



西南科技大学

Southwest University of Science and Technology

# 本科毕业设计(论文)

题目名称：多通道传感器数据实时采集、  
比对与存储系统设计与实现

学 院 名 称	计算机科学与技术学院
专 业 名 称	软件工程
学 生 姓 名	李婉婷
学 号	5120193260
指 导 教 师	刘涛 副教授

二〇二三年六月

# 西南科技大学

## 本科毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日期：        年    月    日

# 西南科技大学

## 本科毕业设计（论文）版权使用授权书

本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南科技大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

保密☐，在\_\_\_\_年解密后适用本授权书。

本论文属于

不保密☐。

(请在以上方框内打“√”)

作者签名：

指导教师签名：

日期：     年     月     日

日期：     年     月     日

# 多通道传感器数据实时采集、比对与存储 系统设计与实现

**摘要：**在某研究院的复杂试验现场中，存在着大量复杂的管道、法兰盘、高温部件以及实验配套装置，在进行长时间无人监守的试验时，可能因为多种原因导致高温气体或者液体燃料发生泄露，为防止因泄漏引起的试验安全事故，本系统采用传统比对算法对试验现场采集到的图像进行比对筛选，从而给图像分析模型提供模型分析的图像数据。此外，系统会将采集的图像数据存储在系统中供用户回放和复查。

本文介绍了一种基于多通道传感器的实时数据采集、比对与存储系统的设计与实现。该系统使用多个传感器同时采集大量图像数据，然后利用数据比对算法实时比对和处理采集图像数据，最终以不同的文件格式存储处理后的数据。

本文首先介绍了选题的背景和研究意义，并通过对国内研究现状的调查，明确了本课题的研究方向。之后对系统研发所需的关键技术进行了详细描述。紧接着，通过系统的需求分析，确定了项目的功能。本文重点介绍了系统的数据处理过程以及直方图方法，PSNR 峰值信噪比，SSIM（structural similarity）结构相似性，感知哈希算法这几种常见的图像比对算法的实验分析结果，最后经过多重分析，包括时间和比对结果等方面的考虑，本系统确定了采用峰值信噪比算法作为最合适的比对算法。本系统没有用户界面，是在 Ubuntu 上长期运行的系统进程。整个系统分为数据采集、数据比对、数据传输和数据存储四个模块。数据采集模块负责从传感器中读取图像数据；数据比对模块将采集的数据利用峰值信噪比算法预设的阈值进行比对，并在超过阈值时推入训练模型；数据传输模块则是将比对的图片通过共享内存和 FTP 进行传输；数据存储模块将采集的数据存储到本地文件夹中。

根据实验结果，本系统可以实现对多通道传感器数据的实时采集、比对和存储，并表现出较高的准确性和稳定性。

**关键词：**多通道传感器；数据采集；数据存储；直方图方法；峰值信噪比；结构相似性；感知哈希算法

# Design and Implementation of Real-time Acquisition, Comparison and Storage of Multi-channel Sensor Data

**Abstract:** In the complex test site of a research institute, there are a large number of complex pipes, flanges, high temperature components and experimental supporting devices. When conducting a long time unsupervised test, high temperature gas or liquid fuel may leak due to various reasons, in order to prevent test safety accidents caused by leakage, this system uses the traditional comparison algorithm to compare and filter the images collected from the test site, so as to give The image analysis model provides the image data for the model analysis. In addition, the system stores the captured image data in the system for playback and review by the user.

This thesis introduces the design and implementation of a real-time data acquisition, comparison and storage system based on multi-channel sensors. The system uses multiple sensors to collect a large amount of image data at the same time, then uses the data comparison algorithm to compare and process the collected image data in real time, and finally stores the processed data in different file formats.

This thesis firstly introduces the background and research significance of the chosen topic, and clarifies the research direction of the subject through a survey of the current state of research in China. This is followed by a detailed description of the key technologies required for the development of the system. Immediately afterwards, the functions of the project are defined through the requirements analysis of the system. The thesis highlights the data processing process of the system and the experimental analysis results of several common image matching algorithms, including the histogram method, PSNR peak signal-to-noise ratio, SSIM (structural similarity) and perceptual hash algorithm. algorithm as the most suitable matching algorithm. The system has no user interface and is a long running system process on Ubuntu. The system is divided into four modules: data acquisition, data comparison, data transmission and data storage. The data acquisition module is responsible for reading image data from the sensor; the data comparison module compares the acquired data using a preset threshold of the peak S/N algorithm and pushes it into the training model when the threshold is exceeded; the data transfer module transfers the compared images via shared memory and FTP;

and the data storage module stores the acquired data in a local folder. According to the experimental results, this system can achieve real-time acquisition, comparison and storage of multi-channel sensor data, and shows high accuracy and stability.

**Key words:** multi-channel sensor; data acquisition; data storage; histogram method; peak signal-to-noise ratio; structural similarity; perceptual hash algorithm

# 目 录

第 1 章 绪论 .....	1
1.1 研究背景和意义 .....	1
1.2 国内外研究现状 .....	2
1.3 研究目的与内容 .....	7
1.4 论文结构安排 .....	9
1.5 课题实施成果 .....	9
1.6 本章小节 .....	9
第 2 章 需求分析 .....	10
2.1 系统需求分析 .....	10
2.2 系统可行性分析 .....	10
2.3 系统的功能需求及设计目标 .....	11
2.4 相关技术 .....	17
2.4.1 Qt .....	17
2.4.2 OpenCV .....	18
2.4.3 FFmpeg .....	19
2.4.4 海康威视传感器 SDK .....	20
2.4.5 红外传感器 SCT_SDK .....	20
2.5.5 传统比对算法的介绍 .....	20
2.5 本章小节 .....	24
第 3 章 传统比对算法比对结果 .....	25
3.1 数据集来源与说明 .....	25
3.2 传统算法的比对结果 .....	25
3.3 本章小结 .....	29
第 4 章 系统设计与实现 .....	30
4.1 系统整体架构图 .....	30
4.2 系统设计 .....	31
4.2.1 多通道传感器数据实时采集算法设计与实现 .....	31
4.2.2 多通道传感器数据实时比对算法设计与实现 .....	39

4.2.3 多通道传感器数据实时传输算法设计与实现 .....	42
4.2.4 多通道传感器数据实时存储算法设计与实现 .....	44
4.3 长期稳定运行的实现 .....	45
4.4 本章小结 .....	46
第 5 章 系统测试 .....	47
5.1 测试目的 .....	47
5.2 测试方法 .....	47
5.3 测试预期结果 .....	47
5.4 测试实际结果 .....	48
5.4.1 多通道传感器数据实时采集模块 .....	48
5.4.2 多通道传感器数据实时比对模块 .....	48
5.4.3 多通道传感器数据实时传输模块 .....	49
5.4.4 多通道传感器数据实时存储模块 .....	49
5.4.5 长期稳定运行功能 .....	50
5.5 本章小节 .....	50
结论 .....	51
致谢 .....	52
参考文献 .....	53
研究成果 .....	57



# 第1章 绪论

## 1.1 研究背景和意义

目前，科技已经逐渐渗透到了人们生活的各个领域，视频图像已经成为计算机视觉技术不可或缺的组成部分。无论是在日常生活中还是在学术研究中，都需要捕获和存储图像数据，以便进行后续处理<sup>[1,2]</sup>。传统的图像处理方法主要利用计算机视觉，使用模型对图像进行训练以便于自动化处理图像信息，以此来提高图像处理的准确性和效率<sup>[3,4]</sup>。

此外，当前的一些模型经过图像训练可以用来识别某些场景中的特定情况<sup>[5]</sup>。例如，在一个复杂的试验现场环境中，有许多管道、法兰盘、高温部件和配套的试验设备，不同的管道承载着不同的介质，如高温高压气体的管道和液体燃料的管道。在试验过程中，高温气体或液体燃料可能会因为各种原因发生泄漏，如震动、高温变形、法兰盘连接不良或周边不规则、破裂等。试验材料可能因为这些原因而发生泄漏，并引起其他安全事故。因此，为了确保试验期间不会发生泄漏事故，需要持续不断地检测管道等结构是否存在泄漏问题。

目前，试验期间使用机器进行安全检测已成为预防试验期间安全事故的重要方式，通过捕捉试验场地的传感器的图像代替人工观察，进行安全情况分析这种方式可以更加直观地展示出试验场地的安全信息<sup>[6,7]</sup>。同时保存试验过程中的图像和视频也方便在试验结束后回看，检测试验中未被发现的安全问题，进一步提高试验的安全性和准确性。

通过观察图像来检测试验场地的安全需要获取大量的图像数据，传感器可以轻松地获取和共享大量的图像数据；然而，传感器采集的图像数据包含大量重复、相似且冗余的图像信息，分析所有这些图像会增加图像分析模型的负担，因此，需要对传感器采集的图像数据进行处理，以提高图像分析的效率，需要使用适当的算法来合理地过滤获得的图像，从而减少图像分析模型的负荷。

图像比对算法可以应用于图像匹配、图像识别、图像分类、图像搜索等方面，图像比对算法的发展和应用对于提高计算机视觉技术的水平、推动相关应用的发展具有重要的意义。通过研究图像比对算法，可以提高图像识别和处理的准确性和效率，但只有比较精密的图像算法才能比对出两张只有局部存在极其微小差异的图像，但如果一味追求图像匹配算法的准确性，而不去考虑图像算法所花费多少时间，当送入模型的数据密度很高时，则无法及时进行计算，从而导致安全事故无法在第一时间得到处理，所以，选

择一个最优的图像比对算法也尤为重要。传统的图像比对算法有直方图算法，PSNR 峰值信噪比，SSIM（structural similarity）结构相似性，感知哈希算法等。

## 1.2 国内外研究现状

图像采集比对存储目前是计算机视觉领域的一个重要研究方向，涉及图像采集、图像比对和图像存储等多个方面<sup>[8-10]</sup>。

数字图像采集技术是数字图像处理中的重要环节，其主要目的是从外部环境中获取图像信息，为后续的图像处理和分析提供数据支持。

如今，图像采集技术已经相当成熟，并广泛应用于各个研究领域，包括但不限于自动人脸识别、医学图像处理、口罩检测、监控录像、数字图书馆等。但是，随着图像分辨率的提高和采集环境的多样化，图像采集技术也面临着一系列的挑战，包括图像清晰度、采集速度、采集稳定性、采集灵敏度等方面<sup>[11]</sup>。对于那些需要在复杂试验现场等多场景中进行采集的任务来说，多通道传感器的优势尤为明显。此外，单通道传感器的视角受限，单通道传感器只能捕捉传感器正对的视角，而多通道传感器可以在不同的位置放置，以捕捉不同的视角，这样就可以捕捉更加广泛的视角，可以更好地覆盖场景，更好地适应不同的应用场景。除此之外，多通道传感器采集可以让后期更加便捷，在后期处理中可以将不同视角的图像信息进行合并，生成更加全面、准确的图像，从而使得处理的效率更加高效。因此，采用单通道图像系统将难以完成复杂试验场图像数据的实时采集和高速实时比对功能<sup>[12]</sup>。

通过使用基于多传感器的图像采集系统，可以对一个场景进行多角度、多方位、多尺度的拍摄，以提高图像采集的效率和质量。相比传统的单通道图像采集技术，基于多传感器的图像采集系统能够弥补其不足之处。比如由美国南加州大学创建的一个基于多摄像机的高精度全息制作系统 Light stage，是一个基于多摄像机的高精度全息制作系统，主要用于数字化人类形象和物体，这个系统使用多个高速相机捕捉不同角度下的图像，并将它们组合在一起，生成高精度的全息图像<sup>[13]</sup>。斯坦福大学研究基于多摄像机的大规模街景图像采集和处理系统 Google Street View，其采用了数百台高精度相机进行街景图像采集。这些相机包括车载相机、自行车相机和行人相机等，相机的镜头数量和角度位置也经过计算和设计，以确保捕捉到尽可能多的街景图像<sup>[14]</sup>。此外，由 Noah Snavely 研究的 Microsoft Photosynth 是一款基于多相机的图像重建和拼接系统，它可以将来自多个摄像机的图像，自动地拼接成一个完整地、具有立体感的三维模型，Photosynth 系统使

用多个相机同时拍摄同一个场景,并且将这些图像进行匹配和重建<sup>[15]</sup>。除此之外,美国麻省理工学院开发了一个混合现实系统 MIT REAL System,这个系统包括了数十个高分辨摄像机,这些摄像机可以实时捕捉房间内的物体和人体<sup>[16]</sup>。相比于单一摄像机或激光扫描仪,多摄像机系统可以从不同的角度和位置对同一物体进行多次扫描,可以实现更加准确和全面的物体和人体扫描。由国防部资助的 Unmanned Underwater Vehicle 无人潜水项目,在这个项目中,多摄像机系统被用来进行水下探测和监视<sup>[17]</sup>。潜水器上装载了多个高分辨率摄像机,这些摄像机可以捕捉水下物体和环境的图像和视频,并将其传输到地面控制中心进行分析和处理。这些都是多通道图像采集的一些应用,充分地说明了多通道图像采集的应用的广泛、实用性、采集效率高等优点。

在许多场景中,例如复杂试验现场,不仅需要采集图像,还需要对这些图像进行训练。然而,高密度的图像数据会给计算模型带来巨大的计算压力,因此在将图像数据送入模型之前,需要对其进行初步的筛选,并满足一定实时性的要求。这就需要使用图像比对算法来对图像进行筛选。

国内有许多研究领域都涉及到图像比对。在医学图像领域,田娟秀等学者提出的方法是将医学图像进行比对,通过比较不同病例的图像来辅助医生进行诊断<sup>[18]</sup>。这种方法可以减少医生的主观判断,提高医疗诊断的准确性和可靠性。在具体应用中,医生可以将患者的图像与已知的病例图像进行比对,找出相似之处和不同之处,进而确定患者的病情和治疗方案。通过这种方式,医学图像比对在疾病诊断和治疗方案制定方面具有重要的应用价值。在安防监控领域,朱栋栋等学者对不同时间段、不同场景的道路监控图像进行比对来识别出可疑的目标<sup>[19]</sup>。这种方法可以有效提高道路监控的精度和实时性,有助于减少交通事故和其他不良事件的发生。在具体应用中,通过对监控图像的比对分析,系统可以自动识别出可疑的车辆、行人或其他物体,并及时报警通知相关部门或人员进行处理。此外,该方法还可以提高道路监控图像的自动化处理水平,减轻工作人员的工作量,提高工作效率。因此,图像比对在道路监控和公共安全领域具有重要的应用价值。在机器视觉领域,段峰等学者利用多张同一目标的图像,基于计算机视觉技术实现目标跟踪和三维重建<sup>[20]</sup>。首先,采集目标在不同角度下的多张图像,对图像进行特征提取和匹配,得到目标在不同图像中的位置信息;接着,基于这些位置信息,利用三角测量等技术重建出目标的三维模型,并进一步实现目标的跟踪和运动分析。

图像比对在国外也得到了广泛的研究。在地球遥感仪检测领域中,A. A. Zalatoi 等学者提出了对不同空间图片与数字区域进行比较的方法,通过这种方法可以检测出地图

信息的变化和领土的变化,进而实现对地球环境的监测和更新<sup>[21]</sup>。例如,可以利用卫星遥感图像对不同时间的地球表面进行比对,从而观察到地球表面的变化,比如海岸线的变化、森林的变化等,对于环境保护、资源管理等具有重要的意义。此外,该方法还可用于灾害监测,如通过对洪水、地震等自然灾害前后的卫星遥感图像进行比对,可以及时发现灾害的影响范围和程度,从而采取有效的救援和恢复措施。天文学领域是一种高精度的科学,需要对远处天体的图像进行精确的重建和分析。然而,天文图像常常受到观测条件、仪器性能等因素的限制,导致图像分辨率不足。为了解决这个问题,Song Wei等学者提出了一种基于多图像重建的方法,即从低分辨率图像中重建出高分辨率图像<sup>[22]</sup>。该方法通过对多幅低分辨率图像进行比对和组合,利用不同角度、不同时刻的观测数据来提高图像分辨率。通过该方法,天文学家可以更加准确地观测和研究宇宙中的各种天体,从而更好地了解宇宙的本质和演化。

不同应用场景下,图像比对技术所面临的挑战也不同,包括图像质量、图像数量、算法效率等方面。目前的图像比对主要分为传统图像比对方法和基于深度学习的图像比对方法,基于深度学习的图像比对方法虽然比传统的图像比对方法更能处理复杂的图像,但传统的图像比对方法更加简单易用。本项目所涉及到的只是在推入图像训练模型之前的图像筛选工作,因此更适合使用传统的图像比对方法。传统的图像比对方法有直方图方法,PSNR 峰值信噪比,SSIM (structural similarity) 结构相似性,感知哈希算法<sup>[23]</sup>。

直方图是图像特征描述方法之一,它通过计算图像像素的灰度或颜色分布来描述图像的特征<sup>[24]</sup>。刘芳辉、王新余等人提出的基于颜色直方图方法,是一种常用的图像相似度度量方法,也被广泛应用于图像检索、目标识别、图像分类等领域<sup>[25,26]</sup>。该方法首先将图像中的颜色分为若干个离散的颜色级别,然后统计每个颜色级别在图像中出现的像素数量,并将其组成颜色直方图,最后,通过比对不同图像的颜色直方图,计算图像之间的相似度或距离。这种方法简单易行,适用范围广泛,特别是对于颜色分布比较明显的图像,效果更佳。同时,将颜色直方图作为图像特征向量,还可以用于图像的分类和聚类分析。此外,直方图在图像增强和去雾中也有所应用,主要是利用直方图来对图像进行分割和增强处理。谢勇等学者通过使用灰度直方图对图像进行拉伸均匀处理从而提高图像比对度,以此来提高含雾图像的清晰度<sup>[27]</sup>。同时,直方图的应用也是在去雾处理中非常有效的方法,通过计算图像的直方图来推断图像的亮度和色彩信息,从而提取图像特征,进而对图像进行匹配和分类等任务。Zafar Bushra、Nouman Ali 等学者在图像匹配领域进行了深入研究,提出了一些新的方法,以提高匹配的准确性和效率<sup>[28,29]</sup>。通

过引入空间信息的概念,考虑图像中像素的位置和周围像素的信息来确定图像之间的相似度。Kristoffersen S. K.、Giuliani Donatella、Tawhid Md Nurul Ahad 等学者研究了多通道信息方法<sup>[30-32]</sup>,使用多个通道的图像信息来匹配图像,以获得更准确的匹配结果。通过这些方法的应用,图像匹配的准确性和效率得到了进一步的提高,为实际应用提供了更好的支持。

峰值信噪比(Peak Signal-to-Noise Ratio, PSNR)是图像质量评价中常用的一种指标,主要用于评估重建图像与原始图像之间的失真程度,是衡量图像或视频质量的一种常用指标<sup>[33]</sup>。慈睿佳等人在人脸识别中以 PSNR 作为评价指标,以此来衡量重建模糊人脸图像的图像质量<sup>[34]</sup>。该研究旨在解决人脸图像质量下降时对面脸识别精度的影响。由于人脸图像经常受到噪声、模糊等因素的干扰,导致图像质量下降,从而影响人脸识别的准确性<sup>[35,36]</sup>。同时,PSNR 常常被用作比较不同图像编码算法或不同编码参数的效果。Dur e Jabeen 等学者在图像和视频压缩领域中开展了一系列的研究工作,其中包括了使用 PSNR 指标进行图像压缩质量比较以及不同视频编码标准效率的比较<sup>[37]</sup>。JPEG 压缩是一种广泛应用的图像压缩标准,通过舍弃一些像素信息并利用人眼对于颜色细节的不敏感性来减小图像文件大小<sup>[38,39]</sup>。在研究中,Dur e Jabeen 等人使用了 PSNR 作为评价指标来比较不同 JPEG 压缩质量的效果,进而找到一个合适的压缩比例来平衡图像质量和压缩率。除了在图像和视频编码中应用广泛之外,PSNR 在其他图像处理领域也得到了广泛的应用。例如,在图像增强和去噪领域中,PSNR 可以作为一种常用的图像质量评价指标来衡量图像处理后的质量。在这方面,国外的一些学者使用 PSNR 来评估不同去噪算法的效率,以检验它们减小冲激和随机噪声的效果<sup>[40]</sup>。这些研究工作对于提高图像质量和减小图像噪声有着重要的实际意义。

结构相似性(Structural Similarity, SSIM)指的是一种用于测量两幅图像相似程度的指标,其综合考虑了亮度、对比度和结构等方面的信息<sup>[41]</sup>。传统的 SSIM 指标只能处理灰度图像的问题,邓星涛等学者提出了一种基于纹理结构形态和边缘信息的结构相似性指标<sup>[42]</sup>。该方法可以在考虑图像纹理结构的基础上,加入边缘信息,更准确地刻画图像纹理的特征,从而得到更加准确的分割结果。此外,该方法还可以适用于各种不同类型的纹理图像,具有很好的通用性和鲁棒性。国外学者对结构相似性的研究已经持续了几十年,并形成了多个重要的研究领域<sup>[43-45]</sup>。其中,药物设计和生物分子结构分析等领域是应用结构相似性的重要领域之一。在药物设计领域,结构相似性可以用来评估不同化合物之间的相似程度。Gong Enhao 等学者通过比较不同化合物的结构相似性来寻找具

有相似化学结构的化合物,并进行进一步的研究和开发<sup>[46]</sup>。这种方法可以提高药物设计的效率,缩短研发周期,降低研发成本。在生物分子结构分析领域,可以利用结构相似性来比较不同蛋白质或核酸的三维结构。Kasai Ryosuke 等学者通过计算不同生物分子之间的结构相似性来揭示它们之间的相似性和差异性,并研究它们的功能和作用机制<sup>[47]</sup>。这种方法对于理解生物分子的结构和功能具有重要的意义。这些研究对于揭示分子之间的相似性和差异性具有重要意义。

感知哈希算法 (Perceptron Hashing Algorithm) 是一种基于哈希算法的图像加密算法,它可以将图像转换为固定长度的哈希值,并且保证相同的图像生成相同的哈希值,而不同的图像生成不同的哈希值<sup>[48]</sup>。国内的许多学者在感知哈希算法方面进行了相关的研究,其中刘凯等学者将该算法应用于图像加密和压缩<sup>[49]</sup>。在图像加密和压缩中,感知哈希算法能够更加有效地处理大量的图像数据,从而提高图像加密和压缩的安全性和效率<sup>[50]</sup>。相对于传统的加密和压缩算法,感知哈希算法具有更快的运算速度和更高的压缩率,并且可以在保证图像质量的同时进行加密处理,从而保护用户的隐私和数据安全。王林帆等学者将感知哈希算法应用于入侵检测系统中<sup>[51]</sup>。在入侵检测系统中,感知哈希算法可以用于比较当前图像和预设的标准图像之间的差异,从而判断当前是否存在入侵行为。由于感知哈希算法具有快速、准确、实时等特点,因此能够有效地提高入侵检测的效率和准确性,同时也为信息安全领域的发展带来了新的机遇和挑战。Wafa Birouk 等学者将感知哈希算法应用于生物特征的提取,从而来识别输入的指纹图像<sup>[52]</sup>。

图像存储在图像处理中占据着十分重要的地位,将图像存储起来可以方便后续的实验操作如模型训练等。当然,一个好的存储方式可以在尽可能减小图像存储空间的同时,还能提高图像的质量和访问效率,当图片的分辨率过高时,不仅占用空间而且还会增加检索时间,所以在实际中,常常会将图片进行压缩,可以减小图片的尺寸和存储空间,提高图像的访问效率和存储效率。目前,常见的图像文件格式包括 JPEG、PNG、GIF 等。

对于图像的处理,虽然有一些领域采用了多传感器技术,但是普遍看来,更多的是单一的传感器的处理,这给后续的图像处理带来了许多的不方便,此外,目前大多数的图像处理系统都是采集完图像后立即将图像推入图像训练模型之中,这对于轻量级和慢速级的场景没有多大的影响,而本项目涉及海量的图像数据和严苛的实时性,就必须采取一些措施降低图像训练模型的负载量,所以在推入图像训练模型之前需要对图像进行筛选。高精度的筛选需要采用图像比对算法,而传统图像比对算法不分轩輊,但是在面对实际场景中,应该通过大量的实验数据来筛选出最适合当前场景的比对算法。在图像

的存储领域，虽然现在有了许多成熟的压缩技术，但是长时间的实验难免会堆积大量的图像数据，因此需要定期进行清理。与此同时，大量的本地图像检索也成为了一个重要的问题，本项目采用了以目录分类的形式来解决这个问题。

### 1.3 研究目的与内容

本系统旨在解决现场监控图像处理中的一系列问题。由于现场监控图像数据量庞大，直接对其进行分析会使得图像分析模型的负载量过大，从而影响分析模型的准确度。为了解决这一问题，本系统采用了图像数据比对和筛选的方法，通过筛选出与目标相似的图像数据，从而极大减少图像分析模型的负载量，提高其准确度。此外，本系统要求实时性，需要快速地检测出试验场地材料泄漏、管道破裂等异常情况，从而保障试验人员和试验场地的安全。同时，本系统还具有长期稳定运行和在局域网内与不同设备进行通讯的优点，使得其可以广泛应用于各个领域，为各种监控场景提供高效、准确的图像处理和分析服务。

本系统的图像数据来源于共计四个传感器的实时采集，分别是两个光学传感器、一个红外传感器以及一个声学传感器。四个传感器的属性如表 1-1 所示：

表 1-1 传感器属性

属性	光学传感器（1）	光学传感器（2）	红外传感器	声学传感器
采集的数据	图像、视频	图像、视频	图像、视频	视频
帧率	25fps	25fps	25fps	25fps
视频/图像分辨率	1980*1080	2048*1080	640*480	640*480

本系统作为整个工程的服务端，通过网线采集四个摄像头的图像数据，其中，光学和红外是通过 SDK 采集的，声学是通过地址直接获取数据流。与此同时，本系统会与客户端通过 TCP 和 FTP 进行通讯和传输数据，从而完成相应的操作。整个系统的硬件结构图如图 1-1 所示：

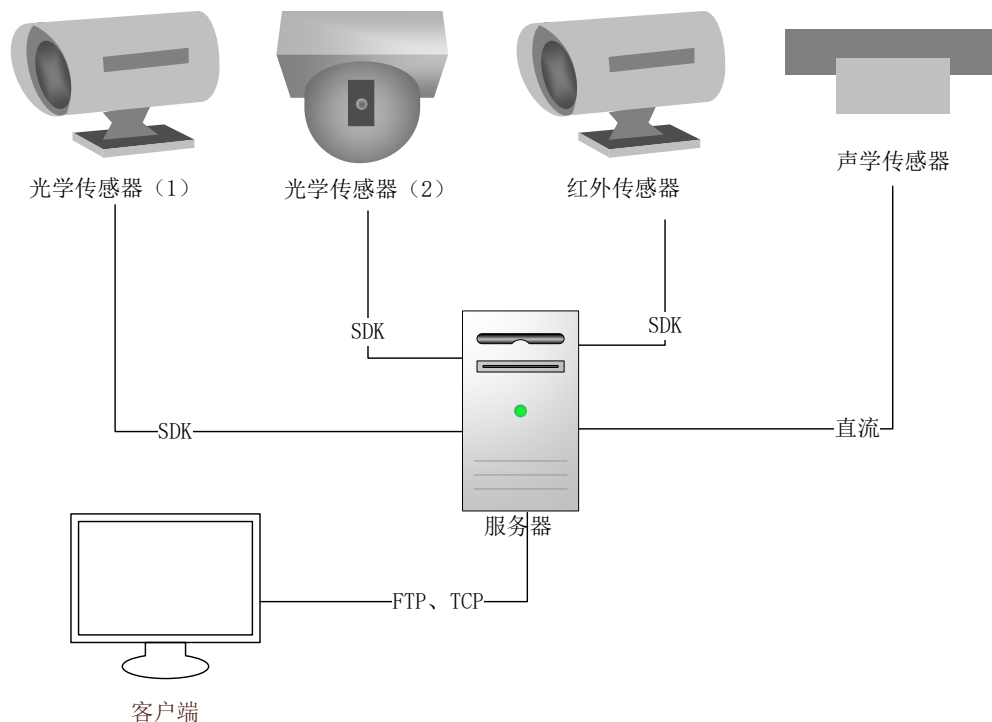


图 1-1 硬件结构图

在设计该软件时，应该具备以下特点：

1、长时间的稳定运行：长时间稳定运行是试验场地监测系统中非常重要的一环，特别是在试验周期较长的情况下更为关键，一旦监测系统发生故障，就可能导致监测数据的丢失和不准确，从而会影响试验的安全性和准确性。

2、可跨平台使用：一个项目可能需要在不同的系统上运行，如果采用同一套采集比对方案，就可以实现跨平台使用，这样可以节省时间和资源，避免不必要的重复劳动。

3、比对精准、速度快：由于实时采集场地中的图像进行比对，接着再传到图像训练模型中去，若是比对不精准就会增大模型的负荷，而若是比对速度缓慢，就无法实时判断所捕捉到的图片场景是否发生了异常。

本项目具体的研究内容为：

- 1、完成实时采集多个传感器大量的图像数据；
- 2、比对各种图像比较算法，找出合适的算法可以做到准确并快速完成对图像数据的比对的算法研究与实现，要求一次图像比较的时间控制在 100ms 内；
- 3、完成图像数据的存储功能，并考虑不同图像数据的不同存储方式以及本地存储的清理工作；



4、完成图像采集、比对、存储的程序设计与实现，并且保证此程序能够实现稳定运行 24 小时以上并与不同进程，不同设备间完成通讯功能。

## 1.4 论文结构安排

本论文以复杂试验现场的安全检测作为具体的研究背景，详细地阐述了图像采集、图像比对及其传统比对算法、图像存储的国内外研究现状。第二章分析了系统的需求，分别从系统的需求分析、可行性和设计目标展开叙述。整个系统可分为数据采集、数据比较、数据传输和数据存储四个模块。另外，还会介绍系统用到的具体的技术，包含编译器、图像处理工具以及传感器接口以及具体描述了几种常见的传统比对算法的原理及其公式。第三章将几种比对算法进行结果比对，选出最适合本文研究背景下的比对算法，最后将结合选出的比对算法设计出本项目的图像比对算法。第四章是本项目的系统设计，详细地介绍了系统的四个模块的具体实现。第五章是本项目的系统测试，对系统所有的功能进行了测试，借助系统测试对系统功能进行检测并完善系统。

## 1.5 课题实施成果

项目自 2022 年 8 月份投入到实际运行中，截至今日，已经采集和处理的视频、图像信息以及项目初步研究成果如表 1-2 所示。

表 1-2 项目实施成果表

安排	进度/成果
投入实际运行的日期	2022.8.10-至今
投入实际运行的时间长度	400 小时
实际处理的总视频数	192000 个
实际处理的总图像数量	36576000 张
公开期刊论文	一种基于峰值信噪比的多通道监控图像采集、比对算法设计
计算机软件著作权	多光学、红外传感器图像采集、存储与比对软件

## 1.6 本章小节

本章是论文的引言部分，介绍了课题的研究背景和意义，以及国内外在图像采集、几种传统图像比较算法、图像存储等方面的研究现状和项目实施计划。

## 第2章 需求分析

### 2.1 系统需求分析

多通道传感器数据采集、比对、存储系统是一个比较复杂的系统，该系统要求实时采集多个传感器的数据并且对图像数据进行实时比对。经充分的项目调研和分析，系统的设计原则如下：

1、 传感器接口：需要确定传感器的接口类型，以便确定采集模块的设计和数据处理方式。

2、 数据采集能力：该系统需要具备多通道传感器数据采集能力，能够处理和转化各类传感器数据。在采集模块中，需要确定采样率和采集速度，并支持高速数据采集，以满足实时性和数据精度要求。

3、 数据比对能力：系统需要将采集到的数据与之前采集到的数据进行比对，并且需要极快的处理时间，从而来保证系统的实时性，所以需要选择适当的数据比对算法，以便发现异常情况或者变化趋势。

4、 数据存储能力：系统需要具备数据存储和管理能力，以便对采集到的数据进行后续分析和回溯。同时，系统需要支持高效的数据检索和查询功能。此外，还需要确定用于存储数据的介质，例如硬盘或固态硬盘，并需要估算数据的存储容量，以确定存储介质的大小和备份策略。最后，还需要确定数据的管理策略，以支持高效的数据检索和查询功能。

5、 系统可靠性：需要设计一个合理的系统框架，确保系统能够在长时间运行的情况下保持稳定和可靠。此外，系统需要支持故障检测和自我修复功能，以便保证数据采集的连续性和稳定性。

6、 系统规范性：规范标准化的开发流程，规范标准化的数据编码不仅能保证系统的稳定性，而且还能使系统具有一定的兼容性，有利于系统的二次开发。

7、 系统安全性：系统需要能够保护采集到的数据不受未授权的访问和攻击，需要采用安全策略，如数据加密、用户认证和访问控制等，以确保数据的保密性和完整性。

### 2.2 系统可行性分析

对多通道传感器数据进行实时采集、比对和存储系统进行可行性分析是确保开发成功的基本前提，可以有效避免系统设计和开发的失败。多通道传感器数据实时采集、比

对与存储系统的可行性分析结果如下所示:

1、 技术可行性: 该系统使用的是成熟的开源库和工具, OpenCV 和 FFmpeg, 这些工具在图像处理和视频处理领域被广泛应用。同时, 采用 Qt 作为编程语言, 具有很好的可移植性, 能够在多种操作系统上进行开发和运行。因此, 从技术角度来看, 该系统是可行的。

2、 经济可行性: 该系统利用开源工具和库进行开发, 从而可以有效地降低开发成本。因为该系统需要实时处理来自多个传感器的数据, 需要使用高性能计算机和存储设备来保证数据采集和处理的速度和准确性。因此, 硬件成本会相对较高。但是, 在满足系统性能需求的同时, 可以考虑采用低成本的硬件来降低成本。因此, 从经济角度来看, 该系统是可行的。

1、 运行环境可行性: 该系统的操作系统是 Ubuntu, 它是一款稳定、安全、可靠且得到广泛用户群体和社区支持的操作系统。同时, Ubuntu 操作系统可以在多种计算机硬件平台上运行, 具有良好的兼容性。因此, 从运行环境的角度来看, 该系统是可行的。

2、 用户需求可行性: 该系统需要满足用户实时采集、比对和存储多通道传感器数据的需求。此外, 由于该系统需要长期稳定运行, 系统需要具有良好的可扩展性和可维护性, 以满足未来可能的需求变化和系统升级。从用户需求的角度来看, 该系统是可行的。

3、 时间可行性: 该系统是实时采集、比对与存储系统, 需要在较短的时间内完成数据的采集、处理和存储。系统使用了 OpenCV 和 FFmpeg 等工具对图像进行处理和压缩, 可以提高数据的处理速度。同时, 由于该系统内的进程可以通过共享内存进行通信, 因此可以提高系统的时间可行性并节约时间。

综上所述, 从技术、经济、运行环境、安全和用户需求等方面来看, 该系统是可行的。然而, 在设计和实现系统时, 需要考虑到系统的性能、稳定性、可靠性和安全性等方面的问题, 以满足实际需求。同时, 需要充分考虑长期稳定运行的需求, 并在系统设计中做出相应的决策。

## 2.3 系统的功能需求及设计目标

在长期无人监守的复杂试验现场中, 会因为某些不可抗的因素造成液体、气体材料的泄露, 从而导致试验安全事故的发生, 通过在试验现场布设可移动的多个传感器来监测试验过程中的场地安全情况, 以此来实时获取传感器的图像数据并进行预处理再推入

图像训练模型之中，整个过程都需要极快的处理速度才能确保数据的可靠性。

在该系统中，可以实时采集多个传感器的图像数据，并将这些图像数据采用传统比对算法进行筛选，最后将这些图像数据存储到本地中。要实现这些目标，需要满足以下设计目标：

- 1、同时采集多个传感器大量的图像数据。由于多通道传感器数据需要实时采集，所以可以通过并行化处理来实现多传感器的同时采集，以提高采集速度和效率。可以考虑使用多线程或异步编程等技术，在系统设计中提高并发性能和效率。
- 2、比对各种图像比较算法，找出合适的算法可以做到准确并快速完成对图像数据的比对的算法研究与实现，要求一次图像比较的时间控制在 100ms 内。可以预先对各种图像比较算法进行实验测试和比对分析，选择出对于当前系统而言，准确性高、且能在 100ms 内完成比对的算法。此外，还需要确保算法的准确性和鲁棒性，避免出现误识别或漏识别等问题。
- 3、完成图像数据的传输功能，对于采集或者比对之后的图像，需要根据系统具体的需求对图像数据进行实时传输，一种是通过 FTP 进行数据传输，一种则是传输给同主机之间的另一个进程，为了实现实时传输的功能，可以使用共享内存的方式对图像数据进行实时传输。
- 4、完成图像数据的存储功能，当图像采集或者比对完成之后，需要立即将需要存储的图像数据存储在本地图录中，并考虑不同图像数据的不同存储方式以及本地存储的清理工作。同时，还需要定期进行本地存储的清理工作，以避免存储空间不足的问题。
- 5、完成系统长期稳定运行的功能，确保可以连续稳定运行 24 小时及以上。
- 6、完成图像采集、比对、存储的程序设计与实现，并且确保此程序能够实现长时间运行并与不同进程、不同设备间完成通讯的功能。可以采用 C++ 等编程语言，并选择合适的框架和库进行开发，比如开源的 Qt 编译器。同时，需要进行系统级的优化和调试，确保程序能够长时间稳定运行，并能够与不同进程、不同设备间进行通讯功能。可以采用 TCP 或者共享内存等方式进行通讯，同时需要确保通讯过程的数据安全性和稳定性。

根据对系统需求的分析，确定了四个主要功能模块，分别为多通道传感器数据实时采集模块、多通道传感器数据实时比对模块、多通道传感器数据实时传输模块和多通道传感器数据实时存储模块。系统的功能结构图如图 2-1 所示：

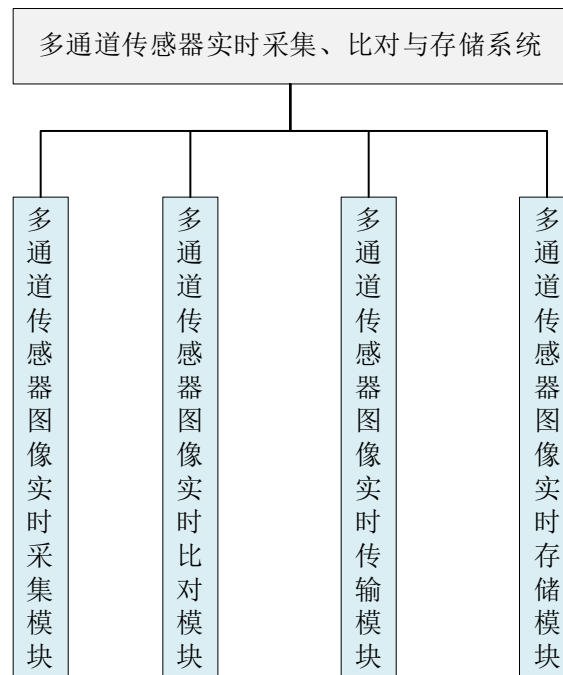


图 2-1 系统功能结构图

### 1、多通道传感器数据实时采集模块

该模块是整个系统的起点，因为未预先存储图像数据，所以系统通过实时采集设备对试验现场进行监控，并通过 SDK 或者直接获取实时流途径来收集采集到的图像数据信息。系统使用的图像数据来自四个传感器，即两个光学传感器、一个红外传感器和一个声学传感器。

其中，两个光学传感器和一个红外传感器是通过 SDK 给程序提供接口的，系统通过 SDK 提供的接口获取到传感器所捕获到的图片数据。这些接口提供了对传感器数据的实时读取和处理功能，可以方便地集成到系统中，为后续的图像处理工作提供了高效的数据来源。

此外，声学传感器是通过使用 OpenCV 连接视频地址获取视频流的。通过 OpenCV 读取声学传感器的地址，系统可以实现对声音信号的实时分析和处理，并将分析结果转换成图像数据，为后续的图像处理和分析提供了有力的支持。

这四个传感器提供了不同类型的数据源，包括光学图像、红外图像、声音信号等，为系统提供了多样化的数据来源。系统可以针对不同类型的数据进行分析和处理，对于采集的红外图像数据的延时应该控制在 1s 之内，采集的光学图像数据的延时在 100ms 之内，以此确保采集的实时性，采集声学传感器受硬件限制，在高频率范围内其延时暂

不做考虑。采集模块的功能结构图如图 2-2 所示：

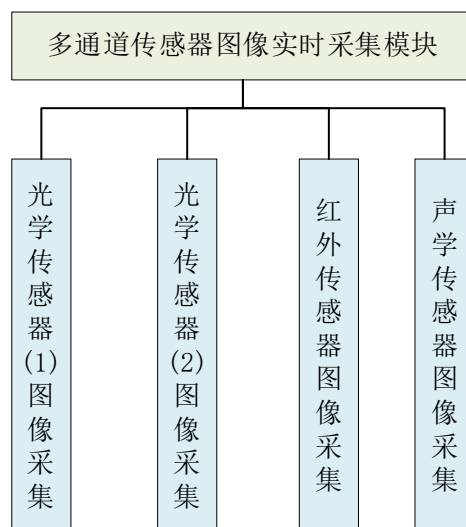


图 2-2 多通道传感器图像实时采集模块功能结构图

## 2、多通道传感器数据实时比对模块

在本系统中，待比较的两张图像为两个光学传感器各自的前后两帧。

在进行图像数据处理之前，保证数据的完整性非常重要，因为在某些情况下，图像数据可能会受到损坏，比如由于传输错误或存储介质故障等原因，如果这些损坏的数据用于算法处理，就会导致最终结果出现偏差或错误。为了避免这种情况的发生，需要在图片数据处理之前对图像的数据完整程度进行检测。

除了数据完整性之外，另一个影响算法比较结果的因素是图像分辨率的差异。在某些情况下，待比较的两张图片的分辨率可能不同，这会导致算法比较结果出现偏差。因此，在进行算法比较之前，需要将待比较的两张图片的分辨率调整为一致。

在对待比较的两张图片进行分辨率调整之后，还需要通过反复试验和设定算法参数来判断图片是否符合当前失真指标。为了确定失真指标以及实现实时比对的功能，可以使用各种图像质量评估技术如直方图算法、结构相似性指标（SSIM）、峰值信噪比（PSNR）、感知哈希等来衡量算法处理后的图像质量。如果前后两张图片的算法参数符合指标，那么就可以判定图片已经发生改变。该步骤是整个算法流程的核心，系统将收集到的图像数据采用合适的算法作比对，以便于后续的处理和分析。当然，在选择图像比对算法时，需要注意算法的准确度以及速度，需要可以检测到肉眼可见的最微小的差异以及实现在 100ms 内完成比对的功能。其功能结构图如图 2-3 所示：

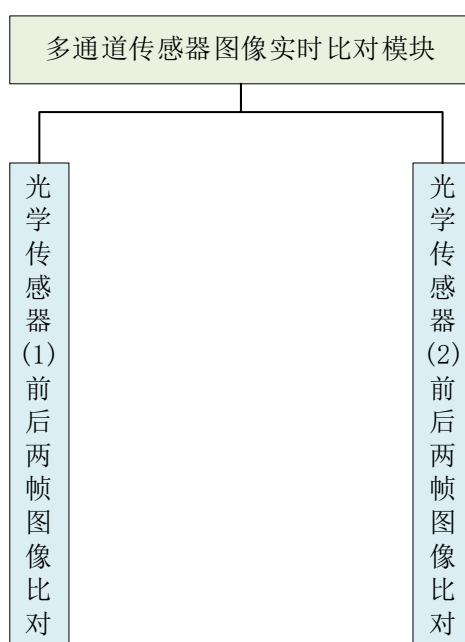


图 2-3 多通道传感器图像实时比对模块功能结构图

### 3、多通道传感器数据实时传输模块

目前，系统处理数据是一项非常重要的任务。为了满足不同的应用需求和场景，系统需要根据具体情况对数据进行不同的处理。其中一种常见的处理方式是上传至共享内存。共享内存是计算机中一种用于进程间通信的技术，它可以让不同的进程共享同一块内存空间，并且共享内存存在传输图像数据时，几乎无延时。将数据上传至共享内存中，可以让不同的进程或系统组件共享这些数据，提高系统的效率和性能，实现实时性。

此外，另一种常见的数据处理方式是使用 FTP 协议传输图像数据。FTP 是一种跨平台的文件传输协议，可用于在不同系统之间进行文件传输。通过使用 FTP 协议，图像数据可以从一个系统传输到另一个系统，保证了数据的完整性和安全性。在整个传输过程中，本系统会根据不同的应用需求和场景进行数据的处理，如上传至共享内存和使用 FTP 传输图像数据，以此来加快传输的速度。

传输模块的功能结构图如图 2-4 所示：

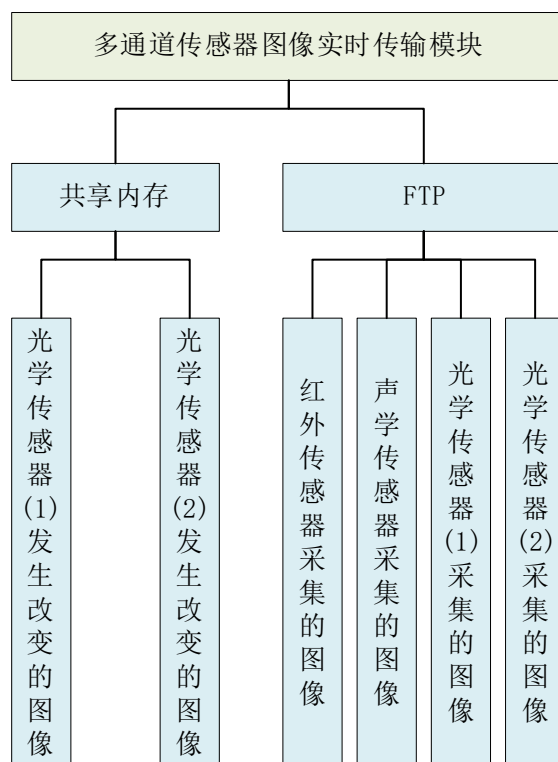


图 2-4 多通道传感器图像实时传输模块功能结构图

#### 4、多通道传感器数据实时存储模块

该模块主要作用是根据比对的处理结果，判断是否需要将数据存储在本地图文件中。

如果判断需要存储，则将处理后的数据信息立即以 JPG 和视频的格式存储在本地图中，以便于后续的查询和分析。

如果系统判断不需要存储该数据，那么就可以将处理后的数据信息从系统中清除，以节省存储资源。这种处理方式可以有效地减少存储资源的消耗，避免数据冗余和浪费。同时，还可以降低数据被非法获取和滥用的风险，保护数据的安全性和隐私性。

除了对数据进行存储和管理之外，系统还会将试验过程的场景以视频的形式记录并存储在数据层中，以便于回溯和分析试验过程。这一功能在试验过程中具有重要作用，可以帮助用户更好地理解试验数据和结果的产生过程，分析各种因素对试验结果的影响，并进一步优化算法和改进试验方案。

在进行存储时，也需要保持其实时性，存储的文件与实际的数据的延时按照不同的传感器有着不同的指标，其中对于红外图像数据的存储延时应该控制在 1s 之内，光学图像数据的存储延时在 100ms 之内，存储的声学图像受硬件限制，在高频率范围内其延时暂不做考虑。该模块的功能结构图如图 2-5 所示：



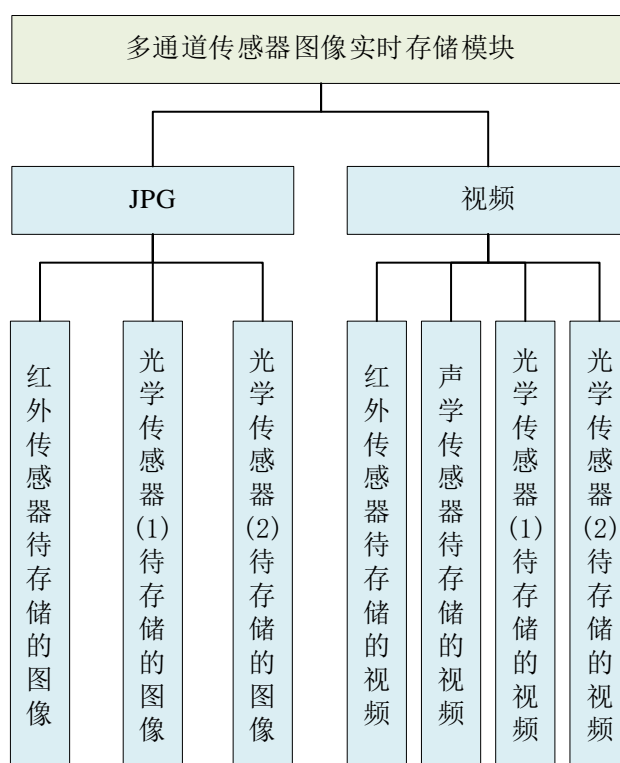


图 2-5 多通道传感器图像实时存储模块功能结构图

## 2.4 相关技术

整个程序的运行平台为 Ubuntu，是基于 Qt 编译器进行的以 C++ 作为编程语言的图像数据采集、比对和存储算法设计。作为服务器上的无界面程序，完成的是对图像数据的采集、比对与存储，并通过局域网建立 TCP 通讯、建立 FTP、开启服务器本机的共享内存来传递各种数据信息。系统引入了 OpenCV 库，其目的是编写各种图像比较算法以找到最优的算法，引入 FFmpeg 实现对视频的编解码。

### 2.4.1 Qt

Qt 是一种跨平台的 GUI（图形用户界面）应用程序开发框架，最初由 Qt Company 开发。Qt 支持开发各种类型的应用程序，包括桌面应用程序、移动应用程序、嵌入式应用程序和游戏等。

Qt 提供了统一的 API，使得开发者可以使用相同的代码在多个不同的操作系统上运行应用程序，而不必重复编写各种不同的代码。Qt 不仅可以在主流的桌面操作系统如 Windows、Linux 和 Mac OS 上运行，而且还可以在移动设备上如 Android 和 iOS 运行。这意味着开发者可以使用 Qt 开发桌面应用程序、移动应用程序和嵌入式应用程序等不

同类型的应用程序，从而扩展了 Qt 的应用范围。由于 Qt 具有强大的跨平台性和易用性，它已经成为许多企业和开发者首选的 GUI 开发框架。其框架如图 2-6 所示：

Qt Application Source Code		
Qt API		
Qt/Windows	Qt/X11	Qt/Macintosh
GDI	X Windows	Carbon
Windows	Unix/Linux	Mac OS X

图 2-6 Qt 跨平台示意图

同时，Qt 源代码完全公开，使得开发者可以深入了解并修改其内部实现，从而满足自己的需求。由于 Qt 是开源软件，全球拥有众多的用户和开发者社区，他们可以共同合作，交流、分享代码、解决问题等等。此外，Qt 是一个非常灵活的开发框架，可以支持多种编程语言，其中 C++ 是主要的编程语言。Qt 使用 C++ 的语法来定义界面元素和控件，并提供了丰富的类库和工具来简化 C++ 编程过程。

除了 C++ 之外，Qt 还支持 Python、JavaScript 等编程语言，这使得开发者可以使用自己熟悉的编程语言进行开发，从而提高开发效率。

Qt 拥有多个类库，其中最重要的是 Qt 核心模块，它提供了基础的跨平台功能支持，例如字符串、容器、文件系统访问等。此外，Qt 还提供了 GUI 类库，可用于创建现代化、美观的用户界面。Qt 的网络类库则提供了各种网络通信功能，例如 TCP、UDP、HTTP、WebSocket 等协议的支持，可轻松实现网络编程。数据库类库则支持多种数据库的访问，例如 SQLite、MySQL、PostgreSQL 等等，开发者可以轻松地进行数据存取操作。另外，图像处理库提供了各种图像处理功能，如图像格式转换、缩放和裁剪等，使得开发者可以轻松地对图像进行处理。这些库可以大大提高开发效率，它使开发人员能够更多地关注业务逻辑的实现，而不必深入研究底层细节的实现。

## 2.4.2 OpenCV

OpenCV 是一种开源的跨平台的计算机视觉和机器学习库，可以用于图像处理、计算机视觉和机器学习等领域的应用程序开发。OpenCV 包含大量的图像处理和计算机视觉算法，例如特征检测、目标跟踪、图像识别和深度学习等。OpenCV 采用 C++ 语言编写，但也提供了接口支持其他的编程语言，如 Python、Java 和 MATLAB 等。OpenCV

提供了各种基础图像处理函数和算法，如图像读写、缩放、旋转、变换、滤波、边缘检测等。此外，OpenCV 还提供了各种特征检测和描述算法，包括 SIFT、SURF、ORB 等，可以用于目标检测、目标跟踪等应用。这些功能和算法可以方便地进行各种图像处理和计算机视觉操作。OpenCV 提供了各种目标检测和识别算法，包括 Haar 分类器、LBP 分类器、深度学习等，可以用于人脸识别、车牌识别、行人检测等应用。OpenCV 提供了各种机器学习算法库，包括支持向量机、决策树、随机森林等，可以用于分类、回归和聚类等应用。OpenCV 将每个模块中的源文件编译成库文件，用户只需添加所需的库文件到自己的项目中并连接到自己的源文件中即可形成可执行程序。OpenCV 的整体框架示意图如图 2-7 所示：

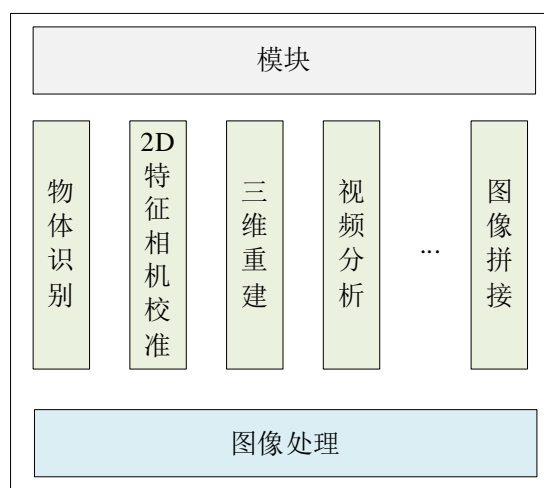


图 2-7 OpenCV 整体框架

### 2.4.3 FFmpeg

FFmpeg 是一款开源跨平台的音视频处理工具库，可用于编码、解码、转码、流媒体传输等多种音视频处理任务。它提供了一套丰富的音视频处理工具和库，其中包括编码器、解码器、过滤器、格式转换器等，可以在多个操作系统平台上使用。FFmpeg 支持多种音视频编解码格式，包括 H.264、MPEG-4、AAC、MP3 等等。它可以将音视频流从一种格式转换为另一种格式，还支持多通道编解码、硬件加速等。FFmpeg 包含了丰富的音视频过滤器，可进行音视频转换、噪音降低、剪辑、水印添加、色彩调整等处理。通过组合不同的过滤器，可以实现各种复杂的音视频处理需求。FFmpeg 可以处理多种音视频封装格式，包括但不限于 AVI、MP4、MKV 和 FLV 等。它可以将音视频流从一种封装格式转换为另一种封装格式，还可以对封装格式进行解析和生成。FFmpeg 支持

多种实时音视频流传输协议,例如 RTMP、HLS、RTSP 等。它可以将本地的音视频流通过网络传输到远程主机,也可以接收远程主机传来的音视频流并进行处理。FFmpeg 支持多种硬件加速技术,包括 Intel Quick Sync Video、NVENC、AMD AMF 等等。使用这些技术可以提高音视频编解码效率,加速处理速度。FFmpeg 能够与许多其他第三方库进行整合,如 x264、libvpx、OpenSSL 等等。这些库可以扩展 FFmpeg 的功能,提供更多的编解码格式和功能。

#### 2.4.4 海康威视传感器 SDK

设备网络 SDK 是一种基于设备专有网络通信协议开发的软件模块,可用于为网络硬盘录像机、NVR、网络摄像机、网络球机、视频服务器、解码器、报警主机、网络存储等设备提供服务。该 SDK 提供了动态库、开发文档和示例程序(C++、Java),可用于二次开发,以实现远程访问和控制设备软件的功能。主要功能包括图像预览、文件播放和下载、PTZ 控制、武装/解除武装、语音对讲、日志管理、解码、远程升级、远程重启/关机、硬盘格式化、参数配置(系统配置、通道配置、串口配置、报警配置、用户配置)、多通道解码器、智能设备功能和访问设备能力设置等。该 SDK 包含网络通讯库、播放库等多个组件,支持 Windows 和 Linux 两个平台的程序调用动态和静态库。

#### 2.4.5 红外传感器 SCT\_SDK

SCT\_SDK 是一种用于连接红外传感器的接口,它提供了一组 API 用于与设备进行网络交互,并进行一些数据操作。其中最常用的功能包括初始化、创建通道、打开通道、获取原始帧等。获取到原始帧数据后,可以对其进行一系列操作,例如测温标记、报警、发送数据、灰度值转温度值等处理,最后将数据存储成图片或视频。SCT\_SDK 支持 Windows 和 Linux 两个版本。

#### 2.4.6 传统比对算法的介绍

##### 1、直方图算法

直方图算法是一种基于像素值的图像匹配方法。它将图像中的像素值分成若干个离散区间,计算每个区间中像素的数量或占比,生成一维直方图向量表示图像。通过比较两幅图像的直方图距离,即直方图向量的差异,来评估它们的相似度或匹配程度。对于两幅图像的像素分布,可以分别计算它们在灰度或颜色空间中的直方图  $A_1$  和  $A_2$  来表示。

为了将直方图  $A_1$  和  $A_2$  归一化到相同的尺度空间，需要将它们分别除以像素总数或归一化因子，以便在不同分辨率或大小的图像之间进行比较。接着，可以通过衡量  $A_1$  和  $A_2$  之间的距离，来比较两个直方图的相似度，以此来评估图像的相似程度。

为了方便图像比较的实现，OpenCV 提供了四种比较方法，包括 Correlation（相关性比较）、Chi-Square（卡方比较）、Intersection（十字交叉性）和 Bhattacharyya distance（巴氏距离）。

直方图比较方法-相关性计算（CV\_COMP\_CORREL）的公式见公式（2-1）所示：

$$d(A_1, A_2) = \frac{\sum_P (A_1(P) - \bar{A}_1)(A_2(P) - \bar{A}_2)}{\sqrt{\sum_P (A_1(P) - \bar{A}_1)^2 (A_2(P) - \bar{A}_2)^2}} \quad (\text{式 2-1})$$

其中， $\bar{A}_K$  的定义见公式（2-2）所示：

$$\bar{A}_K = \frac{1}{N} \sum_G H_K(G) \quad (\text{式 2-2})$$

$N$  表示的是直方图 BIN 的个数， $-1 < \bar{A}_H < 1$ ， $\bar{A}_H$  代表的是均值，当  $\bar{A}_H=1$  时，表示相关性最强。

直方图比较方法-卡方计算（CV\_COMP\_CHISQR）的定义见公式（2-3）所示：

$$d(A_1, A_2) = \sum_G \frac{(A_1(G) - A_2(G))^2}{A_1(G)} \quad (\text{式 2-3})$$

当  $d(A_1, A_2) = 0$  时，图像的相关性越强。

直方图比较方法-巴氏距离计算(CV\_COMP\_BHATTACHARYYA)的定义见公式（2-4）所示：

$$d(A_1, A_2) = \sqrt{1 - \frac{1}{\sqrt{A_1 A_2 N^2}} \sum_G \sqrt{A_1(G) \cdot A_2(G)}} \quad (\text{式 2-4})$$

当  $d(A_1, A_2) = 0$  时，图像的相关性越强。其中， $0 < d(A_1, A_2) < 1$ 。

直方图比较方法-十字计算(CV\_COMP\_INTERSECT)的定义见公式（2-5）所示：

$$d(A_1, A_2) = \sum_G \min(A_1(G), A_2(G)) \quad (\text{式 2-5})$$

## 2、峰值信噪比算法

峰值信噪比（PSNR）是一种常用的客观评估图像质量的指标，它通过比较原始图像和压缩后图像的像素值来计算。在图像压缩领域中，PSNR 被广泛使用，既可以用来评估压缩后图像的质量，也可以作为衡量两个图像相似程度的标准。以一个压缩前的图像  $Q_1$  和压缩后的图像  $Q_2$  为例，PSNR 值可用来衡量图像失真的程度，其单位为对数分贝(dB)。在比较两个图像的相似程度时，PSNR 数值越大表示压缩程度越小，两张图片的相似程度越高；反之，PSNR 数值越小表示压缩程度越大，两张图片的相似程度越低。具体来说，

当 PSNR 值趋近于零时，意味着图像失真非常严重，几乎无法辨别。当 PSNR 值接近最大值时，说明图像质量接近于原始图像而没有受到明显的损失，即图像没有明显的质量劣化。

表 2-1 图像质量参数表

范围	图像质量
$PSNR > 40dB$	图像质量极好
$30dB < PSNR < 40dB$	图像质量好
$20dB < PSNR < 30dB$	图像质量较好
$PSNR < 20dB$	图像质量差

使用 PSNR 可以帮助评估图像的质量。通过 PSNR 值来对图像质量进行等级划分，可以将图像的质量分为不同的等级。表 2-1 展示了不同 PSNR 值对应的图像质量等级。

通过表 2-1 可以得出，当 PSNR 值大于 40dB 时， $Q_1$  与  $Q_2$  的相似度极高；当 PSNR 值在 30dB 到 40dB 之间时， $Q_1$  与  $Q_2$  的相似度高；当 PSNR 值在 20dB 到 30dB 之间时， $Q_1$  与  $Q_2$  的相似度较高；当 PSNR 值小于 20dB 时， $Q_1$  与  $Q_2$  的相似度低。

PSNR 见公式（2-6）所示。

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \quad (式 2-6)$$

信号的最大灰度位数由去噪信号的灰度最大值  $MAX_I$  决定，通常情况下， $MAX_I$  的取值为八位图中的  $N(N=255)$  值。换言之，当  $MAX_I$  的取值不同时，信号的最大灰度位数也会相应地发生变化。

公式（2-6）中，MSE(mean square error)为均方差误差，见公式（2-7）所示。

$$MSE = \frac{1}{cd} \sum_{a=0}^{c-1} \sum_{b=0}^{d-1} \|K(a, b) - F(a, b)\|^2 \quad (式 2-7)$$

公式（2-7）中， $K(a, b)$  和  $F(a, b)$  代表原始图像和重建图像上对应像素的灰度和颜色值， $(c-1)$  和  $(d-1)$  表示有  $(c \times d)$  个像素点。

### 3、 结构相似性算法

2002 年，Wang 提出了一种图像质量客观评估方法，被称为结构相似性法（SSIM），该方法是基于图像结构失真的衡量方式。这种方法在静态的图像质量评价中表现出较好的效果。SSIM 指数是一种基于结构相似性理论的图像质量评价方法，它考虑了亮度、对比度和结构等方面的因素。通过将参考图像与待测图像在以上三个方面进行比较，可以用 SSIM 确定一个评估值，用于判断图像质量的丢失情况。因此，在图像处理中，可以通

过比较待测图像来检测图像是否存在信息丢失，从而确定两张图像之间的相似程度。

SSIM 的公式定义见公式（2-8）：

$$SSIM(O, P) = [A(O, P)]^\alpha \cdot [B(O, P)]^\beta \cdot [C(O, P)]^\gamma \quad (\text{式 2-8})$$

其中  $O, P$  则表示的是输入的参考图像  $O, P$ ，公式（2-8）中的  $A(O, P)$  比较  $O$  和  $P$  的亮度， $B(O, P)$  比较  $O$  和  $P$  的比对度， $C(O, P)$  比较  $O$  和  $P$  的结构，其中  $\alpha > 0, \beta > 0, \gamma > 0$ ， $A(O, P)$ 、 $B(O, P)$ 、 $C(O, P)$  的公式定义分别见公式（2-9）-（2-11）：

$$A(O, P) = \frac{2\mu_O\mu_P + U_1}{\mu_O^2 + \mu_P^2 + U_1} \quad (\text{式 2-9})$$

$$B(O, P) = \frac{2\sigma_O\sigma_P + U_2}{\sigma_O^2 + \sigma_P^2 + U_2} \quad (\text{式 2-10})$$

$$C(O, P) = \frac{\sigma_{OP} + U_3}{\sigma_O\sigma_P + U_3} \quad (\text{式 2-11})$$

$\mu_O$ 、 $\mu_P$  表示图像  $O$  和  $P$  的均值， $\sigma_O$ 、 $\sigma_P$  表示图像  $O$  和  $P$  的方差系数， $\sigma_{OP}$  表示图像  $O$  和  $P$  的过程噪声协方差。 $U_1$ 、 $U_2$ 、 $U_3$  都为常数，其作用是维持  $A(O, P)$ 、 $B(O, P)$ 、 $C(O, P)$  的稳定，SSIM 的值越大，代表两张图片的相似度越高。

在实际应用时，一般会将参数  $\alpha = \beta = \gamma = 1$  以及  $U_3 = U_2/2$ ，演变之后的式子见公式（2-12）：

$$SSIM(O, P) = \frac{(2\mu_O\mu_P + U_1)(2\sigma_{OP} + U_2)}{(\mu_O^2 + \mu_P^2 + U_1)(\sigma_O^2 + \sigma_P^2 + U_2)} \quad (\text{式 2-12})$$

#### 4、感知哈希算法

感知哈希（Perceptual Hashing）算法是一种用于图像相似度比较的算法，它可以将图像转化为一个固定长度的二进制字符串，这个字符串被称为图像的哈希值。通过比较两个图像的哈希值的差异来判断它们的相似度。

感知哈希算法的工作原理是将图像缩小至较小尺寸，接着将其转化成灰度图像。接着，对图像进行一些特定的处理，如离散余弦变换（Discrete Cosine Transform, DCT），取图像的均值等，最后将处理后的结果进行二值化，得到一个固定长度的二进制字符串。

感知哈希的算法步骤如下：

- 1) 缩放：将图像缩放为固定大小的尺寸，如 32x32 或 64x64。
- 2) 灰度化：将图像转换为灰度图像。
- 3) DCT 变换：对灰度图像进行 DCT 变换。
- 4) 取均值：将 DCT 变换后的图像取左上角的 8x8 区域，计算其像素的均值。
- 5) 二值化：将图像中的每个像素与均值进行比较，将大于等于均值的像素设为 1，

将小于均值的像素设为 0，得到一个固定长度的二进制字符串。

## 2.5 本章小节

本章详细阐述了系统的需求、可行性、功能需求和模块设计。通过对技术、经济、运行环境、安全以及用户需求的分析，确定了系统的实现目标。同时，按照系统功能的不同，将其分成了四个模块，包括多通道传感器数据实时采集模块、多通道传感器数据实时比对模块、多通道传感器数据实时传输模块和多通道传感器数据实时存储模块，并对每个模块进行了详细的设计说明。另外，本章还介绍了 QT、OpenCV、FFmpeg、光学红外传感器的 SDK 等系统开发相关技术和工具以及传统比对算法的具体概念和公式。本系统的设计与开发，综合利用了上述技术和平台，完成系统各功能模块的设计和实现。



## 第3章 传统比对算法比对结果

在图像比对模块中，可以通过计算图像的失真程度来判断图像是否发生了变化。某些传统的比对算法可以计算出失真数值。具体而言，本项目会采用比对一些常见的图像比对算法：直方图算法、峰值信噪比算法、结构相似性算法、感知哈希算法，通过比对结果再设计出一个适合本项目的比对算法。

### 3.1 数据集来源与说明

本次算法比对所采用的数据集来源于 TTL 数据集，TTL 数据集是一个用于语义分割任务的大规模图像数据集。该数据集涵盖了 57 种不同类别的物体，包括人、车辆、动物、建筑等。该数据集中包含 12030 张高分辨率图像，每张图像的尺寸为 256x256。每个像素都标注了对应的物体类别，标注采用的是 RGB 颜色编码，可以直接在图像上显示出各类别的分割结果。TTL 数据集收集自不同的互联网来源，如 Flickr、Google、Baidu 等，因此 TTL 数据集中的图像涵盖了多种风格和场景，这有助于提高模型的鲁棒性。

TTL 数据集可用于训练和测试各种语义分割模型，例如 U-Net、FCN 等。这种技术已经在学术界和工业界的各种研究中得到了广泛应用，包括自动驾驶汽车、医疗图像分析等领域。

### 3.2 传统算法的比对结果

首先将 TTL 数据集中 6015 对图像数据使用直方图算法、峰值信噪比算法、结构相似性算法和感知哈希算法进行比较，再分别计算出每种算法比对的最小时间、平均时间和最大时间，比对的结果如表 3-1 所示：

表 3-1 TTL 数据集比较结果

比较算法	最小比较时间/ms	平均比较时间/ms	最大比较时间/ms
直方图	0.061576128005981445	0.08651306809530790000	1.28697443008422850
峰值信噪比	0.0	0.00005008922649836996	0.01567220687866211
结构相似性	0.015570878982543945	0.03271769462579107000	0.06249094009399414
感知哈希	0.0	0.00156510353881125640	0.26556205749511720

由表 3-1 可见，峰值信噪比和感知哈希算法消耗的最小比较时间在四种传统比较算法中为最小。通过对峰值信噪比算法和其他三种算法的比较时间进行分析，可以发现峰值信噪比算法的平均和最大比较时间均较少，因此在比对时间方面表现优于其他三种算

法。

为了进一步证明峰值信噪比算法在时间上的优势，本论文对原有的 TTL 数据集进行了一些处理。原始的 TTL 数据集虽然涉及到各种变化诸如颜色类似、形状类似、风格类型等。但在复杂试验场地下，待比较的两张图片是极其相似的，为了模拟真实的场景，本文将 TTL 数据集中所有图像数据进行了十次模糊变化，模糊程度自 1-10 递增，每一张图像都将分别与自身模糊变化的 10 张图像进行比对，比对完成后，分别对每一次模糊变化的比较时间进行计算，求出其最小比较时间、平均比较时间和最大比较时间。四种传统比对算法不同模糊度时间比较结果如表 3-2、表 3-3、表 3-4、表 3-5 所示：

表 3-2 直方图方法不同模糊度时间比较

模糊程度	最小比较时间/ms	平均比较时间/ms	最大比较时间/ms
1	0.062345743179321290	0.08901394263764570	0.33710837364196780
2	0.061612129211425780	0.08883466712653587	0.34367108345031740
3	0.062322378158569336	0.10097119651629542	0.29676723480224610
4	0.062325954437255860	0.09213573209266314	0.34401893615722656
5	0.062312841415405270	0.08886936200428089	0.23437666893005370
6	0.062455654144287110	0.08946548725105977	0.30198717117309570
7	0.062483549118041990	0.09054881450244019	0.30171132087707520
8	0.062322616577148440	0.08578878214036240	0.38696646690368650
9	0.062454223632812500	0.12542685955044436	0.37461233139038086
10	0.062485694885253906	0.11566878427392924	0.21984696388244630

表 3-3 峰值信噪比不同模糊度时间比较

模糊程度	最小比较时间/ms	平均比较时间/ms	最大比较时间/ms
1	0.0	0.000061737292028920200	0.015669584274291992
2	0.0	0.000049912959262914490	0.015631437301635742
3	0.0	0.000059349974888320380	0.015666246414184570
4	0.0	0.000042876341099156410	0.015634059906005860
5	0.0	0.000061227436969405100	0.015627622604370117
6	0.0	0.000050012924625590160	0.015661478042602540
7	0.0	0.000045666611402706610	0.015654563903808594
8	0.0	0.000051069299280891195	0.015650987625122070
9	0.0	0.000075959604379046680	0.015624046325683594
10	0.0	0.000084799285343262120	0.015626668930053710

表 3-4 结构相似性不同模糊度时间比较

模糊程度	最小比较时间/ms	平均比较时间/ms	最大比较时间/ms
1	0.016731500625610350	0.032507732086942680	0.140336275100708000
2	0.015617847442626953	0.032220846994262090	0.050333261489868164
3	0.017735958099365234	0.032855027255076524	0.055192232131958010

表 3-4（续）

模糊程度	最小比较时间/ms	平均比较时间/ms	最大比较时间/ms
4	0.015616655349731445	0.032241015541285944	0.062544584274291990
5	0.015619039535522461	0.031982593901040450	0.051436185836791990
6	0.015616178512573242	0.031914603779539900	0.052972078323364260
7	0.015618801116943360	0.032372671231961910	0.124351978302001950
8	0.015617132186889648	0.031910818651727310	0.072798490524291990
9	0.015616893768310547	0.033576020813939580	0.111291170120239260
10	0.015672206878662110	0.033189566692311070	0.049175024032592770

表 3-5 感知哈希不同模糊度时间比较

模糊程度	最小比较时间/ms	平均比较时间/ms	最大比较时间/ms
1	0.0	0.0016028173546541362	0.359863758087158200
2	0.0	0.0014489348293440796	0.015669345855712890
3	0.0	0.0013932498018640533	0.016400098800659180
4	0.0	0.0016081351393575185	0.015675306320190430
5	0.0	0.0018080730787040982	0.015685081481933594
6	0.0	0.0015177920968554362	0.016193866729736328
7	0.0	0.0015438600668586102	0.017095565795898438
8	0.0	0.0021885640008788452	0.025926828384399414
9	0.0	0.0021020076716828130	0.016301870346069336
10	0.01567220687866211	0.0331895666923110700	0.049175024032592770

除此之外，分别将这四种传统比对算法十次模糊变化的时间比对绘制成了折线图，如图 3-1 所示：

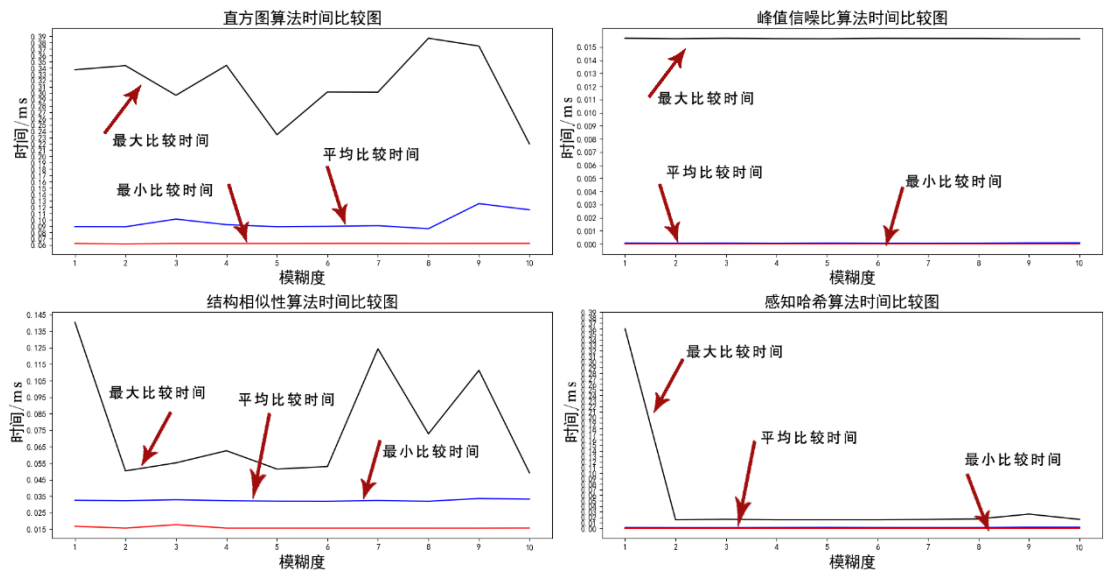


图 3-1 传统比对算法不同模糊度时间比较折线图

由表 3-2、表 3-3、表 3-4、表 3-5 以及图 3-1 可以看出，峰值信噪比算法在十次模

糊变化中，无论是最小比较时间还是平均比较时间或是最大比较时间，都是四种传统比对算法中耗时最少的比对算法，这也进一步说明了峰值信噪比在时间上是四种传统比对算法最优的算法。

当然，衡量一个算法的优劣不能从一个方面进行比对，本系统是建立在复杂试验场地的背景下的，采集的图像用于检测试验材料是否发生泄漏，在要求实时性的同时，也对算法的准确性有要求。在上述对 TTL 数据集所有数据进行十次模糊变化的基础上，求出每次模糊变化每个传统比对算法的最小结果值、平均结果值和最大结果值。其统计结果如图 3-2 所示：

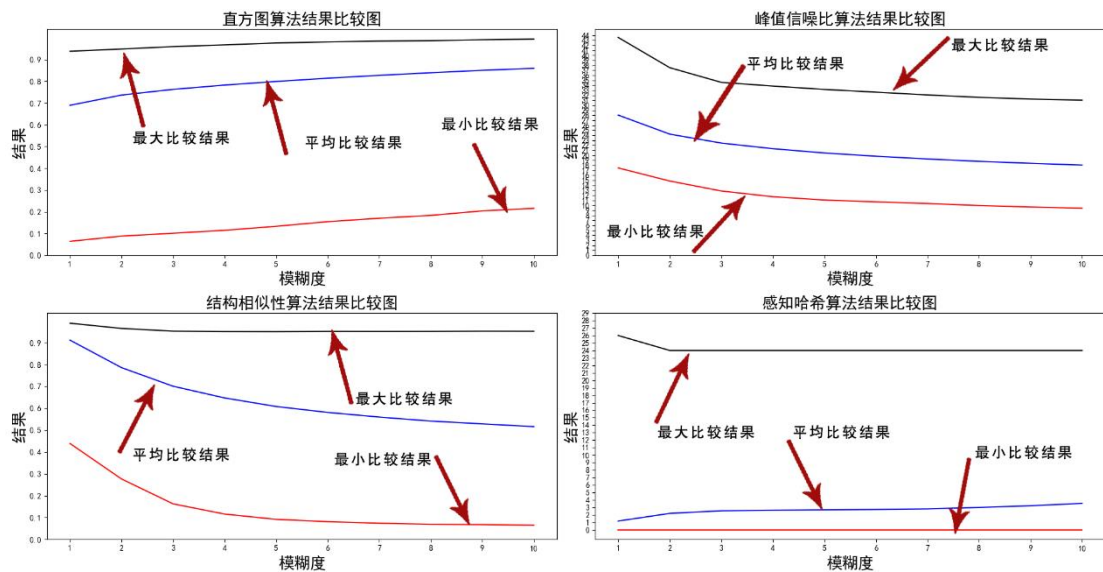


图 3-2 传统比对算法不同模糊度结果比较折线图

表 3-6 展示了四种比较算法平均结果标准差、最小结果标准差和最大结果标准差。标准差越小，表明数据变化越小；标准差越大，表明数据变化越大，从表 3-6 中可以看出，峰值信噪比无论从平均结果标准差、最小结果标准差还是最大结果标准差比较，都是四种比对算法标准差最大的比对算法，由此说明，峰值信噪比算法的结果变化比较大，峰值信噪比算法能比较更多差异性的图片，其包容性更强。

表 3-6 传统比对算法不同模糊度结果标准差

算法	直方图算法	峰值信噪比算法	结构相似性算法	感知哈希算法
平均结果标准差	0.0513019908	2.9523183751	0.1211982646	0.6045847817
最小结果标准差	0.0483746280	2.4521533559	0.1160106327	0.0
最大结果标准差	0.0179348419	3.6403941073	0.0114624351	0.6

经过大量试验验证，峰值信噪比算法在速度和精度上均优于其他三种传统比对算法。特别适用于某些复杂实验现场中的多通道传感器实时采集图像数据的快速分析和比对。

同时，通过图 3-2 可以得出，在模糊程度为 1 时，峰值信噪比的平均阈值约为 28，因此，可以将本系统的峰值信噪比算法的阈值设为 28，作为判断待比较的两张图像是否发生变化的指标。

### 3.3 本章小结

本章使用 TTL 数据集对四种传统算法的性能进行了比对，比对的依据为比对结果的标准差和比对所用的平均时间、最小时间和最大时间。总共有两次比对，第一次是利用 TTL 数据集 6015 对数据对四种传统算法的平均时间、最小时间和最大时间进行了比对，第二次将 TTL 数据集 12030 张图片分别进行 10 次的模糊化处理，再分别将未处理的图片和模糊化后的图片进行比对，最后，再将每个算法的平均比对时间、最小比对时间和最大比对时间以及比对结果的标准差进行比对。本章通过多种比对方式来确定出最适合本系统的比对算法为峰值信噪比算法。

## 第4章 系统设计与实现

本系统可分为多通道传感器数据实时采集、多通道传感器数据实时比对、多通道传感器数据实时传输、多通道传感器数据实时存储这四个模块，其中数据的比对根据上文中的传统图像比对算法结果分析得出的结论进行算法设计。

### 4.1 系统整体架构图

本系统被设计为两个层次：服务层和数据层。服务层主要负责系统中各个模块的交互以及图像数据的采集、比对和传输。数据层则是存储系统中的所有需要的图像数据。

本系统主要应用于复杂试验现场多通道试验图像的监控和管理，旨在快速地对采集到的图像数据进行比较和筛选，以便于后续的处理和分析。比较的结果不需要直接显示给用户，而是作为下一步操作的输入。

由于该系统的应用场景较为特殊，所以并没有应用层来显示结果，该系统的重点在于图像数据的采集、处理和储存，以确保数据的可靠性和有效性。

该系统可通过多种方式与其他系统集成，以便更高效地处理和利用采集到的图像数据，从而在实际应用中发挥更大的作用。同时，该系统也可以进行定制化的开发，以适应不同的应用需求和场景。系统框架图如图 4-1 所示：

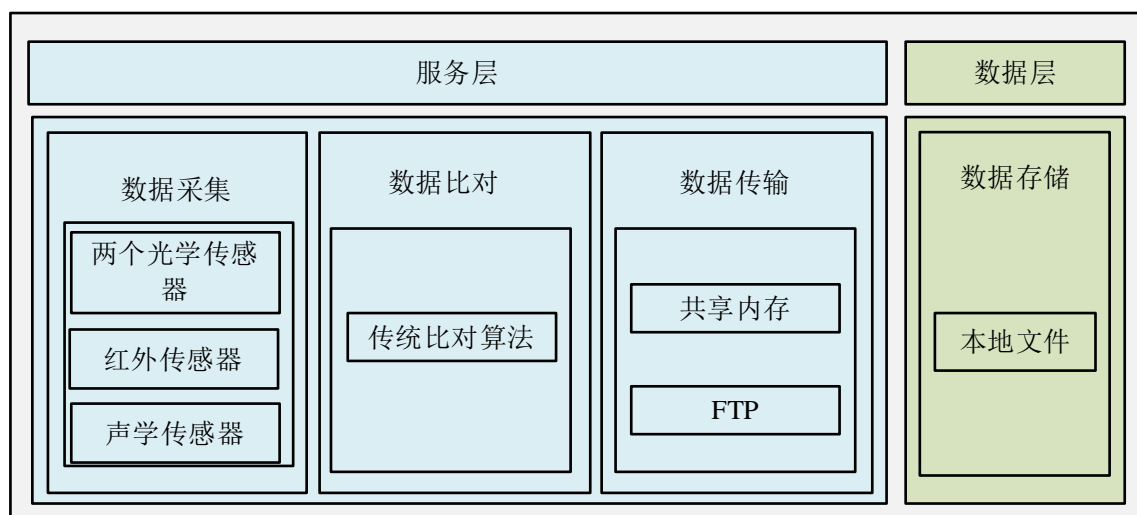


图 4-1 系统整体框架图

由系统整体框架图可以看出，整体框架图显示该系统分为服务层和数据层两部分。其中，服务层旨在快速高效地处理和管理多通道试验现场监控图像采集的数据，以便于后续的分析应用。通过采集、比对和传输图像数据信息，系统可以实现对图像数据的

智能化管理和应用,从而为科研和工程领域提供更加高效和可靠的数据支持。而数据层则是对数据进行存储管理,本系统的数据采用的是本地存储的方式,所以在设计时应考虑到检索的效率和磁盘空间剩余容量的问题,在设计时,采用了规范化的目录分层进行数据存储,另外,会定期删除陈旧的图像数据,以便于新的数据有足够的存储空间进行存储。

## 4.2 系统设计

### 4.2.1 多通道传感器数据实时采集算法设计与实现

本系统在采集图像模块运用了两种方式,一种是通过 SDK 接口采集,一种是通过 OpenCV 直接采集视频流数据。

#### 1、海康威视 SDK

虽然本项目是一个无界面的项目,但是海康威视的抓图和录像接口需要通过预览模块所获得的句柄,所以,在抓图和录像之前需要开启预览。首先需要初始化 SDK,初始化完成后需要设置超时时间以便运行过程中传感器发送连接异常消息时进行设备重连,然后需要设置接收异常信号的回调函数来解决连接或者抓图等过程中所发生的各种异常情况。若初始化失败,系统将进行多次初始化直至成功为止。当初始化完成后,需要注册用户,通过从 INI 文件中读取设备的用户名、密码、端口、IP 等信息进行注册,采用外部文件 INI 也有利于后续的修改,提高代码的扩展性,若注册不成功,就会重新注册,直到注册成功。注册成功后,便会启动预览,如果预览不成功,便会再次尝试预览直到预览成功为止,成功预览后会获得一个句柄,通过这个句柄,可以读取出传感器的界面信息,从而成功通过句柄进行抓图和录像。该系统通过定时器来完成持续抓图和录像,录像时先启动开启录像的定时器,其时间设置为所需录制视频的时间。当定时器到达设定时间后,便会调用停止录像的函数。在程序关闭之前,需要停止预览、注销设备和释放 SDK 资源,如果缺少这些操作,会导致登录的用户一直在线,当在线用户过多时,便无法再进行登录注册连接传感器。图 4-2 是使用海康威视 SDK 抓图录像的流程图:

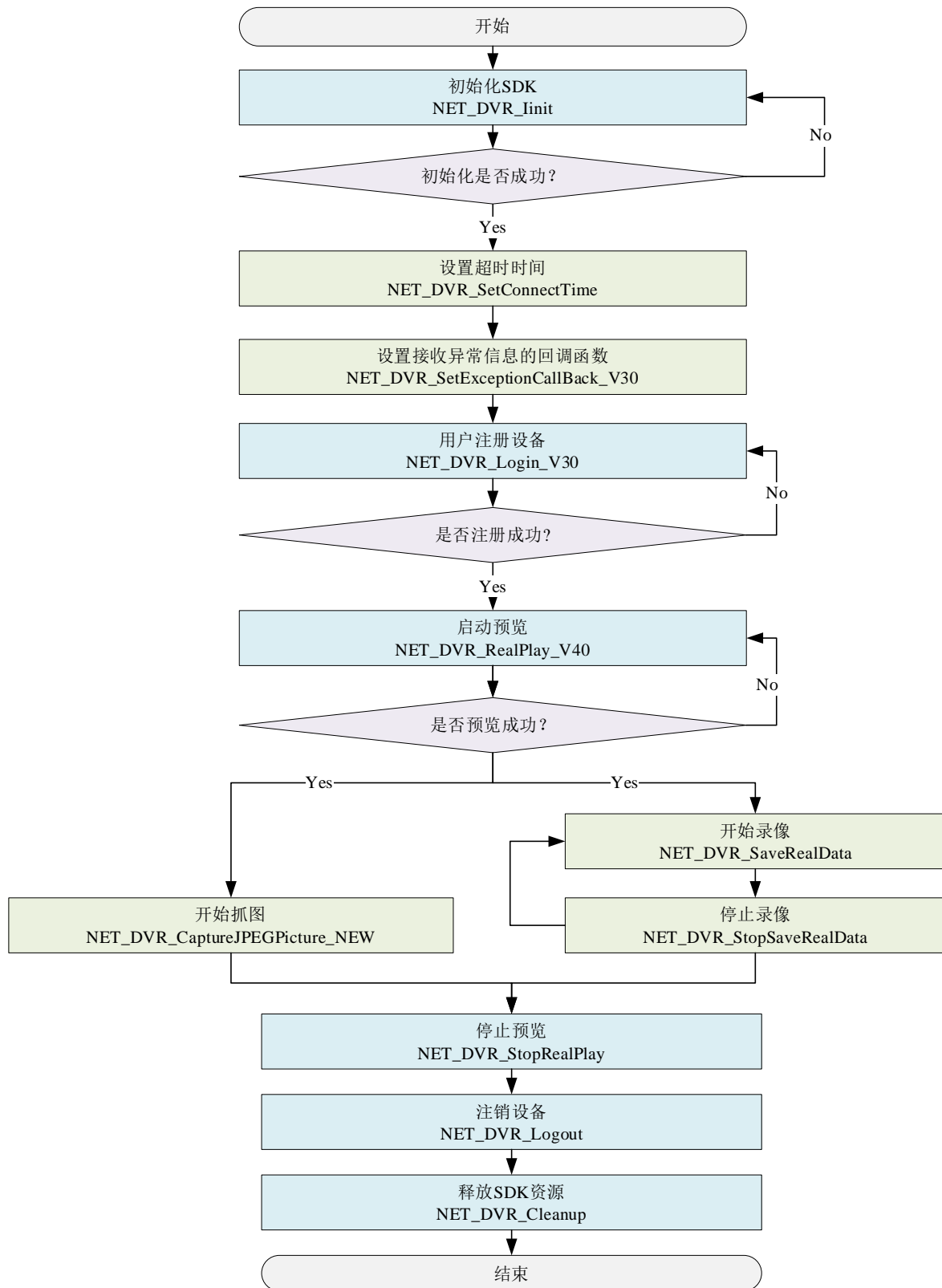


图 4-2 海康威视抓图录像流程图



## 2、 红外传感器 SDK

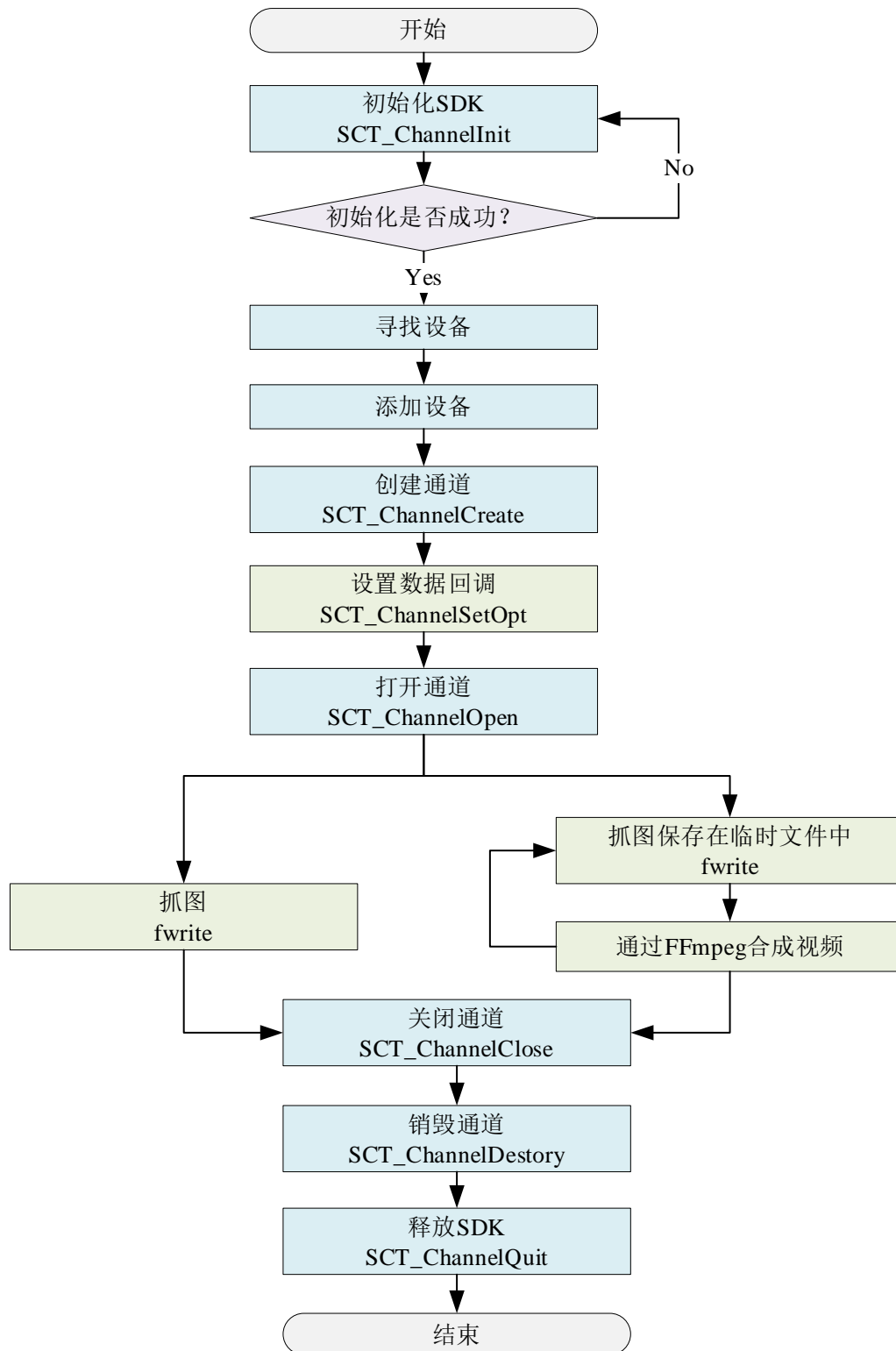


图 4-3 红外 SDK 抓图录像流程图

红外传感器同海康威视的传感器一致,需要先初始化 SDK。不一致的是,红外传感器不需要注册用户。用户名、密码、端口和 IP 会在后续的开启通道时获取到,初始化失败也会重新初始化直至初始化成功。初始化成功后会创建通道,获得句柄,通过句柄获取图片数据。创建通道后会设置回调函数,该函数决定了采集的图像格式,本项目采集的图像保存为 JPG 格式,此回调函数还负责确保后续代码中持续采集图像的功能。在设置好回调函数后,便开启通道,成功开启通道后,便会开始采集图像,通过创建通道时得到的句柄获取图像数据,再通过 fwrite 函数将图像数据以 JPG 形式保存。值得注意的是,红外 SDK 并没有提供录像的接口,所以先用一个临时文件夹存储一个视频所需要的所有图像数据,当采集完一个视频所需的所有图像后,再使用 FFmpeg 将图像合成视频,合成后,再删除临时文件。在关闭程序前,与海康威视一致,需要关闭通道、销毁通道和释放 SDK 资源以防在线用户过多导致无法连接传感器。图 4-3 是使用红外 SDK 抓图录像的流程图。

### 3、 OpenCV 直连视频流

声学传感器通过 OpenCV 读取视频流获取视频图像数据,需要先成功打开视频地址。如果打开失败,系统将重新打开视频地址直至成功为止。

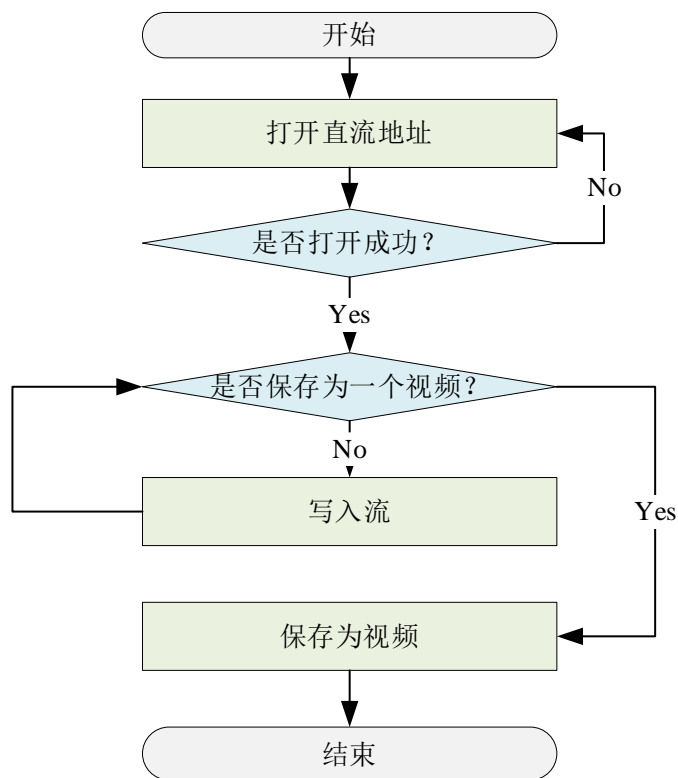


图 4-4 声学传感器视频保存流程图

打开成功之后便可以对读取到的视频流做出相应处理。在本系统中，声学传感器不需要保存图像，只需要保存视频，因此对读取到的视频流通过一个 while 循环保存相对应的流时长视频。其流程图如图 4-4 所示。

声学传感器保存视频的关键代码如图 4-5、图 4-6 所示：

```
const String paths = path.toStdString();
capture.open(paths);
while(1){
    while(!capture.isOpened()){
        usleep(1000);
    }
    SaveVideo();
}
```

图 4-5 打开实时流关键代码

```
void ACOUSTICS::SaveVideo()
{
    QString debug_text = QString("%1 Channel :Start Saving the Video of Acoutics").arg(idnumber);
    Debug_Print1(debug_text);
    writelog(path_log[idnumber],debug_text,Level2);
    QDateTime time = QDateTime::currentDateTime();//获取系统现在的时间
    QString str = time.toString("mm-ss_zzz");//设置显示格式
    QString Filename = path_video[idnumber] + str + ".mp4";

    char* outputvideo = nullptr;
    QByteArray ba = Filename.toLatin1();
    outputvideo = ba.data();

    Size size = Size(capture.get(CAP_PROP_FRAME_WIDTH),capture.get(CAP_PROP_FRAME_HEIGHT));
    VideoWriter wri(outputvideo,VideoWriter::fourcc('m','p','4','v'),capture.get(CAP_PROP_FPS),size,true);
    Point p = Point(10,400);
    Mat frame;
    numbers = video_time[idnumber]*capture.get(CAP_PROP_FPS)/1000;
    qDebug() << numbers;
    int temp = numbers;
    while(numbers > 0 && capture.read(frame)){
        wri.write(frame);
        numbers--;
    }
    wri.release();
}
```

图 4-6 保存视频关键代码

#### 4、 传感器实时采集

为了实现同时采集多个传感器的实时数据的功能，本系统需要采用多线程并行采集技术。在 Qt 中，常常采用信号槽去管理多线程。在本系统中，当接收到开始采集信号时，便开始实时采集多个传感器图像数据，当接收到暂停采集信号时，便暂停采集多个传感器图像数据，直至下一次接收到开始采集信号时，又开始采集图像数据。在每一次采集图像数据前，应该确认各个摄像头处于正常连通状态，以防止发生因传感器并未连接成功就采集图像和传感器中途连接断开从而导致无法采集图像数据的情况。传感器实时采集的流程图如图 4-7 所示：

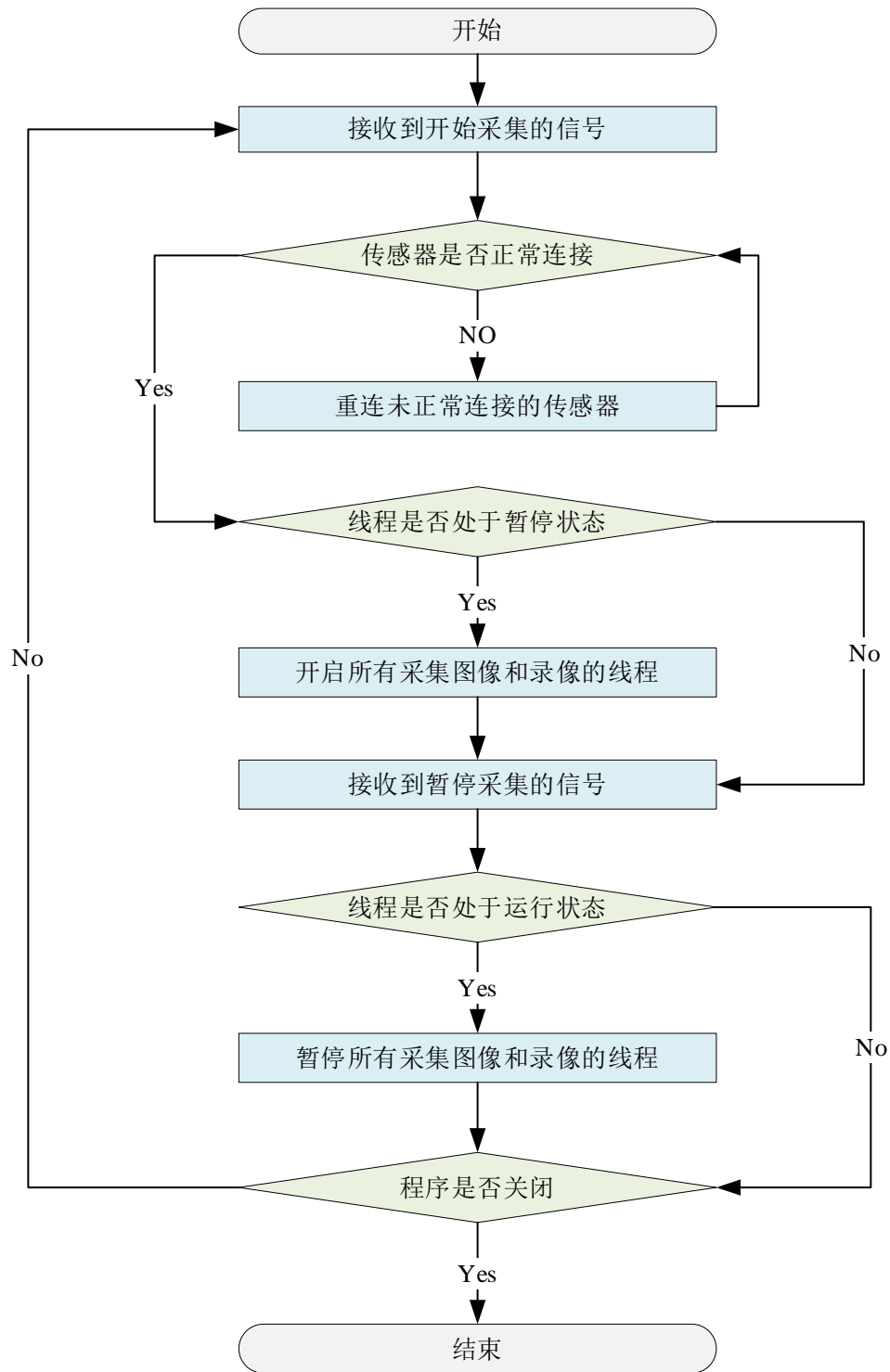


图 4-7 传感器实时采集流程图

传感器的实时采集关键代码如图 4-8 所示：

```

void Control::StartThreadOfConnectTcp()
{
    client = new Client();
    ConnectThread = new QThread();
    connect(client,&Client::SendTcpSocketStat,this,&Control::ConnectSeverStatSlot);
    connect(client,&Client::ProcessStart_HIK1,this,&Control::Restart_HIK1);
    connect(client,&Client::ProcessStart_HIK2,this,&Control::Restart_HIK2);
    connect(client,&Client::ProcessStart_SCT,this,&Control::Restart_SCT);
    connect(client,&Client::ProcessEnd_HIK1,this,&Control::Pause_HIK1);
    connect(client,&Client::ProcessEnd_HIK2,this,&Control::Pause_HIK2);
    connect(client,&Client::ProcessEnd_SCT,this,&Control::Pause_SCT);
    connect(client,&Client::ProcessStartAcoutics,this,&Control::StartAcoutics);
    connect(client,&Client::PowerOff,this,&Control::PowerOff);
    connect(this,&Control::SendAllCameradown,client,&Client::SetAllCameraStat);

    connect(client,&Client::ProcessStartWork,this,&Control::SendIsAllok);
    connect(client,&Client::ProcessdeleteDisk,this,&Control::DeleteDisk);
    connect(this,&Control::TcpStart,client,&Client::InitConnect);
    connect(this,&Control::SendToTcpToSendTextsignal,client,&Client::SendTextToSever);
    //将对象绑定到线程中去
    client->moveToThread(ConnectThread);
#ifdef TCP
    qDebug() << "This is Tcp";
    ConnectThread->start();
#endif
    //开始建立连接
    emit TcpStart();
}

```

图 4-8 传感器实时采集关键代码

图 4-9 和图 4-10 展示的是接收到开始采集图像和暂停采集图像的信号槽连接的关键代码。

```

void Control::Restart_HIK1()
{
    if(!Is_HIK1_ok || !Is_HIK2_ok || !Is_SCT1_ok || !Is_SCT2_ok){
        InitAgain();
        Debug_Print1("开始采集出错");
        timer_restart_hik1->start();
    }else{
        timer_restart_hik1->stop();
        Debug_Print1( "Start collecting HIK1 pictures-----");
        QString debug_text = "Start collecting HIK1 pictrues-----";
        Debug_Print(debug_text);
        writelog(path_log[0],debug_text,Level1);
        HIKComparedPhotoThread_first->start();
        HIKCatchPhotoOPThread_first->start();
        HIKVideoStageThread_first->start();
        emit BeginDo_HIK1();
    }
}

```

图 4-9 采集控制信号槽设定关键代码(a)

```

void Control::Pause_HIK1()
{
    if(!Is_HIK1_ok || !Is_HIK2_ok || !Is_SCT1_ok || !Is_SCT2_ok){
        InitAgain();
        timer_pause_hik1->start();
    }else{
        timer_pause_hik1->stop();
        Debug_Print1( "Stop collecting hik1 pictures-----");
        QString debug_text = "Stop collecting hik1 pictruess-----";
        Debug_Print(debug_text);
        writelog(path_log[0],debug_text,Level1);
        emit PauseDo_HIK1();
    }
}

```

图 4-10 采集控制信号槽设定关键代码(b)

```

void HIKOpration::RestartPhotoTimer()
{
    if(Photo_timer){
        Photo_timer->start();
    }
}

void HIKOpration::PausePhotoTimer()
{
    if(Photo_timer && Photo_timer->isActive()){
        Photo_timer->stop();
    }
}

```

图 4-11 线程内定时器控制关键代码

图 4-9 和图 4-10 展示的是其中一个光学传感器的开启采集线程和暂停采集线程的关键代码，另一个光学传感器同此一致，每一个光学传感器和声学传感器在采集时分别有一个图像采集线程和录像采集线程，而声学传感器不需要采集图像，所以声学传感器只设置了一个线程用于保存视频录像。此处的开始线程和暂停线程是通过信号槽连通的，当程序发出开始采集、录像线程或暂停采集、录像信号时，便触发已经设定的槽函数从而开启或者暂停相应的线程内的定时器，实现线程的开启或暂停。图 4-11 展示了线程内定时器的开启与暂停的关键代码。

图 4-12 展示了开始采集光学、红外以及声学传感器的终端运行图，当客户端发送采集指令时，本系统则会立即开始采集图像数据，对于采集的红外图像数据的延时控制在 1s 之内，采集的光学图像数据的延时在 100ms 之内，采集声学传感器受硬件限制，在高频范围内其延时暂不做考虑，并且保证可以同时采集四个传感器的图像数据。



当系统实时采集多个通道的图像数据后，便会对采集的图像数据进行比对，图像采集和图像比对分别处于两个线程中，这两个线程共用着同一个队列。比对线程在检测到公用队列中有两张或两张以上的图片时，会从队列中取出其中两张图片进行比对，并在队列中删除这两张图片。因为需要快速比对图像，所以除了在比对时加入传统比对算法，该系统还将比较线程绑定到一个单独的内核，以减少在不同内核上调度带来的开销，提高比较效率。其关键代码如图 4-13 所示。

根据第三章的比对结果，本系统使用峰值信噪比算法作为比对算法，根据峰值信噪比算法的定义和公式，本系统设计出基于峰值信噪比算法的比对算法。在使用峰值信噪比比对算法进行图像比对之前，需要确保待比较的两张图像分辨率相同且图像完整。模块整体流程图如图 4-14 所示：

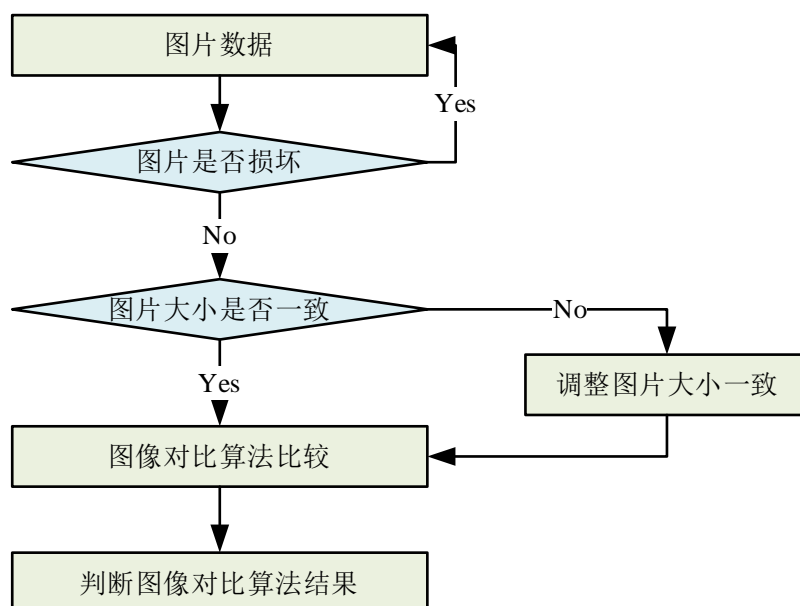


图 4-14 多通道传感器数据实时比对流程图

由第三章的结论可以得出，试验得出的峰值信噪比算法的阈值为 28。因此，当待比较的两张图片通过峰值信噪比算法比对出的结果大于等于 28 时，则判定图片没有发生变化，当待比较的两张图片通过峰值信噪比算法比对出的结果小于 28 时，则判定图片发生了变化，关键代码如图 4-15 所示：



```

QVector<Mat> ComparePhoto::SendPhotoformSDK()
{
    if(!JudgePhotoIsOk(SharePhoto[0]) || !JudgePhotoIsOk(SharePhoto[1]))
    {
        QString debug_text = QString("%1 : Unable to read photos!").arg(idnumber);
        Debug_Print(debug_text);
        writelog(path_log[idnumber],debug_text,Level1);
        QVector<Mat> mat;
        return mat;
    }
    struct timespec tic,toc,dur;
    clock_gettime(CLOCK_PROCESS_CPUTIME_ID,&tic);
    double a = 0.0;
    a = getPSNR(SharePhoto[0],SharePhoto[1]);
    clock_gettime(CLOCK_PROCESS_CPUTIME_ID,&toc);
    dur = diff(tic,toc);
    //以追加的方式打开文件
    QString debug_text = QString("%1 Channel : The time for only comparing the image similarity is %2 sec: %3 nsec\t"
        "The Similarity is %4")
        .arg(idnumber)
        .arg(dur.tv_sec)
        .arg(dur.tv_nsec)
        .arg(a);
    Debug_Print(debug_text);
    writelog(path_log[idnumber],debug_text,Level1);
    if(a < similar)
    {
        QVector<Mat> temp_mat;
        temp_mat.push_back(SharePhoto[0]);
        temp_mat.push_back(SharePhoto[1]);
        return temp_mat;
    }
    QVector<Mat> mat;
    return mat;
}

```

图 4-15 多通道实时比对关键代码

其中，峰值信噪比算法实现的代码如图 4-16 所示：

```

double ComparePhoto::getPSNR( Mat &I1, Mat &I2)
{
    if(I1.cols != I2.cols || I1.rows != I2.rows){
        return 0;
    }
    Mat s1;
    absdiff(I1,I2,s1); //|I1 - I2|
    s1.convertTo(s1,CV_32F);
    s1 = s1.mul(s1); //|I1 - I2|^2
    Scalar s = sum(s1);
    double sse = s.val[0] + s.val[1] + s.val[2];
    double mse = sse/static_cast<double>(static_cast<double>(I1.channels()) * I1.total());
    double psnr;
    if(sse <= 1e-10)
    {
        psnr = 0;
    }
    else
    {
        psnr = 10.0*log10((255*255)/mse);
        //保留十位小数
        QString str = QString::number(psnr);
        bool ok;
        psnr = QString(str).toDouble(&ok);
    }

    s1.release();
    return psnr;
}

```

图 4-16 峰值信噪比算法关键代码

图 4-17 展示的是其中某一个传感器的运行日志，上半部分展示的是待比较的两张图像没有存在差异性，下半部分展示的是待比较的两张图像存在着差异性，可以通过运行日志展示的计算消耗时间看出，无论待比较的两张图像是否存在差异，比较的时间都低于 100ms。

```
00_50_50_c0_00_08,2023-02-23 16:07:54.686,2: 0 Channel : The total time of Compare Photo are 0 sec: 32992808 nsec
00_50_50_c0_00_08,2023-02-23 16:07:54.848,0: 0 Channel : The size of Catch Photo are: 5
00_50_50_c0_00_08,2023-02-23 16:07:54.848,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.047,2: 0 Channel : The time for only comparing the image similarity is 0 sec: 13654213 nsec The Similarity is 35.055
00_50_50_c0_00_08,2023-02-23 16:07:54.858,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 13767034 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.033,0: 0 Channel : The size of Catch Photo are: 5
00_50_50_c0_00_08,2023-02-23 16:07:55.034,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.047,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 30986295 nsec The Similarity is 29.6125
00_50_50_c0_00_08,2023-02-23 16:07:55.047,2: 0 Channel : The total time of Compare Photo are 0 sec: 31121009 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.079,2: 0 Channel : Stop recording successfully
00_50_50_c0_00_08,2023-02-23 16:07:55.081,1: 0 Channel : Video is recording-----
00_50_50_c0_00_08,2023-02-23 16:07:55.202,0: 0 Channel : The size of Catch Photo are: 5
00_50_50_c0_00_08,2023-02-23 16:07:55.202,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.393,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 21639000 nsec The Similarity is 29.8822
00_50_50_c0_00_08,2023-02-23 16:07:55.215,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 21777468 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.379,0: 0 Channel : The size of Catch Photo are: 5
00_50_50_c0_00_08,2023-02-23 16:07:55.380,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.393,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 29930413 nsec The Similarity is 27.1411
00_50_50_c0_00_08,2023-02-23 16:07:55.393,1: 0 Channel : Picture changes!!!!!!!
00_50_50_c0_00_08,2023-02-23 16:07:55.405,0: 0 Channel : Enter shared memory
00_50_50_c0_00_08,2023-02-23 16:07:55.495,0: 0 Channel : Time to save pictures to shared memory is 0 sec: 22188677 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.549,0: 0 Channel : The size of Catch Photo are: 6
00_50_50_c0_00_08,2023-02-23 16:07:55.579,0: 0 Channel : The time to save pictures to the folder is 0 sec: 252294803 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.579,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 537124751 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.579,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.592,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 49607398 nsec The Similarity is 26.4526
00_50_50_c0_00_08,2023-02-23 16:07:55.592,1: 0 Channel : Picture changes!!!!!!!
00_50_50_c0_00_08,2023-02-23 16:07:55.603,0: 0 Channel : Enter shared memory
00_50_50_c0_00_08,2023-02-23 16:07:55.690,0: 0 Channel : Time to save pictures to shared memory is 0 sec: 294545860 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.714,0: 0 Channel : The size of Catch Photo are: 6
00_50_50_c0_00_08,2023-02-23 16:07:55.749,0: 0 Channel : The time to save pictures to the folder is 0 sec: 199222828 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.750,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 572801436 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.750,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.760,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 22720815 nsec The Similarity is 14.9419
00_50_50_c0_00_08,2023-02-23 16:07:55.760,1: 0 Channel : Picture changes!!!!!!!
00_50_50_c0_00_08,2023-02-23 16:07:55.768,0: 0 Channel : Enter shared memory
00_50_50_c0_00_08,2023-02-23 16:07:55.831,0: 0 Channel : Time to save pictures to shared memory is 0 sec: 144062034 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.887,0: 0 Channel : The size of Catch Photo are: 6
00_50_50_c0_00_08,2023-02-23 16:07:55.887,1: 0 Channel : The time to save pictures to the folder is 0 sec: 174112726 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.887,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 365380150 nsec
00_50_50_c0_00_08,2023-02-23 16:07:55.887,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:55.897,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 36859317 nsec The Similarity is 16.2081
00_50_50_c0_00_08,2023-02-23 16:07:55.897,1: 0 Channel : Picture changes!!!!!!!
00_50_50_c0_00_08,2023-02-23 16:07:55.905,0: 0 Channel : Enter shared memory
00_50_50_c0_00_08,2023-02-23 16:07:55.968,0: 0 Channel : Time to save pictures to shared memory is 0 sec: 215139336 nsec
00_50_50_c0_00_08,2023-02-23 16:07:56.024,0: 0 Channel : The time to save pictures to the folder is 0 sec: 201558135 nsec
00_50_50_c0_00_08,2023-02-23 16:07:56.024,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 479588581 nsec
00_50_50_c0_00_08,2023-02-23 16:07:56.042,0: 0 Channel : The size of Catch Photo are: 5
00_50_50_c0_00_08,2023-02-23 16:07:56.042,1: 0 Channel : size is 5
00_50_50_c0_00_08,2023-02-23 16:07:56.055,0: 0 Channel : The time for only comparing the image similarity is 0 sec: 44727740 nsec The Similarity is 18.5357
00_50_50_c0_00_08,2023-02-23 16:07:56.055,1: 0 Channel : Picture changes!!!!!!!
00_50_50_c0_00_08,2023-02-23 16:07:56.065,0: 0 Channel : Enter shared memory
00_50_50_c0_00_08,2023-02-23 16:07:56.130,0: 0 Channel : Time to save pictures to shared memory is 0 sec: 197877188 nsec
00_50_50_c0_00_08,2023-02-23 16:07:56.186,0: 0 Channel : The time to save pictures to the folder is 0 sec: 129659816 nsec
00_50_50_c0_00_08,2023-02-23 16:07:56.186,2: 0 Channel : The Total Time of Compare Photo are 0 sec: 410842956 nsec
00_50_50_c0_00_08,2023-02-23 16:07:56.222,0: 0 Channel : The size of Catch Photo are: 5
00_50_50_c0_00_08,2023-02-23 16:07:56.223,1: 0 Channel : size is 5
```

图 4-17 某传感器比对日志图

#### 4.2.3 多通道传感器数据实时传输算法设计与实现

该系统提供了两种数据实时传输的设计方法。其中，第一种方法是在 Ubuntu 系统中添加 FTP 用户，指定 FTP 目录作为保存图像数据的目录，所有采集到的图像和视频数据将被存储在该 FTP 目录中。而第二种方法则是通过共享内存与同一主机上的另一个进程进行信息传递。当光学传感器采集到的图像在本系统内判定发生了变化后，系统便会将变化的图像数据挂载在共享内存中，以便其他进程可以访问这些数据。这种方式能够减少数据传输的延迟，从而提高系统的实时性和响应速度。使用共享内存可以使得图像数据的传输延时几乎没有。在使用共享内存之前，需要先分配一个共享内存的空间，并确定共享内存的键值和空间大小。键值可以充当另一个进程访问共享内存的标识符，而指定的空间大小则限制了共享内存中可以存储的图像数据的数量。这样，其他进程可以通过键值去访问共享内存，并根据预先指定的空间大小读取图像数据。初始化的关键

代码如图 4-18 所示。当需要传输数据时，先打开共享内存，然后清空共享内存，接着将数据所在空间挂载在共享内存中，最后分离共享内存，传输数据的关键代码如图 4-19 所示。

```
int MyShared::initShare(int number,int idnumber,QString path)
{
    int shmid=0, ret=0;
    key_t key = ftok(".",number);|
    shmid = shmget(key, sizeof(Box), 0666|IPC_CREAT);
    QStringList temp;
    temp << QString::number(sizeof (Box)) << QString::number(key)
        << QString::number(width) << QString::number(height)
        << QString::number(3);
    WriteTextToJson(temp,path);
    SetAllpermissions(path);
    if (shmid > 0)
    {
        debug_text = QString("%1 Channel : Create a shared memory segment,id is %2")
            .arg(idnumber)
            .arg(shmid);
        Debug_Print(debug_text);
        writelog(path_log[idnumber],debug_text,Level3);
    }
    struct shmid_ds shmds;
    ret = shmctl(shmid, IPC_STAT, &shmds);

    if (ret == 0 )
    {
        debug_text = QString("%1 Channel : Size of memory segment is %2 \n Number of attaches are %3")
            .arg(idnumber)
            .arg(shmds.shm_segsz)
            .arg(static_cast<int>(shmds.shm_nattch));
        Debug_Print(debug_text);
        writelog(path_log[idnumber],debug_text,Level3);
    }
    else
    {
        debug_text = QString("%1 Channel : shmctl () call failed\n")
            .arg(idnumber);
        Debug_Print(debug_text);
        writelog(path_log[idnumber],debug_text,Level3);
    }
    return shmid;
}
```

图 4-18 初始化共享内存关键代码

```
void MyShared::HIK_CreateShare(QVector<Mat> SharaPhoto,int shmid,int shmids_size)
{
    void *shm = shmat(shmid,nullptr, 0);
    Box *pBox = reinterpret_cast<Box*>(shm);

    unsigned char *from1 = nullptr;
    from1 = static_cast<unsigned char*>(malloc(shmids_size*5));|
    unsigned char* datas = from1;
    for(int i = 0; i < SharaPhoto.size(); i++){

        memcpy(from1+(i*sizeof(unsigned char)*shmids_size),SharaPhoto[i].data,sizeof(unsigned char)*shmids_size);
        QDateTime current_date_time =QDateTime::currentDateTime();
        QString current_date =QString::number(i) + ".jpg";
        char* fileName;
        QByteArray ba = current_date.toLatin1(); // must
        fileName=ba.data();
        imwrite(fileName,SharaPhoto[i]);
    }
    from1 = datas;
    memset(pBox->ch,'\0',sizeof(pBox->ch));
    memcpy(pBox->ch,from1, SharaPhoto.size()*sizeof(unsigned char)*shmids_size);
    //分离共享内存
    shmdt(shm);
    free(from1);
    from1 = nullptr;
}
```

图 4-19 实时传输关键代码

#### 4.2.4 多通道传感器数据实时存储算法设计与实现

本系统使用本地存储方式保存图像数据,这需要高效地检索需要的图像数据。由于储存的图像数据量可能极其大,为了提高检索速度,本系统采用了分级目录的方式。分级目录按照数据类型、年、月、日、小时、实验场地、设备名称分级,使得客户端可以更快地找到所需的数据,数据文件的命名方式是按照存储时的分、秒和毫秒进行命名。同时,为了方便客户端访问,本系统支持 FTP 协议,使得客户端可以通过 FTP 协议快速获取所需的数据。在运行过程中,对于红外图像数据的存储延时应该控制在 1s 之内,光学图像数据的存储延时在 100ms 之内,存储的声学图像受硬件限制,在高频率范围内其延时暂不做考虑。分级目录示例图如图 4-20 所示:

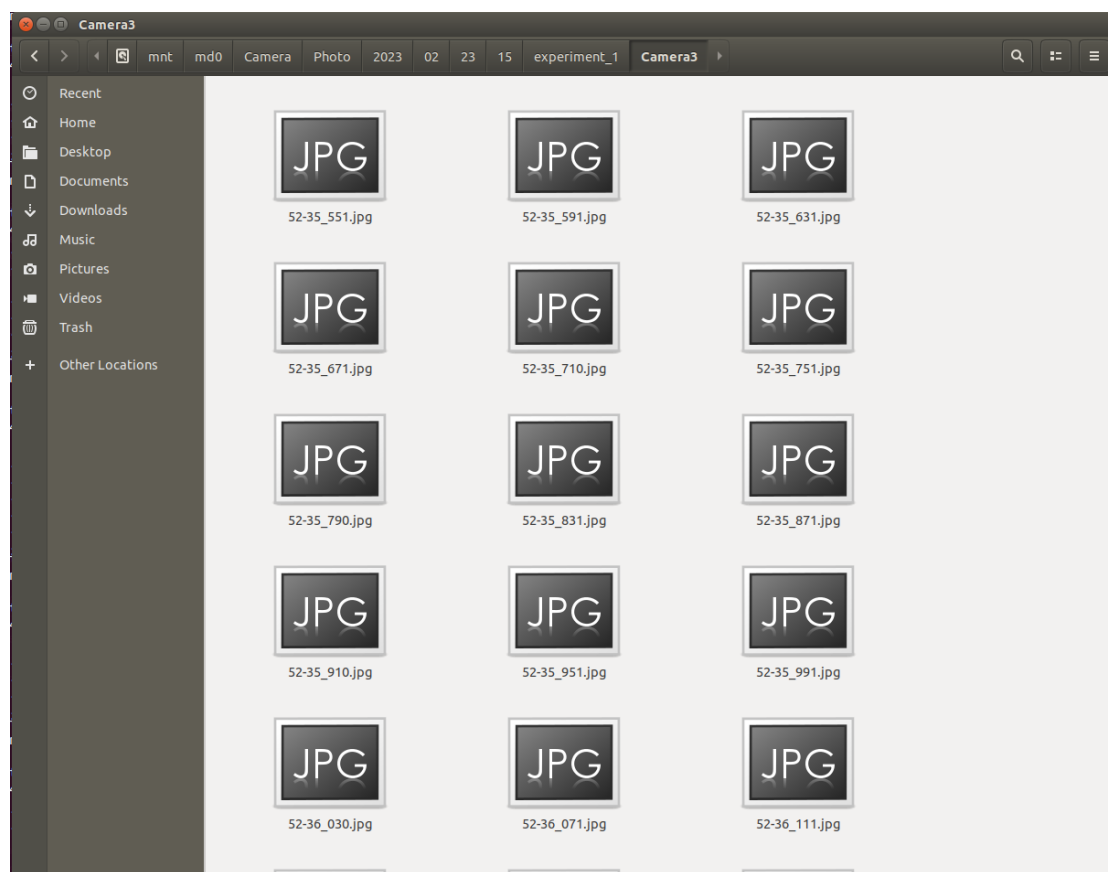


图 4-20 分级目录示例图

其次,也需要考虑磁盘空间是否足够存储大量的图像数据的问题。为了避免因磁盘空间不足而导致数据无法存储的情况,本系统在每次运行时提供已使用磁盘空间的信息。客户端可以根据这个信息来决定是否需要清理磁盘空间。如果客户端决定删除一些数据,本系统会按照客户端传递的百分比从最早的数据开始删除,以确保磁盘空间始终

足够存储新的数据。这样，本系统就能够有效地解决磁盘空间不足的问题，确保图像数据的安全性和可靠性。删除磁盘空间的关键代码如图 4-21 所示：

```
QStringList list;
list.append(all_photo);
list.append(all_video);
list.append(all_log);
for(int i = 0; i < list.size(); i++){
    QStringList TotalDir;
    QStringList YearDir = DirinDir(list[i]);

    for(int j = 0; j < YearDir.size(); j++){
        Debug_Print("The year is " + list[i]+YearDir[j]);
        QStringList MonthDir = DirinDir(list[i]+YearDir[j]);
        for(int z = 0; z < MonthDir.size(); z++){
            Debug_Print("The month is " + list[i]+YearDir[j]+"/"+MonthDir[z]+"/");
            QStringList DayDir = DirinDir(list[i]+YearDir[j]+"/"+MonthDir[z]+"/");
            for(int d = 0; d < DayDir.size(); d++){
                Debug_Print("The day is " + list[i]+YearDir[j]+"/"+MonthDir[z]+"/"+DayDir[d]+"/");
                TotalDir.append(list[i]+YearDir[j]+"/"+MonthDir[z]+"/"+DayDir[d]+"/");
            }
        }
    }
    int size = TotalDir.size()*(deleteProportion/100);
    size = size >= 1?size:1;
    for(int number = 0; number < size; number++){
        Debug_Print1("The delete dir is " + TotalDir[number]);
        QProcess *process = new QProcess();
        process->setEnvironment(process->environment());
        process->start("rm", QStringList() << "-r" << TotalDir[number] << "*");
        process->waitForFinished();
        QProcess *process1 = new QProcess();
        process1->setEnvironment(process1->environment());
        process1->start("rmdir", QStringList() << "-r" << TotalDir[number]);
        process1->waitForFinished();
        delete process;
        process = nullptr;
        delete process1;
        process1 = nullptr;
    }
}
```

图 4-21 删除磁盘空间关键代码

### 4.3 长期稳定运行的实现

```
#!/bin/bash

while [ 1 ]
do
    PIDS=`ps -ef |grep Sever |grep -v grep | awk '{print $2}'`
    if [ "$PIDS" != "" ]; then
        echo "App is runing!"
    else
        cd /home/alice/Documents/Camera/build-Sever-Desktop_Qt_5_13_2_GCC_64bit-Release/
        ./Camera
        sleep 2
    fi
done
```

图 4-22 守护进程脚本

由于本系统需要支持长时间稳定地运行，其运行时间需要超过 24 小时，所以需要系统加以守护进程来完成这一功能。首先将程序编译成可执行文件，接着编写脚本执行编译好的可执行文件，期间持续检测程序是否死亡，如果死亡，便会自动重启进程，

然后创建一个快捷方式使其能够双击运行程序，最后将快捷方式移入开机自启的文件夹中，程序便可以实现开机自启并加以守护进程了。守护进程的脚本代码如图 4-22 所示。

#### 4.4 本章小结

本章详细且具体地介绍了整个系统的多通道传感器数据实时采集、多通道传感器数据实时比对、多通道传感器数据实时传输、多通道传感器数据实时存储这四个模块的设计、关键代码以及实现。在第三章的实验结论上得出最适合本系统的传统比对算法为峰值信噪比算法，并根据其定义设计出比对算法，使比对模块可以更加快速高效且正确地比对出比对结果。并且为实现程序长期稳定地运行的功能，详细介绍了如何为程序加以守护进程守护。

## 第5章 系统测试

### 5.1 测试目的

进行系统测试可以检测系统的正常运行、稳定性能以及评估系统各项功能是否足以满足多通道图像数据实时采集、比对和存储需求,同时判断系统是否能够达到预期目标,并且在需要的情况下进行改进和优化。

### 5.2 测试方法

系统测试是软件开发生命周期中的一个关键步骤,它确保系统在交付给用户之前具有足够的质量和可靠性。在系统测试期间,软件被分为几个功能模块,然后进行确认测试。测试工作如下:

- 1、多通道传感器数据实时采集模块:测试系统中的图像实时采集的功能,同时采集多个传感器的功能以及断开重连的功能,以及其实时性。
- 2、多通道传感器数据实时比对模块:测试系统中的图像实时比对的功能以及其实时性。
- 3、多通道传感器数据实时传输模块:测试系统中 FTP 传输的功能,并测试系统中共享内存传输的功能以及其实时性。
- 4、多通道传感器数据实时存储模块:测试系统中本地存储图像数据的功能,并测试系统中删除图像数据的功能以及其实时性。
- 5、守护进程守护功能:测试守护进程守护程序的功能。

### 5.3 测试预期结果

- 1、多通道传感器数据实时采集模块预期测试结果:系统可以同时采集多个传感器的图像数据,并且在传感器断开连接后会立即重连传感器直至重连成功,可以控制传感器开始采集和暂停采集图像数据。并且,对于采集的红外图像数据的延时应该控制在 1s 之内,采集的光学图像数据的延时在 100ms 之内,采集声学传感器受硬件限制,在高频率范围内其延时暂不做考虑。
- 2、多通道传感器数据实时比对模块预期测试结果:系统可以同时比对多个传感器传递的图像数据,其比对时间在 100ms 内并比对准确率比较高。
- 3、多通道传感器数据实时传输模块预期测试结果:系统通过 FTP 完整地传递

的数据,并且系统可以通过共享内存传递数据,并可以通过共享内存的键值和空间大小获取到共享内存中的数据。并且使用共享内存传输图像数据几乎无延时。

4、多通道传感器数据实时存储模块预期测试结果:系统将需要存储的数据全部存储在本地磁盘上,目录等级划分正确且数据存储位置正确,同时,系统可以根据客户端传递的百分比删除对应的内容以及目录。并且,对于红外图像数据的存储延时应该控制在 1s 之内,光学图像数据的存储延时在 100ms 之内,存储的声学图像数据受硬件限制,在高频率范围内其延时暂不做考虑。

5、长期稳定运行功能:程序能够实现开机自启,同时,当进程中断后,会自动重启进程。并且可以稳定运行 24 小时以上。

## 5.4 测试实际结果

### 5.4.1 多通道传感器数据实时采集模块

系统一旦连接了四个传感器,即可同时对这些传感器进行图像数据采集,并且可以按照每个传感器设定的帧率进行采集。在接收到客户端发送的停止采集信息时,所有传感器将暂停采集工作;在接收到客户端发送的开始采集信号时,所有传感器将重新开始采集工作。当其中一个传感器与系统断开连接时,本系统将持续尝试重新连接该传感器,直至重连成功。相关的用例表如表 5-1 所示:

表 5-1 实时采集模块用例表

用例编号	测试内容	预期结果	测试结果
1	同时采集四个传感器的图像数据	采集成功	通过
2	实时采集四个传感器的图像数据,采集的图像帧率需要符合设定的帧率	采集成功,帧率一致	通过
3	是否可以控制实时采集	当客户端发出开始采集的信号便开始采集,当客户端发出暂停采集的信号便暂停采集	通过
4	采集图像的延时	红外传感器的图像采集延时不超过 1s; 光学传感器的图像采集延时不超过 100ms	通过

### 5.4.2 多通道传感器数据实时比对模块

两个光学传感器在采集图像数据后,会将数据存储到一个队列中。当队列中的图像数量超过两张时,比对线程便会开始从队列中取出最前面的两张图像进行比对。此比对



过程经过严格优化，无论图像发生怎样的变化，其时间都不会超过 100 毫秒。

为了实现快速、准确的图像比对，本系统采用了高效的算法和技术，并结合多线程技术进行优化。经过多次实验和测试，保证了该系统的高效性和准确性，同时保证了图像比对过程的稳定性和可靠性。相关用例表如表 5-2 所示：

表 5-2 多通道实时比对用例表

用例编号	测试内容	预期结果	测试结果
1	实时比对图像数据	比对时间不超过 100 毫秒，比对线程等待采集线程的数据	通过
2	精准的比对结果	可以比对出人眼所能观测的最细微的变化	通过

#### 5.4.3 多通道传感器数据实时传输模块

本系统的传输有两种，第一种是通过 FTP 将本系统所在服务器上的数据传到客户端，当传感器比对模块判定图像发生变化之后，就会将图像数据存储到本地目录上，与图像一同存储的还有各个传感器的视频数据，客户端通过 FTP 连接服务器后，就可以拉取到事先设定好的目录下所有的数据。第二种是通过共享内存进行传输，当传感器比对模块判定图像发生变化之后，本系统会将变化的图像数据传输到共享内存中，供另一个进程读取。相关用例表如表 5-3 所示：

表 5-3 多通道实时传输

用例编号	测试内容	预期结果	测试结果
1	FTP 传输数据	客户端可以通过 FTP 拉取到所有存储在指定目录下的数据	通过
2	共享内存传输	通过共享内存可以准确传输数据，另一个进程能拿到完整数据	通过
3	共享内存传输的延时	共享内存传输的图像数据无延时	通过

#### 5.4.4 多通道传感器数据实时存储模块

对于光学传感器而言，一旦开始采集图像，系统将按照帧数将视频数据保存在本地目录中。在比对模块检测到图像发生变化时，将保存相关的图像数据。针对红外传感器，由于没有比对模块，一旦开始采集数据，图像和视频数据将按照帧数保存在本地目录中。针对声学传感器，一旦开始采集，视频数据将按照帧数保存在本地目录中。数据存储的目录将根据数据类型、时间和设备进行划分。在接收到客户端的删除目录指令时，将按

照客户端传输的百分比删除最早的数据和目录。相关用例表如表 5-4 所示。

表 5-4 多通道实时存储

用例编号	测试内容	预期结果	测试结果
1	光学传感器存储数据	可以存储完整的数据	通过
2	红外传感器存储数据	可以存储完整的数据	通过
3	声学传感器存储数据	可以存储完整的数据	通过
4	本地目录划分正确	划分正确，每小时会自增对应的目录文件	通过
5	删除本地目录	正确删除了目录通过	通过
6	存储图像数据的延时	红外传感器图像数据在本地存储的延时不超过 1s；光学传感器图像数据在本地的存储延时不超过 100ms	通过

#### 5.4.5 长期稳定运行功能

对于整个程序而言，需要能够支持长期稳定地运行，为了达到该功能，需要保证进程中断后会自动重启。除此之外，作为采集数据的无界面程序，需要满足开机自启的要求。相关用例表如表 5-5 所示：

表 5-5 长期稳定运行用例表

用例编号	测试内容	预期结果	测试结果
1	开机自启	可以开机自启	通过
2	持续运行一小时进程情况	稳定运行，未中断	通过
3	持续运行五小时进程情况	稳定运行，未中断	通过
4	持续运行十小时进程情况	稳定运行，未中断	通过
5	持续运行十五小时进程情况	中断一次，自动重启程序	通过
6	持续运行二十小时进程情况	中断一次，自动重启程序	通过
7	持续运行二十五小时进程情况	中断两次，自动重启程序	通过
8	持续运行三十小时进程情况	中断两次，自动重启程序	通过

经过手工测试系统的长期稳定运行功能，可以证实本系统可以达到稳定运行超过 24 小时的指标。

### 5.5 本章小节

在本章中，采用手工测试的方式对系统的各项功能进行测试。经测试，系统各项功能均能实现，各项信息显示和数据处理准确，测试反馈效果好，系统规划设计合理，开发工作扎实。测试结果与预期结果一致，系统开发达到了项目建设的既定目标。

## 结论

文章在阐述了该课题的背景、使用技术的应用基础上,阐述了该领域的国内外研究现状。然后,简要介绍了 Qt、OpenCV、FFmpeg、SDK 等技术。在项目的要求阶段,分析了该系统开发的可行性。在对系统的需求进行分析时,首先对系统的设计原则进行了详细的描述,然后对系统进行了可行性分析,并对系统的功能进行了详细的描述,以及进行了详细的分析和功能设计,为以后的系统的设计和实施打下了基础。

该系统是一个运行在 Ubuntu 操作系统上的无界面多通道传感器实时采集、比对、存储系统,在整个系统中,图像数据量极其庞大,需要非常强的实时性,以便于可以第一时间发现试验场地发生的安全事故。在采集方面,本系统采用了多通道传感器采集的方式进行采集,相对于普遍的单通道传感器采集,该方法采集的面更广,可以更好地覆盖场景,适应不同的应用场景。除此之外,多通道传感器采集可以使后期更加便捷,可以在后期处理中将不同视角的图像信息进行合并,生成更加全面、准确的图像,处理的效率更加高效。除此之外,本系统的图像比对功能可以对图像进行初步的筛选,这可以为后续使用这个图像数据的应用减轻负载量,以便于更加快速地达到所要完成的目标,并且,对于筛选过的数据,可以通过共享内存传输给同主机的另一个进程,保证了高效的传输速率。对于图像的存储,本系统没有使用数据库而是直接存储在本地,这使得存储起来更加方便,进一步保证了数据的安全性,并且,本系统还会定期删除旧数据,这也保证了足够的存储空间。

当然,本系统还存在着一些不足的地方,虽然说,本系统设计的比对算法的精确度和速度都极高,可面对肉眼无法观测到的变化时,还是不能检测出发生了变化,在未来,可以考虑使用深度学习的方法来进行图像比对,以提高系统的准确性。除此之外,本系统是一个无界面系统,在最初设计时,考虑到在试验运行中,本系统并不会连接显示器,所以并没有设置界面,但随着后续的发展,该系统可能会连接多个试验现场的传感器,所以在未来,可以为系统设置一个可视化界面,用来管理控制多个试验现场的传感器的采集,以满足更多的应用需求。此外,本系统的数据有严格的保密性,且数据量极其庞大,如若后期发生丢失和泄漏则会造成巨大的损失,所以未来可以考虑使用分布式存储技术来提高系统的存储性能和可靠性。

## 致谢

文章停留在这一页，但人生之路还是很长。

行文至此，落笔为终，这篇致谢也意味着我的大学生涯将要结束，我也相信，所有的曾经都将成为背景，衬托将要破晓的黎明，纵使有万般不舍，皆是感恩，谨以此文，聊表热忱之心。

低徊愧人子，不敢叹风尘。感谢父母，二十多年来一直无条件支持我，养育之恩，虚文难述，愿吾亲长乐久安。

桃李无言，下自成蹊。感谢刘老师，在高性能计算实验室两年的实践教学以及毕业设计期间给予的关怀和指导，感谢四年来所有老师的教导。学生朽木，承蒙教诲。

追风赶月莫停留，平芜尽处是春山。感谢各位同窗好友，笑相遇，似觉琼枝玉树，暖日明霞光灿。感谢四年中所有的相遇与陪伴，是你们让我在校园中留下无数温暖的回忆，愿各位前程似锦，一路繁花。

阅万物，自生长。感谢自己，无数个自我鼓励的瞬间，无数个自我治愈的时刻，虽然笨拙但从未放弃，虽然缓慢却一直向前。

这一段旅行的结束也是另一场旅行的开始，无形间，命运的纺织已经开始了新一轮的运转。旅途终有一天会迎来终点，不必匆忙。

山水来路，终将于万物中见顶峰。

## 参考文献

- [1] WELLS P N T. Handbook of Image and Video Processing [J]. Physiological Measurement, 2001, 22(1).
- [2] CHEN M, XIONG X, JIANG X, et al. Discussion on Case Base Construction of Postgraduate Digital Image Processing Course under the Perspective of Artificial Intelligence [J]. Curriculum and Teaching Methodology, 2022, 5(13).
- [3] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet Large Scale Visual Recognition Challenge [J]. International Journal of Computer Vision, 2015, 115(3).
- [4] ASLAM Y, SANTHI N. A Comprehensive Survey on Optimization Techniques in Image Processing [J]. Materials Today: Proceedings, 2020, 24(Pt 3).
- [5] 李婉婷, 林志成, 姜东, et al. 一种基于峰值信噪比的某试验现场监控采集图像比对算法设计 [J]. 中国科技信息, 2023, (01): 115-118.
- [6] CONG H, BO S, LEI Z, et al. Event-Triggering State and Fault Estimation for a Class of Nonlinear Systems Subject to Sensor Saturations [J]. Sensors, 2021, 21(4).
- [7] 石晓伟. 图像处理在安全壳缺陷检测中的应用 [J]. 电子测量技术, 2016, 39(06): 89-93.
- [8] HIDEYUKI T, NAOKAZU Y. Image database systems: A survey [J]. Pattern Recognition, 1984, 17(1).
- [9] HUANG S H, HELD D, ABBEEL P, et al. Enabling robots to communicate their objectives [J]. Autonomous Robots, 2019, 43(2).
- [10] WEINI Z. Research on Recognition Method of Basketball Goals Based on Image Analysis of Computer Vision [J]. Journal of Sensors, 2021, 2021.
- [11] 田志伟, 马伟, 杨其长, et al. 温室智能装备系列之一百三十 植物工厂中机器视觉技术应用现状与挑战 [J]. 农业工程技术, 2022, 42(01): 36-45.
- [12] 吴振昊. 一种双通道图像采集及显示系统设计 [D]. 天津大学, 2017.
- [13] DEBEVEC P, HAWKINS T, TCHOU C, et al. Acquiring the reflectance field of a human face [J]. Computer graphics and interactive techniques, 2000, (27):145-146.
- [14] MANIAT M, CAMP C V, KASHANI A R. Deep learning-based visual crack detection using Google Street View images [J]. Neural Computing and Applications, 2021, 33(21).
- [15] 周颐, 孟明, 吴威, et al. 基于视频模型的虚拟现实视频融合系统 [J]. 系统仿真学报, 2018, 30(07):

2550-2557.

[16] THRUN S. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures [J]. The International Journal of Robotics Research, 2006, 25(5-6).

[17] FAIZ A S, HAZRY D, TAHA H A S. Study of Depth Control Method for Unmanned Underwater Vehicle (UUV) [J]. Journal of Environmental Science and Technology, 2017, 10(2).

[18] 田娟秀, 刘国才, 谷珊珊, et al. 医学图像分析深度学习方法研究与挑战 [J]. 自动化学报, 2018, 44(03): 401-424.

[19] 朱栋栋. 基于视觉检测与定位的快速路车辆异常行为识别 [D]. 北京邮电大学, 2021.

[20] 段峰, 王耀南, 雷晓峰, et al. 机器视觉技术及其应用综述 [J]. 自动化博览, 2002, (03): 62-64.

[21] ZALATOI A A, NOVIKAU D I. COMPARISON OF SPACE IMAGES OF A SURFACE OF EARTH TO A STANDARD OF A DIGITAL DISTRICT MAP [J]. Sistemnyj Analiz i Prikladnaâ Informatika, 2016, (1):4-10.

[22] WEI S, WEN M, YING M, et al. Improving the Spatial Resolution of Solar Images Based on an Improved Conditional Denoising Diffusion Probability Model [J]. The Astrophysical Journal Supplement Series, 2022, 263(2).

[23] ZHOU W, CONRAD B A, RAHIM S H, et al. Image quality assessment: from error visibility to structural similarity [J]. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 2004, 13(4).

[24] 曹军峰, 史加成, 罗海波, et al. 采用聚类分割和直方图均衡的图像增强算法 [J]. 红外与激光工程, 2012, 41(12): 3436-3441.

[25] 刘芳辉, 郭慧. 基于颜色体积核特征和改进的 LBP 的图像检索 [J]. 激光杂志, 2017, 38(05): 89-93.

[26] 王新余. 基于混合特征的图像检索算法研究 [D]. 华中科技大学, 2017.

[27] 谢勇, 贾惠珍, 王同罕, et al. 图像去雾算法综述 [J]. 计算机与数字工程, 2022, 50(12): 2765-2774.

[28] BUSHRA Z, REHAN A, NOUMAN A, et al. Image classification by addition of spatial information based on histograms of orthogonal vectors [J]. PloS one, 2018, 13(6).

[29] ALI N, BAJWA K B, SABLATNIG R, et al. Image retrieval by addition of spatial information based on histograms of triangular regions [J]. Computers and Electrical Engineering, 2016, 54.

[30] K. K S, A. L P, P. H, et al. Updated Global Reference Models of Broadband Coherent Infrasound Signals for Atmospheric Studies and Civilian Applications [J]. Earth and Space Science, 2022, 9(7).

- [31] DONATELLA G. Metaheuristic Algorithms Applied to Color Image Segmentation on HSV Space [J]. *Journal of Imaging*, 2022, 8(1).
- [32] AHAD T M N, SIULY S, KATE W, et al. Textural feature based intelligent approach for neurological abnormality detection from brain signal data [J]. *PloS one*, 2022, 17(11).
- [33] 关泽文, 陈建文, 鲍拯. 一种改进的基于峰值信噪比-高阶奇异值分解的天波超视距雷达自适应海杂波抑制算法 [J]. *电子与信息学报*, 2019, 41(07): 1743-1750.
- [34] 慈睿佳, 李仁伟. 视觉传达约束下模糊人脸图像多尺度特征重建模型 [J]. *现代电子技术*, 2023, 46(02): 164-168.
- [35] BO W, JIANXIN Z, HUANQING Y, et al. An ameliorated deep dense convolutional neural network for accurate recognition of casting defects in X-ray images [J]. *Knowledge-Based Systems*, 2021, 226.
- [36] SARAHI S A, JESUS O, ALDO H, et al. Efficient Face Recognition System for Operating in Unconstrained Environments [J]. *Journal of imaging*, 2021, 7(9).
- [37] JABEEN D E, KHAN T, IFTIKHAR R, et al. An Algorithm to Reduce Compression Ratio in Multimedia Applications [J]. *Computers, Materials & Continua*, 2022, 74(1).
- [38] 姜洪溪, 袁磊. JPEG 压缩标准在医学图像压缩中的应用 [J]. *襄樊学院学报*, 2003, (05): 52-54.
- [39] 李定川. JPEG2000 图像压缩标准及其关键算法的应用 [J]. *影像技术*, 2010, 22(04): 26-31.
- [40] M. S, G. Y, C.V. S. Simultaneous Covariance Driven Correspondence (CDC) and transformation estimation in the expectation maximization framework [J]. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007, (10636919).
- [41] MOREIRA D D, ISMAEL R D, COELHO K V, et al. Structural similarity versus secretion composition in colleter of congeneric species of *Prepusa* (Gentianaceae) [J]. *Flora*, 2022, 294.
- [42] 邓星涛, 闵海. 基于纹理结构相似性和边缘信息的水平集纹理图像分割 [J]. *合肥工业大学学报 (自然科学版)*, 2022, 45(01): 30-38+124.
- [43] RITUDHWAJ T, R M A, ADVIKA G, et al. Structural similarity-based prediction of host factors associated with SARS-CoV-2 infection and pathogenesis [J]. *Journal of biomolecular structure & dynamics*, 2021, 40(13):5868-5879.
- [44] SHADE D, MARSZALEK B, WALTON K S. Structural similarity, synthesis, and adsorption properties of aluminum-based metal-organic frameworks [J]. *Adsorption*, 2021, 27: 227–236.
- [45] DI Z, YONG Z, JIAQI Z, et al. Structural similarity preserving GAN for infrared and visible image fusion [J]. *International Journal of Wavelets, Multiresolution and Information Processing*, 2021, 19(01).

- [46] ENHAO G, M P J, MAX W, et al. Deep learning enables reduced gadolinium dose for contrast-enhanced brain MRI [J]. Journal of magnetic resonance imaging : JMRI, 2018, 48(2).
- [47] RYOSUKE K, HIDEKI O. Noise Reduction Using Singular Value Decomposition with Jensen–Shannon Divergence for Coronary Computed Tomography Angiography [J]. Diagnostics, 2023, 13(6).
- [48] 赵莉, 王莎莎, 刘道华, et al. 一种改进感知哈希算法的 2DPCANet 人脸识别方法 [J]. 信阳师范学院学报(自然科学版), 2022, 35(03): 494-498.
- [49] 刘凯. 基于压缩感知的图像哈希算法 [D]. 广西师范大学, 2016.
- [50] 刘向东, 刘泱. 基于压缩感知的图像哈希算法 [J]. 数字通信世界, 2017, (10): 237.
- [51] 王林帆. 基于感知哈希的网络时间隐蔽信道的检测研究 [D]. 华侨大学, 2020.
- [52] BIROUK W, LAHOULOU A, MELIT A, et al. Robust perceptual fingerprint image hashing: a comparative study [J]. International Journal of Biometrics, 2023, 15(1).



## 研究成果

目前，本论文所研究的内容已发表了一篇公开期刊论文<sup>[5]</sup>以及申请了一项计算机软件著作权，计算机软件著作权如图 5-1 所示：

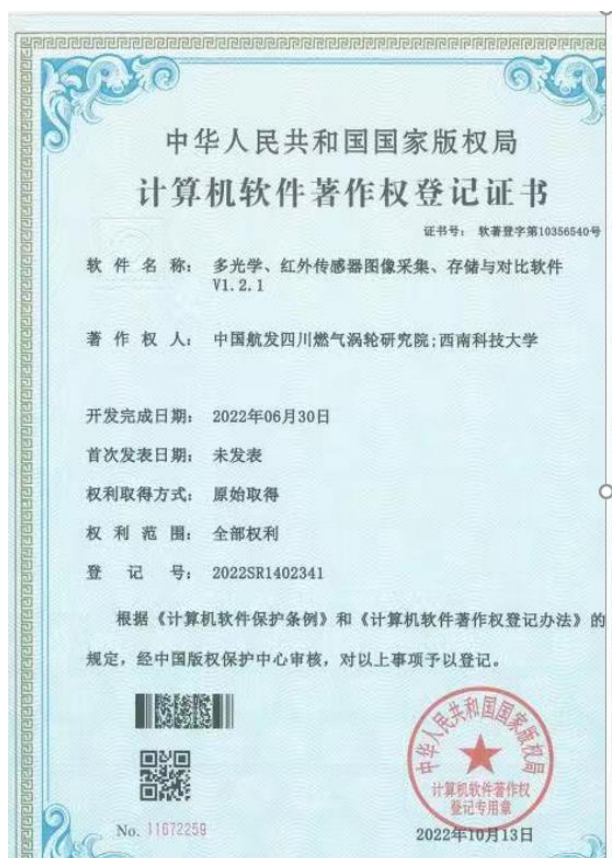


图 5-1 计算机软件著作权登记证书