从繁杂的数学计算中解放出来，更好地用于对系统的控制管理<sup>[76]</sup>。CPU 与 GPU 的能力差异表现在 GPU 适用于计算强度高的多并行浮点运算，而 CPU 更偏重与流程控制。因此 GPU 拥有更多的晶体管，可以在多并行计算时每个数据单元执行相同程序，也因此 GPU 的计算单元结构是非常简单的。

CUDA 框架提供了可应用于图像领域的多个 GPU 加速库，如用于图像快速傅里叶变换的 cuFFT、用于线性代数运算的 cuBLAS 及用于常规数学运算的 CUDA Math Library 等；CUDA 还提供了一系列用于应用程序开发的开发工具，如用于 CUDA C/C++ 程序编译的 NVCC 编译器，Nsight 集成开发环境，CUDA-GDB 命令行调试器、CUDA-MEMCHECK 内存器及 Visual profiler 视觉



剖析器等；此外，CUDA 环境部署完成后还会带有相应的文档与示例，便于开发人员更好地利用 CUDA 进行应用程序的加速开发。

在对跟踪程序的加速处理中，对 CUDA 程序的开发是利用 CUDA 提供的开发工具 Nsight Eclipse Edition 进行的，为节约嵌入式的资源，Jetpack 将 Nsight 安装在了宿主机上，需要通过 SSH 建立宿主机与嵌入式平台的联系，通过交叉编译在宿主机远程调用嵌入式平台的 GPU 资源及 OpenCV、CUDA 等库函数，进行应用程序的远程编译与链接<sup>[80]</sup>，生成二进制的可执行文件可在嵌入式平台上直接运行。

在 CUDA 架构下，程序分 Host 端和 Device 端两部分执行，如图 3-7 所示。通用部分在 Host 端调用 CPU 执行程序，加速部分在 Device 端则是在显示芯片上调用 GPU 执行程序，在 GPU 执行的程序又称为 Kernel<sup>[76]</sup>。

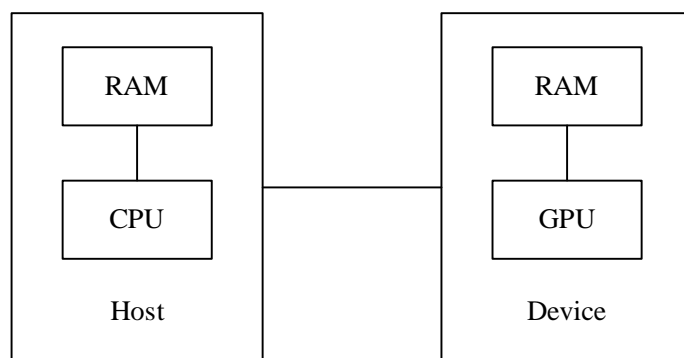


图 3-7 Host 端与 Device 端结构图

利用 CUDA 编程调用 GPU 资源进行加速的过程如图 3-8 所示。首先需要将程序中需要进行加速计算的数据从 CPU 中提取出来，根据数据类型在 GPU 中为其开辟相应的内存空间，然后创建要使用的加速函数的句柄，将要执行的加速函数及参数与句柄联系起来，句柄可创建 1 维、2 维及多维等多种形式<sup>[77]</sup>；之后将存储在 CPU 内存中的数据块复制到在 GPU 中为其开辟的显存中，根据句柄执行加速函数，调用 GPU 资源实现对数据的并行计算，将得到的计算结果从 GPU 显存中再复制并返回 CPU 内存中存储数据的相应区域，完成对数据的并行加速运算。将加速结果返回程序的数据区域，以继续执行后续的处理。

在跟踪技术的 GPU 加速实现中，用到的 CUDA 加速库主要有 cuFFT、cuBLAS 和 CUDA MATH Library。通过对跟踪方法的实现过程进行解析，本课题应用的目标跟踪方案中可用于加速的部分主要有用于用模型的学习过程的快速傅里叶变换、加权求和及常规数学运算，对此针对性地选用了上述三个加速库，通过调用加速函数对数据进行高速运算，提高算法的处理能力。cuFFT

的计算速度比常规傅里叶变换的速度高出 10 倍之多,在 GPU 中调用了数百个处理器内核进行并行计算,充分利用了 GPU 的浮点计算能力。cuBLAS 库是标准的线性代数程序的 GPU 加速库,通过调用库内的 API,可以将计算密集型操作部署到单个 GPU 或通过配置多 GPU 高效地分布对数据的处理工作来加快应用程序的速度,可以利用 cuBLAS 在高性能计算、图像分析和机器视觉领域开发 GPU 加速的各种算法<sup>[78]</sup>。CUDA 数学库是经过业界验证的高精度数学函数集合,在开发 CUDA 应用程序时,只需要在源代码加入常规数学库,即可对相关数学函数进行 GPU 优化,称为高性能数学应用程序,NVIDIA 提供的所有 CUDA 库,都是依赖 CUDA MATH Library 提高了突破性的成果<sup>[79]</sup>。

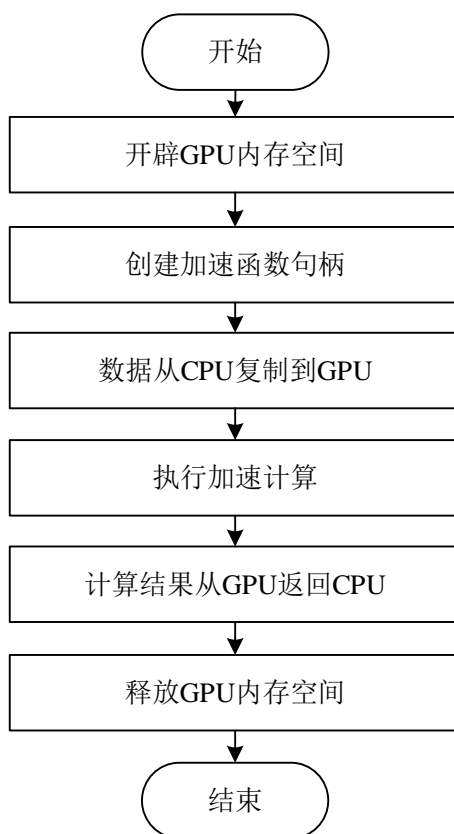


图 3-8 调用 CUDA 库函数的 GPU 加速过程

### 3.3.2 人机交互界面的设计与实现

人机交互过程是基于 Qt 进行开发实现的。作为跨平台的应用程序和 UI 开发框架,Qt 同时支持界面程序及应用程序的开发,支持多种第三方库,可通过创建界面项目集成界面与应用程序,符合课题的应用需求。进行界面设计时,可利用界面编辑器提供的各类控件及对控件属性的设置进行用户界面的快

速部署。利用信号与槽函数机制<sup>[81]</sup>，将要实现的功能在控件触发信号的槽函数内实现，即可建立将界面控件与应用程序的通信，通过对界面控件执行相应的触发信号完成视频的输入、应用程序的过程控制及结果输出等功能。

人机交互界面的设计及对跟踪过程的控制是通过建立 Widgets 项目实现的，界面的设计工作是通过 GUI 布局和格式构建器 Qt Designer 完成的，由于使用了与应用程序中将要使用的相同部件，可以利用格式构建器进行格式的快速设计、创建部件以及对话框等<sup>[82]</sup>。利用界面控件的动作作为触发信号，对应的槽函数则用来接收信号，槽函数可以看作是普通函数，在触发信号发出后被调用用来做出功能响应。

人机交互界面的设计如图 3-9 所示，包括功能选择区、视频显示区及数据显示区三部分。

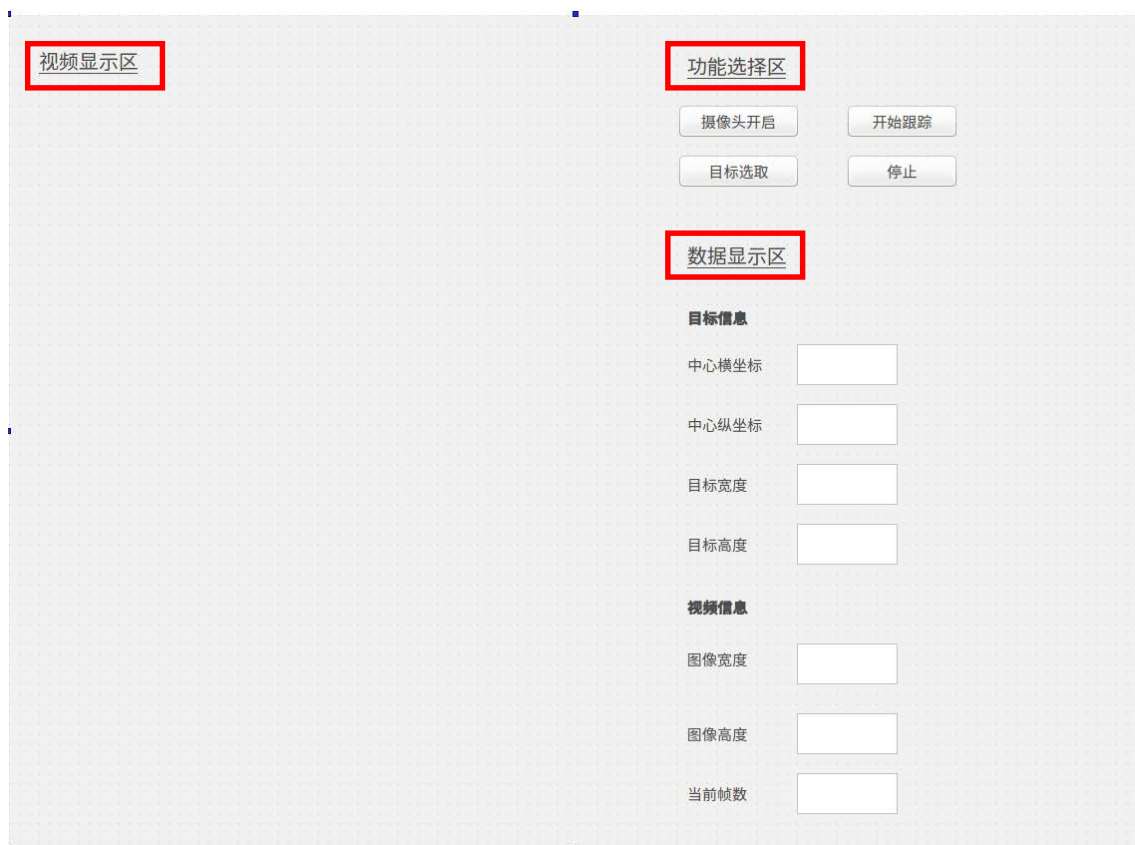


图 3-9 人机交互界面

功能选择区包含了目标跟踪任务中可执行的各个操作，包括打开视频采集设备、选取初始目标位置信息、执行跟踪任务及停止跟踪等功能，通过功能选择区可以实现对整个跟踪任务的控制与选择。视频显示区显示跟踪处理的视频

结果，将目标位置利用矩形框标注并加载在视频中进行结果输出，直观地显示了跟踪结果；数据显示区包含了跟踪任务中的重要输出参数，由视频的基本信息及目标信息两部分构成。视频信息包含了视频帧的尺寸及当前正在跟踪的帧数等信息。目标信息包含了目标中心的横纵坐标、目标宽度、目标高度等信息，是可用于进行继续分析的内容。图像尺寸和目标尺寸是影响跟踪任务的重要因素，图像尺寸的大小对跟踪速度有着直接影响；而目标尺寸的大小则关系着跟踪的准确性与成功与否。

在嵌入式目标跟踪系统的人机交互实现中，需要解决的核心问题是建立界面与应用程序间的通信，通过界面对跟踪程序进行控制并进行跟踪结果的输出显示，需要解决应用程序与界面间的格式冲突问题与程序运行机制的改变等方面的问题。

目标跟踪方案的实现中，对图像数据的承载是通过 **Mat** 类结构实现的，而在 **Qt** 中是以 **Qimage** 类承载图像数据的，二者之间存在图像通道、承载方式等方面的冲突，要使用 **OpenCV** 库函数进行视觉技术的开发，必须使用 **Mat** 类来进行承载并处理图像数据，此类结构可以自动进行内存分配与管理，省去了传统的 **cvMat** 和 **IplImage** 类在使用完后需要对内存进行释放的问题，并且在赋值运算和拷贝时，只复制其信息头，数据指针指向了被拷贝的数据地址，并非利用内存新建一个矩阵，省去了大量的内存占用，对提高处理效率方面有很大作用；而要利用人机交互实现对跟踪程序的控制，必须使用 **Qimage** 类来承载图像，利用该种结构承载的图像可以进行像素级访问并进行修改设计，在进行输出显示时，创建 **Pixmap** 类将结果通过输出到显示控件，可以这样才可以建立图像的显示对象并输出到界面显示跟踪的视觉效果。

为解决人机交互中二者的冲突，需要解决图像的格式兼容问题，为此需要构造格式转换函数，在两类结构间灵活转化，以建立应用程序与交互界面的通信。跟踪程序在人机交互实现中的格式转换机制如图 3-10 所示。

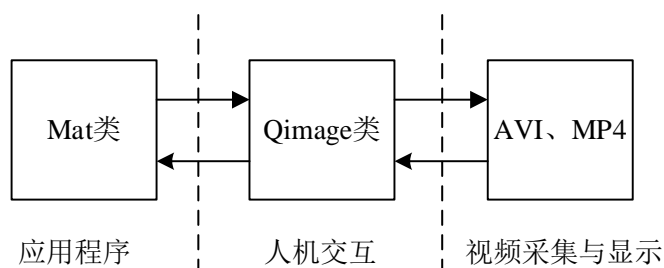


图 3-10 人机交互过程的格式转换机制

通过视频采集设备采集的数据为 **AVI**、**MP4** 等常见视频格式，在通过人

机交互界面进行视频的采集与跟踪结果的输出时，以 Qimage 承载图像；在通过人机交互界面对内部跟踪程序的模式及相关参数进行控制时，图像以 Mat 格式调用 OpenCV 实现特定的跟踪功能。

目标跟踪需要完成对连续图像的目标定位工作，是一个动态处理过程。在对读入的视频文件进行分帧处理后，利用循环结构实现对视频图像序列的依次读入，并逐帧进行目标定位，形成跟踪的动态过程。但槽函数的触发机制与循环进程的处理机制会产生冲突，触发槽函数的进程无法与循环结构进行同时执行。因此，在槽函数中不支持循环结构的使用。为实现通过 Qt 循环处理视频帧，构造了附加槽函数，通过定时器在附加槽函数与触发槽函数建立联系。在控件的触发函数中执行开始视频采集设备与定时功能，将定时结束作为触发信号，在定时器触发的槽函数中执行读入下一帧图像与当前帧图像的目标定位工作，实现跟踪的动态过程。

### 3.4 本章小结

本章研究机载跟踪系统的实现问题，在第 2 章对机载跟踪系统进行方案设计与抗遮挡跟踪技术研究的基础上，针对机载下视目标跟踪器的技术要求，利用嵌入式 GPU 平台 Jetson TX1 进行了机载跟踪器的研制。通过 Jetpack 部署了平台开发环境，基于 CUDA 框架，利用 Eclipse 结合 OpenCV 库以交叉编译的方式实现了抗遮挡跟踪算法的应用开发。针对跟踪器的实时性要求，通过 CUDA 库调用平台的 GPU 资源进行了算法加速，并基于 Qt 设计了人机交互界面，实现对机载跟踪器的控制与结果显示。

## 第 4 章 机载跟踪系统综合测试与分析

### 4.1 测试环境与测试方法

课题的理想测试环境是将跟踪模块搭载在无人机等飞行设备上系统进行功能与性能方面的综合测试。鉴于实验室目前不具备云台摄像头、无人机平台等测试平台，且还要考虑跟踪模块的搭载方式及摄像头的运动等问题，在实际场景的应用测试难度较大。因此，对跟踪设备的研制及测试转在室内进行，利用机载下视场景的测试数据集来弥补测试场景的差异。机载跟踪设备的测试是在 Jetson TX1 开发平台进行的，其测试环境如图 4-1 所示：

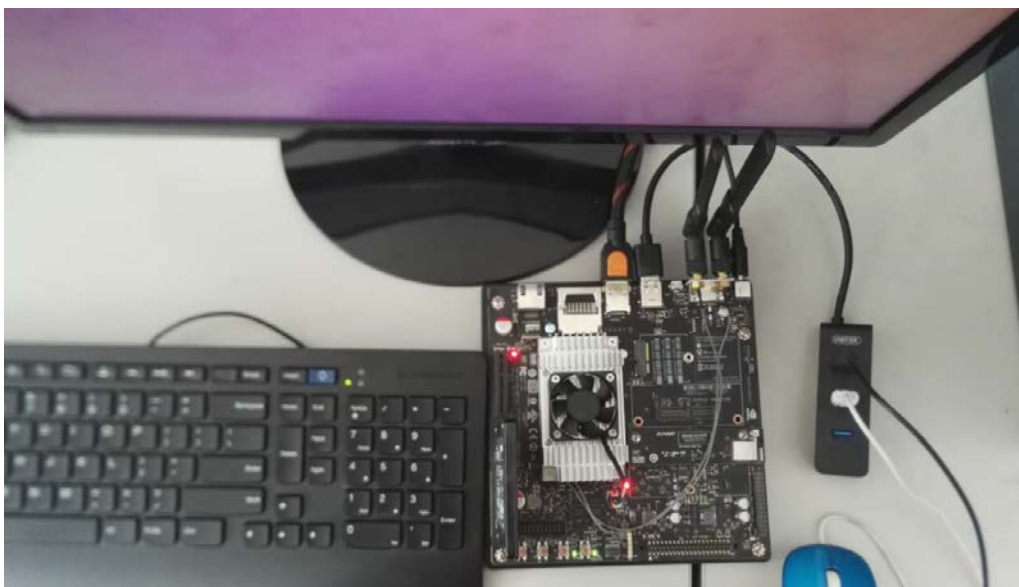


图 4-1 测试环境构成图

由 TX1 开发平台连接输入设备及显示设备等构成。通过 DC 接口接入电源，扩展 USB 接口连接鼠标键盘及摄像头等输入设备，通过 HDMI 接口连接显示设备，同时由于 TX1 存储空间有限，通过 SATA 接口接入了固态硬盘对存储空间进行扩展。由此构成了嵌入式机载跟踪系统的硬件测试环境，通过读入图像序列作为视频数据对跟踪算法的功能及性能进行测试。

为准确测试系统性能，需要利用多种场景的数据集对跟踪系统进行测试分析。本文采用的测试数据集来源于 VIVID 测试平台，数据集的场景以机载传感器平台跟踪地面车辆为主。测试数据分为红外和可见光两类，以图像序列形式提供且对目标进行了人工标注，提供了目标的真实位置等信息，便于对跟踪

性能进行数据分析。课题主要利用两类可见光数据和一类红外数据进行机载跟踪系统的综合测试，三类数据分别介绍如下。

EgTest01 如图 4-2 所示，是一组可见光数据，由 1821 幅 640\*480 像素的 JPG 图像序列构成，描述场景为由 6 辆汽车构成的车队沿跑道行驶。被跟踪车辆为第 5 辆车，且存在相似目标。车队沿车道转变方向后直行，直行过程中被跟踪车辆加速行驶并超过相似车辆，最终到达终点。此组数据无遮挡情况，跟踪难点在于车辆拐弯、加速及有相似目标的干扰，且在直行阶段白色跑道反光严重，目标与背景差异较小，很容易造成跟踪对象丢失的情况，跟踪难度较大。

EgTest04 如图 4-3 所示，是一组可见光数据，由 1833 幅 640\*480 像素的 JPG 图像序列构成，跟踪目标为第二辆车，描述的场景为一组 5 辆汽车组成的车队一条红色道路上行驶，周围土地有离散分布的绿色植被。图像序列有散焦情况，车辆行驶过程中遇到多次树木遮挡。由于机载传感器丢失了一些帧，图像序列中存在完全静止，然后突然产生不连续运动的情况。此组数据存在多处不连续的树木遮挡，且存在散焦、失帧、目标尺寸变化、背景变化等情况，场景较为复杂，跟踪难度大。通常选取部分图像序列用于测试分析。

Pktest02 如图 4-4 所示，是一组红外数据，由 1595 幅 320\*256 像素的 JPG 图像序列构成，跟踪目标为第一辆车，描述的场景为相似车辆依次经过树木及阴影区，短暂直行后进入一段长时间的阴影区域，之后在交叉路口暂停后转变方向，又经过一条有树木造成间歇性遮挡的道路后停止。该组数据存在多处连续或不连续的遮挡区域，且目标与背景或阴影区域的灰度差异小，很容易产生混淆，造成跟踪难度增加。



图 4-2 Egtest01 数据集部分时刻





图 4-3 Egtest04 数据集部分时刻



图 4-4 Pktest02 数据集部分时刻

完成嵌入式机载跟踪设备的研制后,需要通过一系列测试来对系统进行功能性分析及性能指标验证,以评价跟踪系统的整体性能。目标跟踪技术在不断发展,但对跟踪技术及跟踪系统的测试与评估,目前还没有公认的统一标准。课题利用上述三种机载下视场景的图像序列,对动态跟踪过程的各个环节及整体效果从功能与性能两方面进行测试与分析。对机载跟踪系统的功能测试及测试方法如下:

(1)人机交互功能。课题设计了人机交互,便于机载跟踪系统的操控人员进行系统控制及跟踪结果的实时显示。对该功能的测试方法为通过操作界面按



钮开启跟踪功能后, 查看视频及数据的跟踪结果是否在对应区域正确显示。

(2)支持对红外及可见光两种类型的视频数据进行跟踪处理。对该功能的测试方法为分别输入可见光及红外两种类型的视频图像序列, 观察是否能对目标进行准确跟踪, 以验证是否对两种类型的数据都能进行有效跟踪处理。

(3)适应机载下视遮挡场景。本课题针对遮挡问题对目标跟踪方案进行了适应性改进。测试其功能的方法为输入视频图像序列, 通过验证跟踪方案在遮挡的各个阶段的处理能力及整体跟踪情况, 验证抗遮挡跟踪方案的有效性。

性能测试通过模拟各类应用场景对目标跟踪系统的各项性能指标进行测试与对比分析, 本文主要对课题提出的性能指标进行测试分析, 包括以下几个方面:

(1)可跟踪目标的最小尺寸。机载下视场景的目标具有面积小, 特征提取困难等特点, 尺寸越小, 跟踪难度越大。通过测试可执行持续跟踪任务的目标的最小面积, 得到可跟踪目标的最小尺寸, 该指标可以反映跟踪技术的鲁棒性。

(2)可跟踪的最大速度。对该功能的测试方法为通过改变读入图像序列的帧频, 测试可对地面目标进行持续跟踪的最大速度, 以验证算法的实时性。

(3)原始的高速跟踪方案及加入抗遮挡后的跟踪方案在速度及准确率方面的对比分析。本文针对跟踪中的遮挡问题进行了深入研究并设计了抗遮挡处理流程, 要验证其性能, 需要将抗遮挡跟踪方案与原始方案在跟踪速度及准确率方面进行对比分析, 通过速度对比来验证抗遮挡方案对跟踪算法实时性的影响; 并通过可提高的跟踪准确率来验证对遮挡场景的跟踪性能。

此外, 根据系统性能指标, 还需要对可处理的最大图像尺寸、最大跟踪速度等指标进行测试分析。

## 4.2 机载跟踪系统功能测试

在课题研制的嵌入式机载跟踪系统中, 通过人机交互界面控制跟踪系统执行对地跟踪任务, 并显示跟踪结果的测试结果如图 4-5 所示。通过控制功能选择区的按钮开始跟踪任务后, 视频显示区会将跟踪结果实时地显示在界面上, 数据显示区则对目标的位置信息及视频图像的相关信息进行了显示, 并标定了当前已跟踪的帧数。由此验证了机载跟踪系统的人机交互功能。

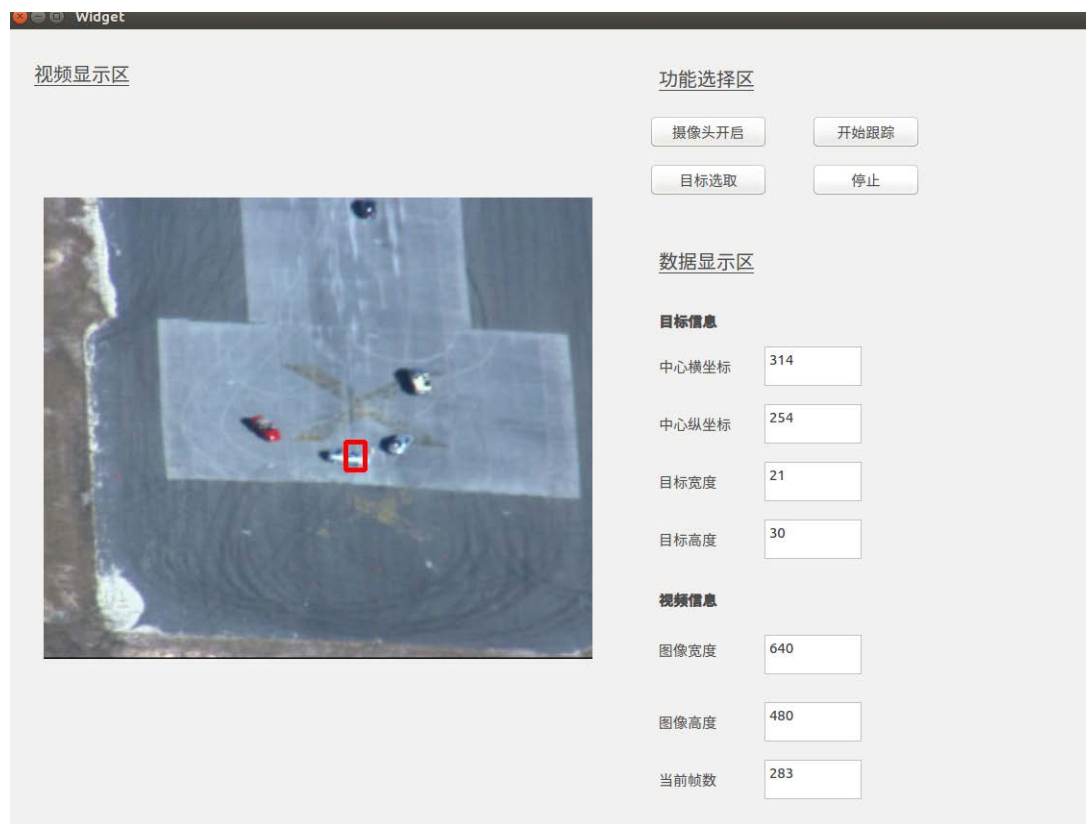


图 4-5 人机交互功能图

机载跟踪系统对可见光及红外两种格式的视频进行跟踪处理的结果分别如图 4-6 及图 4-7 所示。



图 4-6 可见光视频目标跟踪结果



图 4-7 红外视频目标跟踪结果

红外数据由于目标与环境差异小，很容易造成跟踪失败。而从上述跟踪结果可以得出，跟踪方案对红外及可见光数据都有较好的支持，验证了跟踪方案的鲁棒性。

为测试机载跟踪系统对遮挡问题的处理能力，利用红外及可见光两种视频对抗遮挡跟踪方案进行了功能测试，经过遮挡区域各个阶段的跟踪结果如图 4-8、4-9 所示，可以看出抗跟踪方案可以顺利通过遮挡区域，且对红外及可见光视频都能进行处理。



图 4-8 可见光视频遮挡跟踪结果



图 4-9 红外视频遮挡跟踪结果

### 4.3 机载跟踪系统性能测试

根据 2.2.2 节的性能指标，课题从 Egtest04 及 Pktest02 两类图像序列中选取四组具有遮挡区域的特定时段，并结合 Egtest01 无遮挡图像序列共五组实验数据对课题研制的基于嵌入式平台的机载跟踪系统从跟踪速度及准确率方面进行测试，并与原始的高速跟踪方案进行对比，得到的结果如表 4-1 所示。

表 4-1 机载跟踪系统性能测试分析表

测试数据	图像尺寸 (像素)	目标尺寸 (像素)	原始跟踪速 度(帧/秒)	抗遮挡跟踪 速度(帧/秒)	原始准 确率	抗遮挡准 确率
Egtest01 (1-1803)	640*480	30*30	143	142	90%	90%
Egtest04 (1-300)	640*480	15*18	230	225	60%	75%
Egtest04 (600-850)	640*480	32*22	120	110	63%	70%
Pktest02 (1-200)	320*256	37*36	118	120	65%	80%
Pktest02 (700-1300)	320*256	22*29	125	120	90%	89%

本文的跟踪速度指标由跟踪方案处理每帧测试数据的平均时间得出；跟踪的准确率由可成功跟踪的图像帧数占测试数据总帧数的比例得出。分析表中五组测试数据并与系统性能指标作对比可以看出，本文研制的机载跟踪系统可跟



踪的最小目标尺寸为 15\*18 像素，在 20\*20 像素范围内；加入抗遮挡方案后的跟踪速度在 100 帧以上，满足 50 帧/秒的要求；在无遮挡场景下的跟踪准确率在 90%以上，满足指标要求；加入抗遮挡方案后，对有遮挡场景的跟踪准确率在 70%以上，符合系统指标。目前获得的机载下视数据集中图像的尺寸最大为 640\*480 像素，对可处理的最大图像尺寸是通过无人机采集的测试数据进行实验的，经过验证，同样可以达到实时处理的需求。综上，课题研制的机载跟踪系统的各项性能参数均符合系统要求。

#### 4.4 系统综合测试分析

利用表 4-1 中的性能测试数据对原始跟踪方法及加入抗遮挡方案后的跟踪方法从跟踪速度及准确率方面作进一步的对比分析，可以得出以下结论：

(1) 本文提出的抗遮挡方案对跟踪速度影响较小，且能有效提高跟踪准确率。通过高速跟踪方案与抗遮挡跟踪方案的跟踪速度对比可以看出，抗遮挡方案在几乎没有牺牲跟踪速度的情况下，提高了至少 7%的跟踪准确率。基于时空上下文的跟踪方案每秒可达 100 帧以上的跟踪速度，加入抗遮挡方案后在提高目标经过遮挡区域的跟踪能力的同时，仍能保持对目标的高速定位跟踪，可满足机载跟踪系统的实时性要求。

(2) 课题采用的跟踪方案中，影响跟踪速度的主要因素是目标大小，图像大小不会对跟踪速度产生较大影响。通过分析表 4-1 中第 2、3 组数据可以看出，目标越小，跟踪速度越快，目标面积与跟踪速度的变化几乎成比例关系，但目标尺寸的减小会使跟踪准确率相应降低，这是由于与周围环境相比，目标比例过小导致无法提取有效的目标信息造成的。通过分析第 3、4 组数据可以看出，跟踪的目标大小相同而图像大小不同时，跟踪速度几乎相同，由此得出跟踪速度几乎不受图像大小的影响，这与时空上下文跟踪是基于子空间学习的跟踪思想有着直接关系。

根据功能及性能指标，对机载跟踪系统进行综合测试分析可得，本课题研制的嵌入式机载跟踪系统可搭载在无人机等飞行平台实现地面目标的自动跟踪功能；跟踪速度快，可满足实时性跟踪的需求；对场景的适应性好，在可见光及红外背景下都能保证跟踪的准确率；通过对跟踪方案加入了抗遮挡措施，改善了跟踪中遇到遮挡情况的处理能力，提高了系统的跟踪准确性；同时，可以方便地对机载跟踪系统进行控制，系统人机交互友好。综合分析得出，课题研制的机载跟踪系统性能优异，各项指标均满足研制要求，可用于实际场景中

目标自动跟踪任务的执行。

## 4.5 本章小结

本章主要内容为介绍了机载跟踪系统的测试环境,并从功能及性能两方面对系统进行综合测试与分析。首先介绍了机载跟踪系统的测试环境与使用的主要图像序列数据集,之后对机载跟踪系统的抗遮挡能力、场景适应性、人机交互功能及跟踪的速度、准确率等性能进行了全面测试并进行了详细分析,最后得出系统的综合分析结论。

## 结 论

本课题根据项目需求,对机载下视目标跟踪设备的研制从抗遮挡跟踪技术及嵌入式平台的应用实现两方面进行了深入研究。设计了机载跟踪设备研制的总体方案,并针对机载下视环境的遮挡问题对抗遮挡跟踪技术进行了深入研究,设计了抗遮挡跟踪方案,在嵌入式平台进行了应用实现并利用平台的 GPU 资源实现了算法加速,最后通过系统综合测试分析验证了跟踪器的性能。对本文的研究工作总结如下:

(1) 设计了机载跟踪系统的总体研制方案及抗遮挡跟踪方案,并对抗遮挡技术进行了深入研究。课题对机载跟踪系统进行了需求分析,给出系统性能指标并设计了机载跟踪系统的总体研制方案与应用方案。为改善遮挡导致跟踪失败的问题,在对多类跟踪方法进行特性分析的基础上,考虑适应性改进造成算法复杂度提高的问题,从跟踪速度角度出发,选择基于时空上下文的高速目标跟踪技术作为机载系统的跟踪方案。在此基础上设计了抗遮挡方案,从遮挡时刻判断、遮挡区域轨迹预测及遮挡后的目标重捕三方面对抗遮挡技术进行了深入研究并制定了解决方案。

(2) 基于嵌入式 GPU 平台研制了机载跟踪设备。课题基于 CUDA 开发环境,利用 Eclipse 结合 OpenCV 以交叉编译的方式对高速跟踪技术及抗遮挡方案的各个环节进行了技术实现,开发了抗遮挡跟踪方案的应用程序。并针对跟踪器的实时性要求,利用 CUDA 库函数调用平台的 GPU 资源通过并行运算进行算法加速,提高了跟踪速度;在此基础上,基于 Qt 设计了人机交互界面,实现了对机载跟踪器的控制与显示功能。

(3) 测试并分析了机载跟踪系统的整体性能。针对系统的稳定性和可靠性要求,对机载跟踪系统的抗遮挡跟踪、人机交互等功能及跟踪的速度、准确率等性能参数进行了全面测试,测试结果表明,课题研制的机载跟踪系统可对红外及可见光两种场景下的地面目标进行准确跟踪,可处理跟踪过程遇到的遮挡问题,并支持利用人机交互对系统进行远程控制;课题研制的机载跟踪器可跟踪的最小目标尺寸为 15\*18 像素;可达到的跟踪速度在 100 帧/秒以上;遮挡情况的跟踪准确率在 70%以上,比不加入抗遮挡方案提高了至少 7%;非遮挡情况下跟踪准确率在 80%以上,各项指标均符合系统技术要求。

本课题通过研制机载跟踪设备,对目标跟踪技术在嵌入式平台的应用进行了研究并取得了一定成果,对后续研究工作的展望如下:

(1) 拓宽机载跟踪系统的技术研究领域。本课题在嵌入式机载跟踪系统的研制过程中,在算法层面针对目标跟踪技术开展了一系列研究。而机载环境的目标跟踪系统的研究还涉及了电子稳像、目标检测、运动补偿等多种图像技术,是一个复杂的研究领域。由于时间关系,本文的研究工作尚未涉及。后期可通过研究目标检测、视频预处理等技术,提高跟踪系统的性能。

(2) 充分利用 GPU 性能开发应用程序。在对嵌入式 GPU 平台资源的利用方面,本课题通过调用 GPU 加速库函数对跟踪程序的部分环节进行了加速计算,尚未形成完善的 CUDA 应用程序,在调用 GPU 资源进行 CUDA 应用程序的开发方面还有待完善。



## 参考文献

- [1] 刘定通. 复杂背景下视频运动目标检测与跟踪算法研究[D]. 电子科技大学, 2015.
- [2] 肖曦. 航拍视频中的目标跟踪算法研究[D]. 西安电子科技大学, 2009.
- [3] 徐保勇. 机载下视运动目标检测与跟踪技术研究[D]. 国防科学技术大学, 2012.
- [4] Yilmaz A, Shafique K, Shah M. Target tracking in airborne forward looking infrared imagery[J]. Image & Vision Computing, 2003, 21(7):623-635.
- [5] Dale J, Scott D, Dwyer D, et al. Target tracking, moving target detection, stabilisation and enhancement of airborne video[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2005, 5787:154-165.
- [6] 吴繁. 基于 ARM 嵌入式平台的目标跟踪系统的研究[D]. 武汉科技大学, 2009.
- [7] 马翌堃. 基于 GPU 的目标跟踪算法研发与优化[D]. 浙江大学, 2016.
- [8] 刘鹏飞. 基于 GPU 的目标跟踪算法研究与实现[D]. 电子科技大学, 2012.
- [9] 尹宏鹏, 陈波, 柴毅, 等. 基于视觉的目标检测与跟踪综述[J]. 自动化学报, 2016, 42(10):1466-1489.
- [10] 陈全金. 基于视频图像动态目标检测与跟踪算法的硬件实现研究[D]. 电子科技大学, 2012.
- [11] 郭星辰. 空对地复杂背景目标跟踪算法研究——基于 SVM 目标跟踪算法研究[D]. 中国科学院研究生院(长春光学精密机械与物理研究所), 2015.
- [12] 辛哲奎. 基于视觉的小型无人直升机地面目标跟踪技术研究[D]. 南开大学, 2010.
- [13] 刘伟. 基于 ARM 平台的运动目标检测与跟踪系统的研究[D]. 河北工业大学, 2012.
- [14] Davis L, Chelappa R, Rosenfeld A, et al. Visual surveillance and monitoring[J]. Darpa Image Understanding Workshop, 1998.
- [15] 齐会云. 无人机对动态目标高精度定位与跟踪方法研究[D]. 西安电子科技大学, 2015.
- [16] 王勋. 无人机自主跟踪地面运动目标方法与试验研究[D]. 国防科学技术大学, 2012.
- [17] 姜运宇. 旋翼无人机跟踪地面移动目标的视觉控制[D]. 哈尔滨工业大学,

- 2014.
- [18] 罗军. 运动估计在稳像和匹配跟踪中的应用及实现[D]. 国防科学技术大学, 2007.
- [19] Deng M, Guan Q, Xu S. Intelligent video target tracking system based on DSP[C]// International Conference on Computational Problem-Solving. IEEE, 2010:366-369.
- [20] Xu X, Tao R, Wang Y. Real-time infrared target tracking system design and research based on DSP[C]// Multispectral Image Processing and Pattern Recognition. International Society for Optics and Photonics, 2001.
- [21] 孙巧, 张胜修, 张正新,等. 基于加权重叠率的单目标视觉跟踪评价指标[J]. 电子学报, 2017, 45(3):753-761.
- [22] 李文辉. 航拍视频中运动目标的检测与跟踪算法研究[D]. 西安电子科技大学, 2014.
- [23] 钱凯, 陈秀宏, 孙百伟. 一种鲁棒的时空上下文快速跟踪算法[J]. 计算机工程与应用, 2016, 52(12):163-167.
- [24] Gao W, Zhu M, Liu J, et al. Design of real-time target tracking system based on DSP+FPGA[J]. Chinese Journal of Liquid Crystals & Displays, 2014, 29(4):611-616.
- [25] 张永强. 机载光电平台的机动目标跟踪系统研究[D]. 哈尔滨工业大学, 2014.
- [26] Gao X, Mao H, Munson E, et al. Efficient parallel implementation of real-time airborne target tracking system on heterogeneous multicore SoC[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2013, 8713(1):34-37.
- [27] 王欢. 复杂场景下的运动目标检测与跟踪研究[D]. 北京理工大学, 2015.
- [28] Matteucci P B, Byrnes-Preston P, Chen S C, et al. ARM-based visual processing system for prosthetic vision[C]// Engineering in Medicine and Biology Society, Embs, 2011 International Conference of the IEEE. IEEE, 2011:3921-3924.
- [29] 吉淑娇, 朱明, 胡汉平. 基于特征点匹配的电子稳像技术[J]. 中国光学, 2013, (06):841-849.
- [30] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110(3):404-417.
- [31] 沈康辉. 嵌入式 GPU 在超声图像处理中的应用[D]. 哈尔滨工业大学, 2015.
- [32] Chateau T, Chateau T, Gazzah S, et al. A Faster R-CNN Multi-Object Detector

- on a Nvidia Jetson TX1 Embedded System: Demo[C]// International Conference on Distributed Smart Camera. ACM, 2016:208-209.
- [33] 王志民, 徐晓刚. 电子稳像技术综述[J]. 中国图象图形学报, 2010, 15(3):470-480.
- [34] Hegde G, Ramasamy N, Kapre N. CaffePresso: an optimized library for deep learning on embedded accelerator-based platforms[C]// International Conference on Compilers, Architectures and Synthesis for Embedded Systems. ACM, 2016:14.
- [35] 刘万军,董帅含,曲海成. 时空上下文抗遮挡视觉跟踪[J]. 中国图象图形学报,2016,(08):1057-1067.
- [36] Zhang K, Song H. Real-time visual tracking via online weighted multiple instance learning[J]. Pattern Recognition, 2013, 46(1):397-411.
- [37] 贾春肖. 序列图像中的目标跟踪方法研究[D].北京交通大学,2016.
- [38] 石利霞. 机载光电对抗稳定平台目标跟踪干扰技术研究[D].长春理工大学,2009.
- [39] 周鑫,钱秋朦,叶永强,王从庆. 改进后的 TLD 视频目标跟踪方法[J]. 中国图象图形学报,2013,(09):1115-1123.
- [40] 骆荣剑, 李颖, 钱广华,等. 机动目标跟踪中一种改进的自适应卡尔曼滤波算法[J]. 重庆邮电大学学报自然科学版, 2015, 27(1).
- [41] 徐洋, 徐晖, 罗少华等. 基于随机有限集理论的多传感器目标联合检测跟踪算法[J]. 国防科技大学学报, 2013, 35(1):89-96.
- [42] Zhang K, Zhang L, Liu Q, et al. Fast Visual Tracking via Dense Spatio-temporal Context Learning[C]// European Conference on Computer Vision. Springer International Publishing, 2014:127-141.
- [43] Nguyen H T, Ji Q, Smeulders A W M. Spatio-Temporal Context for Robust Multitarget Tracking[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007, 29(1):52.
- [44] 李灿. 基于嵌入式的运动目标检测与跟踪系统的实现[D].成都理工大学,2016.
- [45] 岳兵. 基于 OpenCV 的目标检测与跟踪算法的研究与实现[D].安徽大学,2016
- [46] 薛陈,朱明,刘春香. 遮挡情况下目标跟踪算法综述[J]. 中国光学与应用光学,2009,(05):388-394.
- [47] 贾红宾. 复杂环境中运动目标视觉跟踪研究[D].郑州大学,2011.

- [48] 郭达洁. 监控视频中的前景提取和目标检测跟踪算法研究[D].浙江大学,2016.
- [49] Qin Y, He S, Zhao Y, et al. RoI Pooling Based Fast Multi-Domain Convolutional Neural Networks for Visual Tracking[C]// International Conference on Artificial Intelligence and Industrial Engineering. 2016.
- [50] 张丹. 多帧样本模板更新的视频图像运动目标跟踪算法研究[D].燕山大学,2014.
- [51] Simon X.Yang. Fast-moving target tracking based on mean shift and frame-difference methods[J]. Journal of Systems Engineering and Electronics,2011, (04):587-592.
- [52] 朱鲲. 被动目标自动跟踪与融合技术研究[D].中国舰船研究院,2014.
- [53] 金炫. 复杂场景下的目标自动检测跟踪方法研究[D]. 中国科学院大学, 2014.
- [54] Weng L, Preneel B. A Secure Perceptual Hash Algorithm for Image Content Authentication[C]// Ifip Tc 6 / Tc 11 International Conference on Communications and Multimedia Security. DBLP, 2011:108-121.
- [55] Liu X, Zhang Q, Luan R P, et al. Applications of perceptual hash algorithm in agriculture images[C]// International Congress on Image and Signal Processing. IEEE, 2014:698-702.
- [56] Mahalakshmi T, Muthaiah R, Swaminathan P. Review Article: An Overview of Template Matching Technique in Image Processing[J]. Research Journal of Applied Sciences Engineering & Technology, 2012, 4(24):5469-5473.
- [57] Cox G S. Template Matching and Measures of Match in Image Processing[J]. 1995.
- [58] Vidal F B, Alcalde V H C. Object Visual Tracking Using Window-Matching Techniques and Kalman Filtering[M]// Kalman Filter. InTech, 2011.
- [59] 徐雷果. 机载预警雷达空中多目标跟踪技术研究[D].电子科技大学,2014
- [60] 叶涛. 基于 FPGA 的无人机光电吊舱控制系统设计与实现[D].中北大学,2016
- [61] Radecki K. A real time focused SAR algorithm on the Jetson TK1 board[C]// SPIE Remote Sensing. 2016:1000412.
- [62] Yeh K Y, Cheng H J, Ye J, et al. Constructing a GPU cluster platform based on multiple NVIDIA Jetson TK1[C]// IEEE International Conference on Bioinformatics and Biomedicine. IEEE, 2016:917-922.
- [63] 孟大地,胡玉新,石涛,孙蕊,李晓波. 基于 NVIDIA GPU 的机载 SAR 实时成像处理算法 CUDA 设计与实现[J]. 雷达学报,2013,(04):481-491.

- [64] Yang Z, Zhu Y, Pu Y. Parallel Image Processing Based on CUDA[C]// International Conference on Computer Science and Software Engineering. IEEE, 2008:198-201.
- [65] Thurley M J, Danell V. Fast Morphological Image Processing Open-Source Extensions for GPU Processing With CUDA[J]. IEEE Journal of Selected Topics in Signal Processing, 2012, 6(7):849-855.
- [66] Mathew P S N, Prabakar A S. Certain Approaches of Real Time Object Tracking in Video Sequences on Embedded Linux Platform[J]. International Journal of Innovative Technology & Exploring Engineering, 2013, 2(4).
- [67] ]姚平. CUDA 平台上的 CPU/GPU 异步计算模式[D].中国科学技术大学,2010.
- [68]Deepthi R S, Sankaraiah S. Implementation of mobile platform using Qt and OpenCV for image processing applications[C]// Open Systems. IEEE, 2011:284-289.
- [69] 张雷, 于凤芹. 基于置信图特性的改进时空上下文目标跟踪[J]. 计算机工程, 2016, 42(8):277-281.
- [70] 张雷, 于凤芹. 基于置信图特性的改进时空上下文目标跟踪[J]. 计算机工程, 2016, 42(8):277-281.
- [71]Michalko M, Onuška J, Lavrín A, et al. Tracking the object features in video based on OpenCV[C]// International Conference on Emerging Elearning Technologies and Applications. IEEE, 2017:223-226.
- [72]Kertész C. Enhanced Video Capture Support in OpenCV under Linux[C]// Hungarian Conference on Computer Graphics and Geometry. 2005.
- [73] 杨梅. 基于时空信息的视频目标跟踪算法研究[D].南京邮电大学,2016.
- [74] 文振焜,高金花,朱映映,杜以华,邓良太. 融合时空域变化信息的视频感知哈希算法研究[J]. 电子学报,2014,(06):1163-1167
- [75] 汤贝. 基于模板匹配的目标跟踪算法研究[D].长沙理工大学,2013.
- [76] Patel H. GPU accelerated real time polarimetric image processing through the use of CUDA[C]// Aerospace and Electronics Conference. 2011:177-180.
- [77] Stoica G V, Dogaru R, Stoica E C. Speeding-up image processing in reaction-diffusion cellular neural networks using CUDA-enabled GPU platforms[C]// International Conference on Electronics, Computers and Artificial Intelligence.
- [78] 邓浩,杨菲,潘旭东,游安清. 利用 CUDA 实现上位机的实时目标跟踪[J]. 信息与电子工程,2010,(03):368-371

- [79] 谢尊中. 基于 CUDA 的实时智能视频分析算法研究及应用[D].南京理工大学,2013.
- [80] 张欢庆,高丽,宋承祥. 基于 ARM 的嵌入式 Linux 交叉编译环境的研究与实现[J]. 计算机与数字工程,2012,(02):151-153.
- [81] 金繁,崔培雷. 嵌入式 Qt 中信号与槽机制的研究[J]. 电子设计工程, 2014, 22(24):168-170.
- [82] 朱德新,王爽. 信号和槽机制的研究与应用[J]. 才智, 2011(35):327.

## 攻读学位期间发表的论文及其它成果

### （一）发表的学术论文

- [1] Qiuchen Li, Junbao Li, Cong Hu, Jeng-Shyang Pan. A Framework of Object Tracking Based on STC Algorithm, Electronics Instrumentation & Information Systems , 2017 (EI, 已录用)
- [2] Jun-Bao Li, Qiuchen Li, Jeng-Shyang Pan, Cong Hu. A Framework of Robust Target Tracking Based on Spatio-Temporal Context Learning. [J] Measurement Science Review, 2017(SCI, 已投稿)

## 哈尔滨工业大学学位论文原创性声明及使用权限

### 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于嵌入式 GPU 的机载下视目标跟踪器研制》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：李秋晨 日期：2017 年 6 月 26 日

### 学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1)学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2)学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3)研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：李秋晨 日期：2017 年 6 月 26 日

导师签名：李秋晨 日期：2017 年 6 月 26 日



## 致 谢

行文至此，我的论文已接近尾声，两年的硕士学习生涯即将告一段落。回首这段时光，短暂又难忘，一路走来的点点滴滴一幕幕从脑海中略过，让人感慨良多。藉此论文完成之际，向许许多多生活上给予我关心、工作中给予我帮助、科研中给予我指导的师长、领导、同学、朋友以及亲人表示我最诚挚的谢意，两年来，感谢有你们！

首先衷心感谢我的导师李君宝老师！两年来，李老师一直从生活中给予我最诚挚的关心，从科研工作中给予我最悉心的指导，时刻关注着我的每一步发展，并对我的毕业课题研究给予了最大的支持与帮助！时刻关注着我所研究内容的进度并对研究过程遇到的问题给予耐心指导；同时，李老师在开题、中期及最终答辩的各个环节都最大程度地给予了我帮助，在我硕士生涯两年的研究道路上，凝聚了李老师太多的心血。在为人处事上，李老师为人谦和、平易近人，把我们当朋友一般对待，和李老师相处让人如沐春风；在治学求索上，李老师严谨认真的科研态度、丰富的实践经验、务实的求索精神及踏实肯干的工作作风都深深地影响着我。

感谢研究生两年来陪伴我的室友，沈露、毛晴、唐敏和赵璐瑶同学，虽然处于不同的实验室，但大家一起交流心得、一起欢笑、一起面对困难、互相鼓励，在彼此的帮助与支持下，我们一起度过了这两年半的珍贵时光。特别感谢与我在同一研究所的沈露与毛晴同学，在生活及科研工作中都给予了我无尽的帮助，在我心情烦躁时给予我安慰与耐心开导，最大限度地帮助了我。

感谢崔征师兄、陈修远师兄、王硕师兄及吕旭东师兄，在我的科研工作中给予详尽的指导，从一个先行者的角度在工作及生活中给予了我许多重要的建议，避免我少走了许多弯路。感谢实验室同级的刘环宇同学给予了我很多的帮助，在我研究课题的过程中给予了我无尽的支持，帮助我走过这一路来的磕磕绊绊，同窗之谊弥足珍贵。感谢许剑清师弟对我研究工作的帮助，在我需要遇到技术困难时给予支持，感谢甄玉美和王博雅两位师妹，贴心地关注每一位实验室成员，带动了实验室的活跃氛围，为大家创造了无限欢乐。感谢王冲师弟对我的信任及对我研究工作的支持。

感谢父母两年来对我无尽的包容与支持，他们是我最强有力的后盾，让我能毫无后顾之忧地专注于自己的学业，鼓励着我不断向前。

感谢参与审论文审阅及答辩的各位领导、老师认真审阅我的论文，对我的研究工作提出修改意见并给予耐心指导。